

Alibaba Cloud Apsara Stack Enterprise

User Guide - Analytics and Artificial Intelligence

Version: 1909, Internal: V3.8.1

Issue: 20200116

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.









1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent

ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document

.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK.
Courier font	Courier font is used for commands.	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
{{ or {a b}}	This format is used for a required value, where only one item can be selected.	switch { <i>active</i> <i>stand</i> }

Contents

Legal disclaimer.....	I
Document conventions.....	I
1 MaxCompute.....	1
1.1 What is MaxCompute?.....	1
1.2 Usage notes.....	3
1.3 Quick start.....	6
1.3.1 Overview.....	6
1.3.2 Configure the client.....	7
1.3.3 Add and delete users.....	9
1.3.4 Grant and view permissions.....	10
1.3.4.1 Overview.....	10
1.3.4.2 ACL authorization.....	10
1.3.4.3 Policy authorization.....	11
1.3.4.4 View permissions.....	12
1.3.5 Create and authorize a role.....	12
1.3.6 Create or delete a table.....	13
1.3.6.1 Create a table.....	13
1.3.6.2 Obtain table information.....	14
1.3.6.3 Delete a table.....	15
1.3.7 Import or export data.....	15
1.3.8 Run SQL.....	15
1.3.8.1 Overview.....	15
1.3.8.2 SELECT statement.....	15
1.3.8.3 INSERT statement.....	16
1.3.8.4 JOIN statements.....	17
1.3.8.5 Other limits.....	17
1.3.9 Compile and use UDFs.....	17
1.3.9.1 Overview.....	17
1.3.9.2 UDF example.....	18
1.3.9.3 UDAF example.....	19
1.3.9.4 UDTF example.....	20
1.3.10 Compile and run a MapReduce job.....	20
1.3.11 Compile and run a Graph job.....	21
1.4 Basic concepts and common commands.....	23
1.4.1 Terms.....	23
1.4.2 Common commands.....	32
1.4.2.1 Introduction.....	32
1.4.2.2 Project operations.....	32
1.4.2.3 Table operations.....	35
1.4.2.4 Instance operations.....	38

1.4.2.5 Resource operations.....	41
1.4.2.6 Function operations.....	43
1.4.2.7 Tunnel operations.....	44
1.4.2.8 Other operations.....	52
1.5 MaxCompute SQL.....	56
1.5.1 Overview.....	56
1.5.1.1 Scenarios.....	56
1.5.1.2 Reserved words.....	57
1.5.1.3 Partitioned table.....	57
1.5.1.4 Type conversion.....	58
1.5.1.4.1 Explicit type conversion.....	58
1.5.1.4.2 Implicit type conversion and its scope.....	59
1.5.1.4.3 SQL built-in functions.....	63
1.5.1.4.4 CASE WHEN.....	63
1.5.1.4.5 Partition column.....	64
1.5.1.4.6 UNION ALL.....	64
1.5.1.4.7 Conversion between string and datetime types.....	64
1.5.2 Operators.....	65
1.5.2.1 Relational operators.....	65
1.5.2.2 Arithmetic operators.....	67
1.5.2.3 Bitwise operators.....	68
1.5.2.4 Logical operators.....	68
1.5.3 DDL statements.....	69
1.5.3.1 Table operations.....	69
1.5.3.1.1 Create a table.....	69
1.5.3.1.2 Delete a table.....	72
1.5.3.1.3 Rename a table.....	72
1.5.3.1.4 Modify the comment of a table.....	73
1.5.3.1.5 Modify the lifecycle of a table.....	73
1.5.3.1.6 Disable the lifecycle.....	74
1.5.3.1.7 Modify the LastDataModifiedTime value of a table.....	74
1.5.3.1.8 Clear data from a non-partitioned table.....	75
1.5.3.1.9 Archive table data.....	75
1.5.3.1.10 Forcibly delete data from a table (partition).....	76
1.5.3.2 View-based operation.....	77
1.5.3.2.1 Create a view.....	77
1.5.3.2.2 Delete a view.....	78
1.5.3.2.3 Rename a view.....	78
1.5.3.3 Column and partition operations.....	78
1.5.3.3.1 Add a partition.....	78
1.5.3.3.2 Delete a partition.....	79
1.5.3.3.3 Add a column.....	80
1.5.3.3.4 Change a column name.....	80
1.5.3.3.5 Modify the comment of a column or partition.....	80
1.5.3.3.6 Modify the LastDataModifiedTime value of a partition.....	81

1.5.3.3.7 Modify partition values.....	81
1.5.4 DML statements.....	82
1.5.4.1 INSERT statement.....	82
1.5.4.1.1 Update the data of a table.....	82
1.5.4.1.2 Output data to multiple objects.....	83
1.5.4.1.3 Output data to a dynamic partition.....	84
1.5.4.2 SELECT statement.....	85
1.5.4.2.1 SELECT operation.....	85
1.5.4.2.2 Subquery.....	90
1.5.4.3 UNION statements.....	90
1.5.4.3.1 UNION ALL.....	90
1.5.4.4 JOIN statement.....	91
1.5.4.4.1 JOIN operation.....	91
1.5.4.4.2 MAPJOIN HINT.....	93
1.5.4.5 EXPLAIN statement.....	94
1.5.4.6 GROUPING SETS.....	97
1.5.4.6.1 Overview.....	97
1.5.4.6.2 Example.....	98
1.5.4.6.3 CUBE and ROLLUP.....	99
1.5.4.6.4 GROUPING and GROUPING_ID.....	99
1.5.5 SELECT TRANSFORM.....	100
1.5.5.1 Overview.....	100
1.5.5.2 SELECT TRANSFORM examples.....	102
1.5.5.2.1 Call Shell scripts.....	102
1.5.5.2.2 Call Python scripts.....	103
1.5.5.2.3 Call Java scripts.....	104
1.5.5.2.4 Call scripts of other languages.....	105
1.5.5.2.5 Call scripts in series.....	106
1.5.5.3 Performance advantages.....	106
1.5.6 UNION, INTERSECT, and EXCEPT.....	107
1.5.7 Built-in functions.....	110
1.5.7.1 Mathematical functions.....	110
1.5.7.1.1 ABS.....	110
1.5.7.1.2 ACOS.....	111
1.5.7.1.3 ASIN.....	112
1.5.7.1.4 ATAN.....	112
1.5.7.1.5 CEIL.....	113
1.5.7.1.6 CONV.....	113
1.5.7.1.7 COS.....	114
1.5.7.1.8 COSH.....	114
1.5.7.1.9 COT.....	115
1.5.7.1.10 EXP.....	115
1.5.7.1.11 FLOOR.....	115
1.5.7.1.12 LN.....	116
1.5.7.1.13 LOG.....	116

1.5.7.1.14 POW.....	117
1.5.7.1.15 RAND.....	117
1.5.7.1.16 ROUND.....	118
1.5.7.1.17 SIN.....	118
1.5.7.1.18 SINH.....	119
1.5.7.1.19 SQRT.....	119
1.5.7.1.20 TAN.....	120
1.5.7.1.21 TANH.....	120
1.5.7.1.22 TRUNC.....	120
1.5.7.1.23 Additional mathematical functions.....	121
1.5.7.1.24 LOG2.....	122
1.5.7.1.25 LOG10.....	122
1.5.7.1.26 BIN.....	122
1.5.7.1.27 HEX.....	123
1.5.7.1.28 UNHEX.....	123
1.5.7.1.29 RADIANS.....	124
1.5.7.1.30 DEGREES.....	124
1.5.7.1.31 SIGN.....	125
1.5.7.1.32 E.....	125
1.5.7.1.33 PI.....	125
1.5.7.1.34 FACTORIAL.....	126
1.5.7.1.35 CBRT.....	126
1.5.7.1.36 SHIFTLEFT.....	126
1.5.7.1.37 SHIFTRIGHT.....	127
1.5.7.1.38 SHIFTRIGHTUNSIGNED.....	127
1.5.7.2 String processing functions.....	128
1.5.7.2.1 CHAR_MATCHCOUNT.....	128
1.5.7.2.2 CHR.....	128
1.5.7.2.3 CONCAT.....	129
1.5.7.2.4 INSTR.....	129
1.5.7.2.5 IS_ENCODING.....	130
1.5.7.2.6 KEYVALUE.....	131
1.5.7.2.7 LENGTH.....	132
1.5.7.2.8 LENGTHB.....	132
1.5.7.2.9 MD5.....	133
1.5.7.2.10 PARSE_URL.....	133
1.5.7.2.11 REGEXP_EXTRACT.....	134
1.5.7.2.12 REGEXP_INSTR.....	135
1.5.7.2.13 REGEXP_SUBSTR.....	136
1.5.7.2.14 REGEXP_COUNT.....	137
1.5.7.2.15 SPLIT_PART.....	137
1.5.7.2.16 REGEXP_REPLACE.....	138
1.5.7.2.17 SUBSTR.....	139
1.5.7.2.18 TOLOWER.....	140
1.5.7.2.19 TOUPPER.....	141

1.5.7.2.20 TO_CHAR.....	141
1.5.7.2.21 TRIM.....	142
1.5.7.2.22 LTRIM.....	142
1.5.7.2.23 RTRIM.....	142
1.5.7.2.24 REVERSE.....	143
1.5.7.2.25 SPACE.....	143
1.5.7.2.26 REPEAT.....	144
1.5.7.2.27 ASCII.....	144
1.5.7.2.28 URL_ENCODE.....	145
1.5.7.2.29 URL_DECODE.....	146
1.5.7.2.30 Additional string processing functions.....	146
1.5.7.2.31 CONCAT_WS.....	147
1.5.7.2.32 LPAD.....	147
1.5.7.2.33 RPAD.....	148
1.5.7.2.34 REPLACE.....	148
1.5.7.2.35 SOUNDEX.....	149
1.5.7.2.36 SUBSTRING_INDEX.....	149
1.5.7.2.37 TRANSLATE.....	150
1.5.7.3 Date processing functions.....	150
1.5.7.3.1 DATEADD.....	150
1.5.7.3.2 DATEDIFF.....	152
1.5.7.3.3 DATEPART.....	152
1.5.7.3.4 DATETRUNC.....	153
1.5.7.3.5 GETDATE.....	153
1.5.7.3.6 ISDATE.....	154
1.5.7.3.7 LASTDAY.....	154
1.5.7.3.8 TO_DATE.....	155
1.5.7.3.9 TO_CHAR.....	156
1.5.7.3.10 UNIX_TIMESTAMP.....	156
1.5.7.3.11 FROM_UNIXTIME.....	157
1.5.7.3.12 WEEKDAY.....	157
1.5.7.3.13 WEEKOFYEAR.....	158
1.5.7.3.14 Additional date functions.....	159
1.5.7.3.15 YEAR.....	159
1.5.7.3.16 QUARTER.....	160
1.5.7.3.17 MONTH.....	160
1.5.7.3.18 DAY.....	160
1.5.7.3.19 DAYOFMONTH.....	161
1.5.7.3.20 HOUR.....	161
1.5.7.3.21 MINUTE.....	162
1.5.7.3.22 SECOND.....	162
1.5.7.3.23 FROM_UTC_TIMESTAMP.....	163
1.5.7.3.24 CURRENT_TIMESTAMP.....	163
1.5.7.3.25 ADD_MONTHS.....	164
1.5.7.3.26 LAST_DAY.....	164

1.5.7.3.27 NEXT_DAY.....	165
1.5.7.3.28 MONTHS_BETWEEN.....	165
1.5.7.4 Window functions.....	166
1.5.7.4.1 Overview.....	166
1.5.7.4.2 COUNT.....	167
1.5.7.4.3 AVG.....	168
1.5.7.4.4 MAX.....	169
1.5.7.4.5 MIN.....	169
1.5.7.4.6 MEDIAN.....	170
1.5.7.4.7 STDDEV.....	170
1.5.7.4.8 STDDEV_SAMP.....	171
1.5.7.4.9 SUM.....	172
1.5.7.4.10 DENSE_RANK.....	172
1.5.7.4.11 RANK.....	174
1.5.7.4.12 LAG.....	175
1.5.7.4.13 LEAD.....	176
1.5.7.4.14 PERCENT_RANK.....	177
1.5.7.4.15 ROW_NUMBER.....	177
1.5.7.4.16 CLUSTER_SAMPLE.....	178
1.5.7.4.17 NTILE.....	180
1.5.7.4.18 NTH_VALUE.....	181
1.5.7.4.19 CUME_DIST.....	182
1.5.7.4.20 FIRST_VALUE.....	183
1.5.7.4.21 LAST_VALUE.....	185
1.5.7.5 Aggregate functions.....	187
1.5.7.5.1 Overview.....	187
1.5.7.5.2 COUNT.....	187
1.5.7.5.3 AVG.....	188
1.5.7.5.4 MAX.....	189
1.5.7.5.5 MIN.....	189
1.5.7.5.6 MEDIAN.....	190
1.5.7.5.7 STDDEV.....	190
1.5.7.5.8 STDDEV_SAMP.....	191
1.5.7.5.9 SUM.....	191
1.5.7.5.10 WM_CONCAT.....	192
1.5.7.5.11 PERCENTILE.....	192
1.5.7.5.12 Additional aggregate functions.....	193
1.5.7.5.13 COLLECT_LIST.....	194
1.5.7.5.14 COLLECT_SET.....	194
1.5.7.5.15 VARIANCE/VAR_POP.....	194
1.5.7.5.16 VAR_SAMP.....	195
1.5.7.5.17 COVAR_POP.....	196
1.5.7.5.18 COVAR_SAMP.....	196
1.5.7.6 Other functions.....	197
1.5.7.6.1 ARRAY.....	197

1.5.7.6.2 ARRAY_CONTAINS.....	197
1.5.7.6.3 CAST.....	198
1.5.7.6.4 COALESCE.....	198
1.5.7.6.5 DECODE.....	199
1.5.7.6.6 EXPLODE.....	200
1.5.7.6.7 GET_IDCARD_AGE.....	201
1.5.7.6.8 GET_IDCARD_BIRTHDAY.....	201
1.5.7.6.9 GET_IDCARD_SEX.....	201
1.5.7.6.10 GREATEST.....	202
1.5.7.6.11 INDEX.....	202
1.5.7.6.12 MAX_PT.....	203
1.5.7.6.13 ORDINAL.....	204
1.5.7.6.14 LEAST.....	205
1.5.7.6.15 SIZE.....	205
1.5.7.6.16 SPLIT.....	206
1.5.7.6.17 STR_TO_MAP.....	206
1.5.7.6.18 UNIQUE_ID.....	207
1.5.7.6.19 UUID.....	207
1.5.7.6.20 SAMPLE.....	207
1.5.7.6.21 CASE WHEN expression.....	208
1.5.7.6.22 IF.....	209
1.5.7.6.23 Additional functions.....	210
1.5.7.6.24 MAP.....	210
1.5.7.6.25 MAP_KEYS.....	210
1.5.7.6.26 MAP_VALUES.....	211
1.5.7.6.27 SORT_ARRAY.....	211
1.5.7.6.28 POSEXPLODE.....	211
1.5.7.6.29 STRUCT.....	212
1.5.7.6.30 NAMED_STRUCT.....	212
1.5.7.6.31 INLINE.....	213
1.5.7.6.32 BETWEEN AND expression.....	213
1.5.7.6.33 NVL.....	214
1.5.8 UDFs.....	215
1.5.8.1 Overview.....	215
1.5.8.2 Types of parameters and returned values.....	216
1.5.8.3 UDFs.....	218
1.5.8.4 UDAFs.....	219
1.5.8.5 UDTFs.....	222
1.5.8.5.1 Overview.....	222
1.5.8.5.2 UDTF description.....	223
1.5.8.6 Python UDFs.....	226
1.5.8.6.1 Restricted environment.....	226
1.5.8.6.2 Third-party libraries.....	228
1.5.8.6.3 Types of parameters and returned values.....	228
1.5.8.6.4 UDFs.....	230

1.5.8.6.5 UDAFs.....	230
1.5.8.6.6 UDTFs.....	231
1.5.8.6.7 Reference resources.....	232
1.5.9 UDTs.....	234
1.5.9.1 Overview.....	234
1.5.9.2 Feature summary.....	235
1.5.9.3 Feature details.....	236
1.5.9.4 More examples.....	242
1.5.9.4.1 Example of using Java arrays.....	242
1.5.9.4.2 Example of using JSON.....	242
1.5.9.4.3 Example of using composite types.....	242
1.5.9.4.4 Example of aggregation.....	243
1.5.9.4.5 Example of using table-valued functions.....	244
1.5.9.5 Feature advantages.....	244
1.5.9.6 Performance advantages.....	245
1.5.9.7 Security advantages.....	245
1.5.10 UDJ.....	245
1.5.10.1 Overview.....	245
1.5.10.2 UDJ usage.....	246
1.5.10.2.1 Examples.....	246
1.5.10.2.2 Use Java to write the UDJ code.....	247
1.5.10.2.3 Create a UDJ function in MaxCompute.....	249
1.5.10.2.4 Use UDJ in MaxCompute SQL.....	249
1.5.10.2.5 Pre-sorting.....	252
1.5.10.3 Performance advantages.....	254
1.5.11 MaxCompute SQL limits.....	255
1.5.12 Common MaxCompute SQL errors and solutions.....	257
1.5.12.1 Data skew.....	257
1.5.12.1.1 Overview.....	257
1.5.12.1.2 GROUP BY skew.....	257
1.5.12.1.3 DISTRIBUTE BY skew.....	258
1.5.12.1.4 JOIN skew.....	258
1.5.12.1.5 MULTI-DISTINCT skew.....	258
1.5.12.1.6 Data skew caused by misuse of dynamic partitioning.....	259
1.5.12.2 Quota and resource usage.....	259
1.5.12.3 MaxCompute storage optimization tips.....	261
1.5.12.4 UDF OOM error.....	263
1.5.13 Common MaxCompute SQL parameter settings.....	264
1.5.13.1 MAP configurations.....	264
1.5.13.2 JOIN configurations.....	264
1.5.13.3 Reduce configurations.....	265
1.5.13.4 UDF configurations.....	265
1.5.13.5 MAPJOIN configurations.....	266
1.5.13.6 Configure data skew.....	266
1.5.14 MapReduce-to-SQL conversion for execution.....	267

1.5.14.1 Overview.....	267
1.5.14.2 Local running settings.....	267
1.5.14.3 DataWorks running settings.....	268
1.5.14.4 View running details.....	268
1.5.15 Appendix.....	270
1.5.15.1 Escape character.....	270
1.5.15.2 LIKE matching.....	271
1.5.15.3 Regular expressions.....	272
1.5.15.4 Reserved words.....	274
1.6 MaxCompute Tunnel.....	275
1.6.1 Tunnel SDK overview.....	275
1.6.1.1 Overview.....	275
1.6.1.2 TableTunnel.....	275
1.6.1.3 UploadSession.....	276
1.6.1.4 DownloadSession.....	278
1.6.2 Tunnel SDK example.....	279
1.6.2.1 Simple upload example.....	279
1.6.2.2 Simple download example.....	281
1.6.2.3 Multithread upload example.....	282
1.6.2.4 Multithread download example.....	284
1.6.3 Appendix.....	286
1.6.3.1 Tunnel upload/download FAQ.....	286
1.6.3.2 Common tunnel error codes.....	289
1.7 MaxCompute MapReduce.....	290
1.7.1 Overview.....	290
1.7.1.1 MapReduce.....	290
1.7.1.2 Extended MapReduce.....	293
1.7.2 Features.....	293
1.7.2.1 Run command.....	293
1.7.2.2 Concepts.....	295
1.7.2.2.1 MapReduce.....	295
1.7.2.2.2 Sorting.....	295
1.7.2.2.3 Partition.....	296
1.7.2.2.4 Combiner.....	296
1.7.2.2.5 Input and output.....	296
1.7.2.2.6 Read data from resources.....	296
1.7.2.2.7 Run MapReduce tasks locally.....	297
1.7.3 SDK introduction.....	300
1.7.3.1 Major API overview.....	300
1.7.3.2 API description.....	301
1.7.3.2.1 MapperBase.....	301
1.7.3.2.2 ReducerBase.....	301
1.7.3.2.3 TaskContext.....	302
1.7.3.2.4 JobConf.....	303
1.7.3.2.5 JobClient.....	305

1.7.3.2.6 RunningJob.....	305
1.7.3.2.7 InputUtils.....	305
1.7.3.2.8 OutputUtils.....	306
1.7.3.2.9 Pipeline.....	306
1.7.4 Data types.....	307
1.7.5 Limits.....	308
1.7.6 Sample programs.....	309
1.7.6.1 WordCount example.....	309
1.7.6.2 MapOnly example.....	310
1.7.6.3 Example: Input and output data to multiple objects.....	312
1.7.6.4 Multi-task example.....	314
1.7.6.5 Secondary sorting example.....	316
1.7.6.6 Resource usage example.....	318
1.7.6.7 Example for using counters.....	319
1.7.6.8 grep example.....	321
1.7.6.9 JOIN example.....	324
1.7.6.10 Sleep example.....	326
1.7.6.11 unique example.....	329
1.7.6.12 Sort example.....	331
1.7.6.13 Example of using partitioned table as an input.....	332
1.7.6.14 Pipeline example.....	334
1.8 MaxCompute Graph.....	335
1.8.1 Graph overview.....	335
1.8.1.1 Graph overview.....	335
1.8.1.2 Graph data structure.....	336
1.8.1.3 Graph logic.....	337
1.8.1.3.1 Load graph.....	337
1.8.1.3.2 Iterative computation.....	338
1.8.1.3.3 End of iteration.....	339
1.8.2 Graph feature overview.....	339
1.8.2.1 Run a job.....	339
1.8.2.2 Input and output.....	341
1.8.2.3 Read data from resources.....	342
1.8.2.3.1 Add resource in Graph program.....	342
1.8.2.3.2 Use resources in Graph.....	342
1.8.3 Graph SDK introduction.....	343
1.8.4 Development and debugging.....	344
1.8.4.1 Development procedure.....	344
1.8.4.2 Development example.....	344
1.8.4.3 Local debugging.....	345
1.8.4.4 Temporary directory for local jobs.....	348
1.8.4.5 Cluster debugging.....	349
1.8.4.6 Performance optimization.....	349
1.8.4.6.1 Configure job parameters.....	349
1.8.4.6.2 Use Combiner.....	351

1.8.4.6.3 Reduce data input.....	351
1.8.4.6.4 JAR packages.....	351
1.8.5 Application limits.....	352
1.8.6 Sample programs.....	353
1.8.6.1 SSSP.....	353
1.8.6.2 PageRank.....	356
1.8.6.3 K-means clustering.....	358
1.8.6.4 BiPartiteMatching.....	362
1.8.6.5 Strongly-connected component.....	365
1.8.6.6 Connected component.....	372
1.8.6.7 Topological sorting.....	374
1.8.6.8 Linear regression.....	377
1.8.6.9 Count triangles.....	381
1.8.6.10 GraphLoader.....	383
1.9 Java SDK.....	389
1.10 Java sandbox limits.....	389
1.11 Volume lifecycle management.....	392
1.11.1 Overview.....	392
1.11.2 Volume lifecycle operations.....	393
1.12 Spark on MaxCompute.....	393
1.12.1 Overview.....	393
1.12.2 Project resources.....	394
1.12.3 Environment settings.....	394
1.12.3.1 Decompress the Spark on MaxCompute release package.....	394
1.12.3.2 Set environment variables.....	394
1.12.3.3 Configure Spark-defaults.conf.....	395
1.12.4 Quick start.....	396
1.12.5 Common cases.....	397
1.12.5.1 WordCount example.....	397
1.12.5.2 OSS access example.....	398
1.12.5.3 MaxCompute table read/write example.....	399
1.12.5.4 MaxCompute Table Spark-SQL example.....	401
1.12.5.5 MaxCompute self-developed Console mode example.....	402
1.12.5.6 MaxCompute Table PySpark example.....	403
1.12.5.7 Mllib example.....	404
1.12.5.8 PySpark interactive execution example.....	405
1.12.5.9 Spark-shell interactive execution example (read tables).....	405
1.12.5.10 Spark-shell interactive execution example (Mllib and OSS read/write).....	405
1.12.5.11 SparkR interactive execution example.....	406
1.12.5.12 GraphX-PageRank example.....	407
1.12.5.13 Spark Streaming - NetworkWordCount example.....	408
1.12.6 Maven dependencies.....	409
1.12.7 Special notes.....	410
1.12.7.1 Streaming tasks.....	410

1.12.7.2 Tracking Url.....	410
1.12.8 APIs supported by Spark.....	411
1.12.8.1 Spark Shell.....	411
1.12.8.2 Spark R.....	411
1.12.8.3 Spark SQL.....	412
1.12.8.4 Spark JDBC.....	412
1.12.9 Spark dynamic resource allocation.....	412
1.13 Elasticsearch on Maxcompute.....	414
1.13.1 Overview.....	414
1.13.2 Workflow.....	415
1.13.2.1 Overview.....	415
1.13.2.2 Distributed retrieval workflow.....	416
1.13.2.3 Full-text retrieval process.....	417
1.13.2.4 Authentication process.....	418
1.13.3 Quick start.....	418
1.13.4 Support for Elasticsearch applications.....	420
1.13.4.1 ElasticSearch typical practice.....	420
1.13.4.2 Elasticsearch on MaxCompute support for VPC.....	420
1.13.5 Special notes.....	421
1.13.5.1 Find the Elasticsearch service domain name.....	421
1.13.5.2 Import table data from MaxCompute to Elasticsearch.....	422
1.14 Non-structured data access and processing (integrated computing scenarios).....	423
1.14.1 Overview.....	423
1.14.2 Internal data sources.....	424
1.14.2.1 OSS data source.....	424
1.14.2.1.1 Preface.....	424
1.14.2.1.2 Use the built-in extractor to read OSS data.....	424
1.14.2.1.2.1 Overview.....	424
1.14.2.1.2.2 Create an external table.....	425
1.14.2.1.2.3 Query an external table.....	425
1.14.2.1.3 Custom extractors.....	426
1.14.2.1.3.1 Overview.....	426
1.14.2.1.3.2 Define StorageHandler.....	426
1.14.2.1.3.3 Define an extractor.....	427
1.14.2.1.3.4 Compile and package code.....	428
1.14.2.1.3.5 Create an external table.....	428
1.14.2.1.3.6 Query an external table.....	429
1.14.2.1.4 Advanced usage.....	429
1.14.2.1.4.1 Use a custom extractor to read external unstructured data.....	429
1.14.2.1.5 Data partitions.....	432
1.14.2.1.5.1 Overview.....	432
1.14.2.1.5.2 Standard organization method and path format of partition data in OSS.....	432

1.14.2.1.5.3 Custom path of partition data in OSS.....	434
1.14.2.1.5.4 Access fully-customized non-partitioned data subsets.....	435
1.14.2.1.6 Output OSS data.....	435
1.14.2.1.6.1 Create an external table.....	435
1.14.2.1.6.2 Write data to a TSV text file by using an INSERT statement on an external table.....	436
1.14.2.1.6.3 Write data to an unstructured file by using an INSERT statement on an external table.....	438
1.14.2.1.6.4 Migrate data between different storage media with MaxCompute.....	438
1.14.2.2 Table Store data source.....	439
1.14.2.2.1 Preface.....	439
1.14.2.2.2 MaxCompute reads and computes data in Table Store.....	440
1.14.2.2.2.1 Prerequisites and assumptions.....	440
1.14.2.2.2.2 Create an external table.....	440
1.14.2.2.2.3 Access Table Store data through an external table.....	441
1.14.2.2.3 Write data from MaxCompute to Table Store.....	442
1.14.2.3 AnalyticDB data source.....	443
1.14.2.3.1 Overview.....	443
1.14.2.3.2 Write data to AnalyticDB.....	443
1.14.2.3.2.1 Create an external table.....	443
1.14.2.3.2.2 Write and query data.....	444
1.14.2.3.3 Read data from AnalyticDB.....	444
1.14.2.4 RDS data source.....	445
1.14.2.4.1 Overview.....	445
1.14.2.4.2 Write data to RDS.....	445
1.14.2.4.2.1 Create an external table.....	445
1.14.2.4.2.2 Write and query data.....	446
1.14.2.4.3 Read data from RDS.....	446
1.14.2.5 HDFS data source (Alibaba Cloud).....	446
1.14.2.5.1 Overview.....	446
1.14.2.5.2 Data processing for common tables.....	447
1.14.2.5.2.1 Write data to HDFS.....	447
1.14.2.5.2.2 Read data from HDFS.....	448
1.14.2.5.3 Data processing for partitioned tables.....	448
1.14.2.6 TDDL data source.....	449
1.14.2.6.1 Overview.....	449
1.14.2.6.2 Prerequisites.....	450
1.14.2.6.3 Create a TDDL external table.....	450
1.14.2.6.3.1 Syntax.....	450
1.14.2.6.3.2 Example.....	454
1.14.2.6.4 Read data from an external table.....	455
1.14.2.6.5 Write data to an external table in the append mode.....	456
1.14.3 External data sources.....	456
1.14.3.1 HDFS data source (open-source).....	456

1.14.3.1.1 Overview.....	456
1.14.3.1.2 Write data to HDFS.....	457
1.14.3.1.2.1 Create an external table.....	457
1.14.3.1.2.2 Write and query data.....	457
1.14.3.1.3 Read data from HDFS.....	457
1.14.3.2 MongoDB data source.....	458
1.14.3.2.1 Overview.....	458
1.14.3.2.2 Prerequisites.....	458
1.14.3.2.3 Write data to MongoDB.....	459
1.14.3.2.3.1 Create an external table.....	459
1.14.3.2.3.2 Write and query data.....	460
1.14.3.2.4 Read data from MongoDB.....	460
1.14.3.3 HBase data source.....	460
1.14.3.3.1 Overview.....	460
1.14.3.3.2 Write data to HBase.....	461
1.14.3.3.2.1 Create an external table.....	461
1.14.3.3.2.2 Write and query data.....	461
1.14.3.3.3 Read data from HBase.....	461
1.15 Unstructured data access and processing (inside MaxCompute).....	462
1.15.1 Overview.....	462
1.15.2 Create a volume external table.....	463
1.15.2.1 Syntax.....	463
1.15.2.2 Use the built-in StorageHandler to create an external table...	464
1.15.2.3 Use a custom StorageHandler to create a table.....	465
1.15.3 Access a volume external table.....	466
1.16 Security solution.....	466
1.16.1 Target users.....	466
1.16.2 Quick start.....	466
1.16.3 User authentication.....	470
1.16.4 Project user and authorization management.....	470
1.16.4.1 Overview.....	470
1.16.4.2 User management.....	470
1.16.4.3 Role management.....	471
1.16.4.4 ACL authorization actions.....	472
1.16.4.5 View permissions.....	475
1.16.5 Cross-project resource sharing.....	476
1.16.5.1 Overview.....	476
1.16.5.2 Package usage.....	477
1.16.5.2.1 Operations for package creators.....	477
1.16.5.2.2 Operations for package users.....	478
1.16.6 Project protection.....	479
1.16.6.1 Overview.....	479
1.16.6.2 Data protection.....	480
1.16.6.3 Data export methods when project protection is enabled.....	480
1.16.6.4 Resource sharing and data protection.....	482

1.16.7 Project security configuration.....	483
1.16.8 Authorization policies.....	483
1.16.8.1 Policy overview.....	483
1.16.8.2 Policy-related terms.....	486
1.16.8.3 Access policy structure.....	487
1.16.8.3.1 Overview.....	487
1.16.8.3.2 Authorization statement structure.....	487
1.16.8.3.3 Conditional block structure.....	488
1.16.8.3.4 Conditional action type.....	488
1.16.8.3.5 Conditional keywords.....	489
1.16.8.4 Access policy norm.....	490
1.16.8.4.1 Principal naming convention.....	490
1.16.8.4.2 Resource naming convention.....	491
1.16.8.4.3 Action naming.....	492
1.16.8.4.4 Condition keys naming.....	492
1.16.8.4.5 Access policy example.....	493
1.16.8.5 Differences between policy authorization and ACL authorization.....	493
1.16.8.6 Application limits.....	494
1.16.9 Collection of security statements.....	495
1.16.9.1 Project security configuration.....	495
1.16.9.2 Project permission management.....	496
1.16.9.3 Package-based resource sharing.....	498
1.17 Frequently-used tools.....	499
1.17.1 MaxCompute console.....	499
1.17.1.1 Usage notes.....	499
1.17.1.2 Install the client.....	499
1.17.1.3 Configuration description.....	500
1.17.2 Eclipse development plugin.....	504
1.17.2.1 Install Eclipse.....	504
1.17.2.2 Create a project.....	507
1.17.2.2.1 Method 1.....	507
1.17.2.2.2 Method 2.....	510
1.17.2.3 MapReduce running example.....	513
1.17.2.3.1 Quickly run a WordCount example.....	513
1.17.2.3.2 Run a custom MapReduce program.....	516
1.17.2.4 UDF development and running example.....	530
1.17.2.4.1 Local debug UDF programs.....	530
1.17.2.4.1.1 Run a UDF from the menu bar.....	530
1.17.2.4.1.2 Use the right-click shortcut menu to quickly run a UDF... ..	533
1.17.2.4.2 Run a UDF program.....	536
1.17.2.5 Graph running example.....	539
1.18 MaxCompute FAQ.....	543
2 DataWorks.....	548
2.1 What is DataWorks?.....	548

2.2 Planning and preparation.....	551
2.2.1 Planning and preparation.....	551
2.2.2 Workspace types.....	552
2.3 Quick start.....	553
2.3.1 Log on to the DataWorks console.....	553
2.3.2 Create a DataWorks workspace.....	556
2.3.3 Create a workflow.....	557
2.3.4 Configure monitoring policies.....	560
2.3.5 Create a resource.....	560
2.3.6 Import local files to MaxCompute.....	561
2.3.7 Use Spark.....	562
2.3.8 Submit and publish nodes.....	564
2.4 Data analytics.....	564
2.4.1 Solution.....	564
2.4.2 SQL coding guidelines and specifications.....	566
2.4.3 Business flows.....	571
2.4.3.1 Description.....	571
2.4.3.2 Resource.....	573
2.4.3.3 Create a function.....	575
2.4.4 Console features.....	577
2.4.4.1 Wizard.....	577
2.4.4.2 Version.....	579
2.4.4.3 Structure.....	580
2.4.4.4 Lineage.....	582
2.4.5 Node types.....	583
2.4.5.1 Node types.....	583
2.4.5.2 Data synchronization node.....	584
2.4.5.3 ODPS SQL nodes.....	585
2.4.5.4 ODPS Spark node.....	588
2.4.5.5 ODPS MR nodes.....	589
2.4.5.6 PyODPS nodes.....	591
2.4.5.7 Shell nodes.....	593
2.4.5.8 SQL component nodes.....	595
2.4.5.9 Virtual nodes.....	597
2.4.5.10 Cross-tenant collaboration node.....	598
2.4.5.11 Assignment node.....	599
2.4.5.12 Branch node.....	602
2.4.5.13 MERGE node.....	605
2.4.5.14 do-while node.....	607
2.4.5.15 Custom node type.....	612
2.4.5.15.1 Overview.....	612
2.4.5.15.2 Create a custom wrapper.....	614
2.4.5.15.3 Create a custom node type.....	616
2.4.6 Schedule.....	618
2.4.6.1 Basic attributes.....	618

2.4.6.2 Parameter configuration.....	619
2.4.6.3 Schedule.....	631
2.4.6.4 Dependencies.....	641
2.4.6.5 Resource type.....	646
2.4.7 Manage configurations.....	646
2.4.7.1 Configuration Center.....	646
2.4.7.2 Configuration Center.....	646
2.4.7.3 Project Configuration.....	648
2.4.7.4 Templates.....	649
2.4.7.5 Theme Management.....	649
2.4.7.6 Table Levels.....	650
2.4.8 Deploy.....	650
2.4.8.1 Publish nodes.....	650
2.4.8.2 Clone nodes across workspaces.....	652
2.4.9 Ad-hoc business flows.....	652
2.4.9.1 Description.....	652
2.4.9.2 Functions.....	654
2.4.9.3 Resources.....	655
2.4.9.4 Tables.....	656
2.4.10 Ad-hoc nodes.....	661
2.4.10.1 ODPS SQL nodes.....	661
2.4.10.2 PyODPS nodes.....	662
2.4.10.3 Ad-hoc data synchronization nodes.....	664
2.4.10.4 ODPS MR nodes.....	667
2.4.10.5 SQL script template.....	669
2.4.10.6 Zero-load node.....	672
2.4.10.7 Shell nodes.....	673
2.4.11 Configure parameters for ad-hoc tasks.....	674
2.4.11.1 Basic Information.....	674
2.4.11.2 Configure parameters for ad-hoc nodes.....	676
2.4.12 Components.....	682
2.4.12.1 Create components.....	682
2.4.12.2 Use components.....	686
2.4.13 Query.....	687
2.4.14 Runtime Log.....	689
2.4.15 Public Tables.....	689
2.4.16 Tables.....	690
2.4.17 Functions.....	694
2.4.18 Recycle bin.....	694
2.4.19 Editor keyboard shortcuts.....	694
2.4.20 Use EMR in DataWorks.....	697
2.5 Administration.....	700
2.5.1 Overview.....	700
2.5.2 Permissions.....	700
2.5.2.1 Role permissions.....	700

2.5.2.2 Developers.....	706
2.5.2.3 Deployment expert.....	706
2.5.2.4 Administration expert.....	706
2.5.2.5 Workspace administrator.....	707
2.5.3 O&M Overview.....	707
2.5.4 Task List.....	707
2.5.4.1 Recurring tasks.....	707
2.5.4.2 Ad-hoc tasks.....	708
2.5.5 Task O&M.....	708
2.5.6 Monitor.....	710
2.5.6.1 Overview.....	710
2.5.6.2 Feature description.....	712
2.5.6.2.1 Baseline alert and event alert.....	712
2.5.6.2.2 Custom alert trigger.....	714
2.5.6.3 Instructions.....	716
2.5.6.3.1 Baseline instances.....	716
2.5.6.3.2 Events.....	717
2.5.6.3.3 Alert triggers.....	717
2.5.6.3.4 Alerts.....	718
2.5.6.4 FAQ related to the Monitor module.....	718
2.5.6.4.1 Why was my alert reported to someone else?.....	718
2.5.6.4.2 What can I do if I do not want to receive alerts for unimportant nodes?.....	719
2.5.6.4.3 Why is no alert reported for a baseline break?.....	719
2.5.6.4.4 Can I disable DataWorks from reporting an alert for a node that slows down?.....	719
2.5.6.4.5 Why did I fail to receive an alert for an error node?.....	720
2.5.6.4.6 What can I do if I receive an alert at night?.....	720
2.6 Organization management.....	720
2.6.1 Project management.....	720
2.6.1.1 Description.....	720
2.6.1.2 Create a workspace.....	720
2.6.2 Member management.....	721
2.6.3 Resource groups.....	722
2.6.3.1 About scheduling resources.....	722
2.6.3.2 Create a scheduling resource.....	722
2.6.3.3 Change the workspace of scheduling resources.....	723
2.6.3.4 Manage servers.....	724
2.6.4 Compute engine.....	725
2.6.4.1 Configure the compute engine.....	725
2.7 Project Management.....	726
2.7.1 Configure a workspace.....	726
2.7.1.1 Basic property settings.....	726
2.7.1.2 Compute engine.....	727
2.7.2 Member management.....	727

2.7.3 Permission management.....	728
2.7.4 MaxCompute management.....	734
2.7.4.1 Basic settings.....	734
2.7.4.2 Customize user roles.....	735
2.8 Data Integration.....	735
2.8.1 Data Integration.....	735
2.8.1.1 Overview.....	735
2.8.1.2 Basic concepts.....	738
2.8.2 Data sources.....	739
2.8.2.1 Supported data sources.....	739
2.8.2.2 Data transmission.....	742
2.8.2.3 Test data store connectivity.....	743
2.8.2.4 Add a DataHub connection.....	748
2.8.2.5 Add a DM connection.....	749
2.8.2.6 Add FTP data sources.....	751
2.8.2.7 Add HDFS data sources.....	753
2.8.2.8 Add LogHub data sources.....	754
2.8.2.9 Add MaxCompute data sources.....	755
2.8.2.10 Add Memcached data sources.....	756
2.8.2.11 Add MySQL data sources.....	757
2.8.2.12 Add Oracle data sources.....	760
2.8.2.13 Add OSS data sources.....	762
2.8.2.14 Add a Table Store connection.....	763
2.8.2.15 Add a PostgreSQL connection.....	764
2.8.2.16 Add Redis data sources.....	767
2.8.2.17 Add a MongoDB connection.....	769
2.8.3 Configure data synchronization tasks.....	773
2.8.3.1 Configure a data synchronization node by using the codeless UI.....	773
2.8.3.2 Configure a data synchronization node by using the code editor.....	777
2.8.3.3 Configure the reader.....	782
2.8.3.3.1 Configure the HBase reader.....	782
2.8.3.3.2 Configure the HDFS reader.....	790
2.8.3.3.3 Configure MaxCompute Reader.....	800
2.8.3.3.4 Configure MongoDB Reader.....	806
2.8.3.3.5 Configure the DB2 reader.....	811
2.8.3.3.6 Configure the MySQL reader.....	817
2.8.3.3.7 Configure Oracle Reader.....	824
2.8.3.3.8 Configure the OSS reader.....	833
2.8.3.3.9 Configure FTP Reader.....	840
2.8.3.3.10 Configure the OTS reader.....	848
2.8.3.3.11 Configure PostgreSQL Reader.....	854
2.8.3.3.12 Configure the LogHub reader.....	863
2.8.3.3.13 Configure the OTSReader-Internal reader.....	871

2.8.3.3.14 Configure the OTSStream reader.....	880
2.8.3.3.15 Configure the RDBMS reader.....	887
2.8.3.3.16 Configuring the StreamCompute reader.....	894
2.8.3.3.17 Configure Elasticsearch Reader.....	897
2.8.3.4 Configure the writer.....	901
2.8.3.4.1 Configure the DataHub writer.....	901
2.8.3.4.2 Configure the DB2 writer.....	904
2.8.3.4.3 Configure the FTP writer.....	907
2.8.3.4.4 Configure HBase Writer.....	912
2.8.3.4.5 Configure the HBase11xsql writer.....	918
2.8.3.4.6 Configure the HDFS writer.....	921
2.8.3.4.7 Configure the MaxCompute writer.....	930
2.8.3.4.8 Configure the Memcache (OCS) writer.....	938
2.8.3.4.9 Configure MongoDB Writer.....	942
2.8.3.4.10 Configure MySQL Writer.....	947
2.8.3.4.11 Configure Oracle Writer.....	953
2.8.3.4.12 Configure the OSS writer.....	959
2.8.3.4.13 Configure the PostgreSQL writer.....	965
2.8.3.4.14 Configure the Redis writer.....	969
2.8.3.4.15 Configure Elasticsearch Writer.....	974
2.8.3.4.16 Configure LogHub Writer.....	981
2.8.3.4.17 Configure the OpenSearch writer.....	983
2.8.3.4.18 Configure the Table Store (OTS) writer.....	988
2.8.3.4.19 Configure RDBMS Writer.....	994
2.8.3.4.20 Configure the Stream writer.....	999
2.8.3.5 Optimize synchronization performance.....	1000
2.8.4 Full-database migration.....	1005
2.8.4.1 Overview.....	1005
2.8.4.2 Migrate a MySQL database.....	1007
2.8.4.3 Migrate an Oracle database.....	1009
2.8.5 Best practices.....	1011
2.8.5.1 Synchronize data when the source or destination is deployed on a private network.....	1011
2.8.5.2 Data integration when the networks of both data sources at the source and destination ends are disconnected.....	1015
2.8.5.3 Incremental data synchronization.....	1019
2.8.6 FAQs.....	1024
2.8.6.1 What can I do if the status of the node is Pending (Resources)?	1024
2.8.6.2 RDS data synchronization fails.....	1025
2.8.6.3 How do I troubleshoot data integration issues?.....	1026
2.8.6.4 Data synchronization task failure when the column name of the synchronized table is a keyword.....	1043
2.8.6.5 Customize a table name for the data synchronization task.....	1044
2.8.6.6 The specified encoding is incorrect.....	1045

2.8.6.7 The specified data types are supported for full database migration.....	1046
2.9 Data Quality.....	1047
2.9.1 Overview.....	1047
2.9.2 Features.....	1048
2.9.2.1 Dashboard.....	1048
2.9.2.2 My Subscriptions.....	1049
2.9.2.3 Rules.....	1049
2.9.2.4 Search by Node.....	1054
2.9.3 User guide.....	1055
2.9.3.1 MaxCompute monitoring.....	1055
2.9.3.2 DataHub monitoring.....	1062
2.10 Realtime Analysis.....	1067
2.10.1 Apply for joining a workspace.....	1067
2.10.2 Apply for data access permissions.....	1068
2.10.3 Ad hoc query.....	1068
2.10.4 Personal tables.....	1068
2.11 Data Service.....	1069
2.11.1 Overview.....	1069
2.11.2 Terms.....	1070
2.11.3 Create an API.....	1070
2.11.3.1 Configure data sources.....	1072
2.11.3.2 Generate APIs in wizard mode.....	1072
2.11.3.3 Create an API by specifying scripts.....	1075
2.11.4 Register an API.....	1078
2.11.5 Test APIs.....	1081
2.11.6 Delete APIs.....	1081
2.11.7 Publish APIs.....	1082
2.11.8 Call APIs.....	1083
2.11.9 FAQ.....	1084
2.12 Data Protection.....	1085
2.12.1 Overview.....	1085
2.12.2 Services.....	1085
2.12.3 Access Data Protection.....	1086
2.12.4 Configure rules for defining sensitive data.....	1087
2.12.5 View the distribution of data.....	1088
2.12.6 View the information about data activities.....	1088
2.12.7 View the information about data export.....	1088
2.12.8 Manage the data security levels.....	1089
2.12.9 Manage data that is incorrectly detected.....	1089
2.12.10 Customize de-identification rules.....	1090
2.13 Data Asset Management.....	1091
2.13.1 Overview.....	1091
2.13.2 Asset administrator (View data asset information).....	1092
2.13.3 Asset user.....	1092

2.13.4 Asset manager.....	1093
2.13.5 Create a data asset category.....	1093
2.13.6 Manage tables.....	1094
2.13.7 Departments.....	1095
2.14 Security Center.....	1095
2.14.1 Overview.....	1095
2.14.2 My Permissions.....	1096
2.14.3 Authorizations.....	1099
2.14.4 Approval Center.....	1100
2.14.5 FAQ.....	1101
2.15 DataOS API.....	1103
2.16 App Studio.....	1107
2.16.1 Overview.....	1107
2.16.2 Get started with App Studio.....	1109
2.16.3 Navigation pane.....	1118
2.16.3.1 View and manage projects.....	1118
2.16.3.2 View and manage templates.....	1118
2.16.4 Project management.....	1118
2.16.5 Code editing.....	1119
2.16.5.1 Overview.....	1119
2.16.5.2 Generate code snippets.....	1121
2.16.5.3 Run UT.....	1122
2.16.5.4 Search all content of files.....	1123
2.16.6 Debugging.....	1123
2.16.6.1 Configuration and startup.....	1123
2.16.6.2 Online debugging.....	1124
2.16.6.3 Breakpoint types.....	1125
2.16.6.4 Breakpoint operations.....	1126
2.16.6.5 Terminal.....	1128
2.16.6.6 Hot code replacement.....	1128
2.16.7 WYSIWYG designer.....	1129
2.16.7.1 Basic usage.....	1129
2.16.7.2 Code mode.....	1132
2.16.7.3 DSL syntax.....	1133
2.16.7.4 Global data flow.....	1134
2.16.7.5 Save, preview, run, and hot code replacement.....	1136
2.16.7.6 Navigation configuration.....	1136
2.17 Data Map.....	1137
2.17.1 Overview.....	1137
2.17.2 View the overall information.....	1138
2.17.3 Manage data.....	1139
2.17.4 View table details.....	1142
2.17.5 Manage permissions.....	1146
2.17.6 Apply for data permissions.....	1146
2.17.7 Manage configurations.....	1148

3 Realtime Compute.....	1150
3.1 What is Realtime Compute?.....	1150
3.2 Quick start.....	1151
3.2.1 Log on to the Realtime Compute console.....	1151
3.2.2 Real-time security monitoring.....	1152
3.2.2.1 Overview.....	1152
3.2.2.2 Preparations.....	1153
3.2.2.3 Development.....	1154
3.2.2.4 Administration.....	1156
3.2.3 Frequently used words.....	1156
3.2.3.1 Overview.....	1156
3.2.3.2 Code development.....	1157
3.2.3.3 Code debugging.....	1159
3.2.3.4 Administration.....	1161
3.2.4 Big screen service for the Tmall Double Eleven Global Shopping Festival.....	1161
3.2.4.1 Overview.....	1161
3.2.4.2 Scenario description.....	1162
3.2.4.3 Preparations.....	1163
3.2.4.4 Register a data store.....	1163
3.2.4.5 Development.....	1164
3.2.4.6 Administration.....	1165
3.3 Project management.....	1166
3.4 Data storage.....	1168
3.4.1 Overview.....	1168
3.4.2 Overview.....	1168
3.4.2.1 Overview.....	1168
3.4.2.2 Types.....	1169
3.4.2.3 Scenarios.....	1169
3.4.3 Register a DataHub project.....	1171
3.4.4 Register a Log Service project.....	1173
3.4.5 Register a Table Store instance.....	1175
3.4.6 Register an RDS instance.....	1177
3.5 Data development.....	1181
3.5.1 Create a job.....	1181
3.5.2 Development.....	1182
3.5.2.1 SQL code assistance.....	1182
3.5.2.2 SQL code version management.....	1182
3.5.2.3 Data storage management.....	1183
3.5.3 Debug the code.....	1184
3.5.4 Publish the SQL file for a job.....	1187
4 Quick BI.....	1188
4.1 What is Quick BI?.....	1188
4.2 Log on to the Quick BI console.....	1189

4.3 Data modeling.....	1190
4.3.1 Overview.....	1190
4.3.2 Data sources.....	1190
4.3.2.1 Overview.....	1190
4.3.2.2 Cloud data sources.....	1190
4.3.2.2.1 View whitelisted IP addresses.....	1190
4.3.2.2.2 MaxCompute.....	1192
4.3.2.2.3 MySQL.....	1194
4.3.2.2.4 SQL Server.....	1196
4.3.2.2.5 AnalyticDB.....	1198
4.3.2.2.6 HybridDB For MYSQL.....	1200
4.3.2.2.7 AnalyticDB for PostgreSQL.....	1202
4.3.2.2.8 PostgreSQL.....	1204
4.3.2.2.9 PPAS.....	1205
4.3.2.2.10 Hive.....	1207
4.3.2.2.11 Data Lake Analytics.....	1208
4.3.2.2.12 DRDS.....	1209
4.3.2.3 User-created data sources.....	1210
4.3.2.3.1 MySQL.....	1210
4.3.2.3.2 SQL Server.....	1212
4.3.2.3.3 PostgreSQL.....	1214
4.3.2.3.4 Oracle.....	1216
4.3.2.3.5 Hive.....	1217
4.3.2.3.6 Vertica.....	1218
4.3.2.3.7 IBM DB2 LUW.....	1220
4.3.2.3.8 SAP IQ (Sybase IQ).....	1221
4.3.2.3.9 SAP HANA.....	1222
4.3.2.4 Data sources.....	1223
4.3.2.5 Add a data source.....	1223
4.3.2.6 Edit a data source.....	1224
4.3.2.7 Delete a data source.....	1225
4.3.2.8 Search for a data source.....	1225
4.3.2.9 Search for tables under a data source.....	1226
4.3.2.10 Query table details.....	1227
4.3.3 Datasets.....	1227
4.3.3.1 Overview.....	1227
4.3.3.2 Create datasets.....	1227
4.3.3.2.1 Create datasets from data sources.....	1227
4.3.3.2.2 Upload CSV files to create datasets.....	1228
4.3.3.2.3 Use custom SQL statements to create datasets.....	1229
4.3.3.3 Specify a method to name dimensions and measures.....	1230
4.3.3.4 Edit a dataset.....	1231
4.3.3.4.1 Edit a dimension.....	1231
4.3.3.4.2 Edit a measure.....	1233
4.3.3.4.3 Toolbar.....	1235

4.3.3.4.4 Preview data.....	1237
4.3.3.4.5 Table join and examples.....	1237
4.3.3.4.6 Calculated fields.....	1240
4.3.3.4.6.1 Overview.....	1240
4.3.3.4.6.2 Rules for using calculated fields.....	1241
4.3.3.4.6.3 Types of calculated measures.....	1241
4.3.3.4.6.4 Examples of using a calculated field.....	1242
4.3.3.4.6.5 Add a calculated field.....	1243
4.3.3.4.7 Grouping fields.....	1246
4.3.3.5 Rename a dataset.....	1247
4.3.3.6 Search for a dataset.....	1248
4.3.3.7 Transfer a dataset.....	1248
4.3.3.8 Create a dataset folder.....	1249
4.3.3.9 Rename a dataset folder.....	1250
4.3.3.10 Delete a dataset.....	1250
4.3.3.11 Set row-level permissions.....	1252
4.4 Dashboards.....	1252
4.4.1 Dashboard overview.....	1252
4.4.1.1 Dashboard features.....	1252
4.4.1.2 Chart types and scenarios.....	1253
4.4.1.3 Data elements of a chart.....	1254
4.4.2 Access a dashboard.....	1260
4.4.3 Areas of a dashboard.....	1260
4.4.3.1 Overview.....	1260
4.4.3.2 Dataset selection area.....	1262
4.4.3.2.1 Switch datasets.....	1262
4.4.3.2.2 Search for a dimension or measure.....	1262
4.4.3.3 Dashboard graphic design area.....	1263
4.4.3.3.1 Select fields.....	1263
4.4.3.3.2 Color legend.....	1265
4.4.3.3.3 Sorting.....	1267
4.4.3.3.4 Filter by field.....	1269
4.4.3.3.5 Filter interaction.....	1270
4.4.3.4 Dashboard display area.....	1273
4.4.3.4.1 Overview.....	1273
4.4.3.4.2 Toolbar.....	1273
4.4.3.4.3 Adjust chart position.....	1273
4.4.3.4.4 View chart data.....	1273
4.4.3.4.5 Change chart types.....	1274
4.4.3.4.6 Add to favorites.....	1277
4.4.3.4.7 Delete a chart.....	1277
4.4.3.4.8 Widgets.....	1278
4.4.3.4.8.1 Overview.....	1278
4.4.3.4.8.2 Filter bar.....	1278
4.4.3.4.8.3 Text area.....	1290

4.4.3.4.8.4 IFrame.....	1291
4.4.3.4.8.5 Tab.....	1291
4.4.3.4.8.6 Image.....	1293
4.4.4 Create a dashboard.....	1294
4.4.4.1 Line charts.....	1294
4.4.4.2 Area charts.....	1297
4.4.4.3 Vertical bar charts.....	1299
4.4.4.4 Horizontal bar charts.....	1307
4.4.4.5 Progress bar charts.....	1310
4.4.4.6 Pie charts.....	1311
4.4.4.7 Bubble maps.....	1313
4.4.4.8 Colored maps.....	1316
4.4.4.9 Geo bubble maps.....	1319
4.4.4.10 Geo maps.....	1320
4.4.4.11 Cross tables.....	1323
4.4.4.12 Pivot tables.....	1329
4.4.4.13 Gauges.....	1330
4.4.4.14 Radar charts.....	1334
4.4.4.15 Scatter charts.....	1336
4.4.4.16 Bubble charts.....	1338
4.4.4.17 Funnel charts.....	1339
4.4.4.18 Kanban.....	1341
4.4.4.19 Treemaps.....	1343
4.4.4.20 Polar diagrams.....	1346
4.4.4.21 Word clouds.....	1349
4.4.4.22 Tornado-leaned funnel charts.....	1350
4.4.4.23 Hierarchy charts.....	1356
Scenario 1: Compare the order quantity of different products in provinces in different regions.....	1356
Scenario 2: Display the average profit of different products in different municipalities.....	1362
4.4.4.24 Flow analysis charts.....	1367
4.4.5 Full Screen mode.....	1373
4.4.6 Search for a dashboard.....	1379
4.4.7 Create a dashboard folder.....	1379
4.4.8 Rename a dashboard folder.....	1380
4.4.9 Share a dashboard.....	1380
4.4.10 Make a dashboard public.....	1381
4.5 Workbooks.....	1382
4.5.1 Overview.....	1382
4.5.2 Create a workbook.....	1383
4.5.3 Switch datasets.....	1384
4.5.4 Search for a dimension or measure.....	1385
4.5.5 Fonts.....	1386
4.5.6 Alignment modes.....	1387

4.5.7 Text and number formats.....	1388
4.5.8 Style, cell, and pane settings.....	1389
4.5.9 Insert images, hyperlinks, and drop-down boxes.....	1390
4.5.10 Set a table style.....	1392
4.5.11 Set conditional formatting.....	1393
4.5.12 Search for a workbook.....	1396
4.5.13 Create a workbook folder.....	1396
4.5.14 Rename a workbook folder.....	1397
4.5.15 Share a workbook.....	1397
4.5.16 Make a workbook public.....	1398
4.6 BI portals.....	1399
4.6.1 Overview.....	1399
4.6.2 Create a BI portal.....	1399
4.6.3 Page settings.....	1399
4.6.4 Menu settings.....	1402
4.7 Organization.....	1404
4.7.1 Overview.....	1404
4.7.2 Create an organization.....	1405
4.7.3 Modify organization information.....	1405
4.7.4 Leave an organization.....	1407
4.7.5 Add a member.....	1407
Obtain the Apsara Stack tenant account.....	1410
Obtain the Apsara Stack RAM user account.....	1411
Add an Apsara Stack tenant account.....	1411
Add a RAM user account.....	1412
Add multiple members at the same time.....	1413
4.7.6 Manage member tags.....	1414
4.7.7 Edit a member.....	1416
4.7.8 Remove a member.....	1417
4.7.9 Query the workspace that a user belongs to.....	1418
4.7.10 Search for members.....	1418
4.7.11 Workspaces.....	1419
4.7.11.1 Overview.....	1419
4.7.11.2 What is a workspace?.....	1419
4.7.11.3 Differences between the personal workspace and a workspace.....	1423
4.7.12 Create a workspace.....	1423
4.7.13 Edit workspace information.....	1424
4.7.14 Leave a workspace.....	1425
4.7.15 Transfer a workspace to another owner.....	1426
4.7.16 Delete a workspace.....	1427
4.7.17 Add a member to a workspace.....	1427
4.7.18 Edit settings of a workspace member.....	1428
4.7.19 Search for a member in a workspace.....	1428
4.7.20 Delete a member from a workspace.....	1429

4.8 Permissions.....	1429
4.8.1 Overview.....	1429
4.8.2 Data objects.....	1430
4.8.3 Row-level permission management.....	1431
4.8.4 Configure BI portal menu permissions.....	1435
4.8.5 Share data objects in the personal workspace.....	1437
4.8.6 Share a data object in a workspace.....	1437
4.8.7 Publish data objects that are stored in a personal workspace...	1438
4.8.8 Make a data object public in a workspace.....	1438
4.9 Report statistics.....	1439
4.9.1 Usage statistics.....	1439
4.9.2 Lineage analysis.....	1439
4.9.3 Access statistics.....	1440
5 DataQ - Smart Tag Service.....	1442
5.1 What is DataQ - Smart Tag Service?.....	1442
5.2 Quick start.....	1443
5.2.1 Log on to the DataQ console.....	1443
5.2.2 Create workspaces.....	1445
5.2.3 Create private tags.....	1446
5.2.4 Create shared tags.....	1449
5.2.5 Apply for using shared tags.....	1450
5.3 Analysis APIs.....	1452
5.3.1 Overview.....	1452
5.3.2 APIs.....	1453
5.3.3 API factory.....	1453
5.3.4 Fast search.....	1454
5.4 Dashboards.....	1455
5.4.1 Overview.....	1455
5.4.2 Manage datasets.....	1455
5.4.3 Manage reports.....	1456
5.4.4 Report permissions.....	1457
5.5 Tag factory.....	1458
5.5.1 Overview.....	1458
5.5.2 Tag schemes.....	1458
5.5.2.1 Overview.....	1458
5.5.2.2 Create tag schemes.....	1458
5.5.2.3 Submit tag schemes.....	1460
5.5.2.4 Run tag schemes.....	1461
5.5.3 Tag tasks.....	1462
5.6 Tag sync.....	1462
5.6.1 Overview.....	1462
5.6.2 Sync schedules.....	1463
5.6.3 Sync tasks.....	1464
5.6.4 Task O&M.....	1464
5.7 Other features.....	1465

5.7.1 Homepage.....	1465
5.7.2 Tag center.....	1465
5.7.2.1 Overview.....	1465
5.7.2.2 The overview chart.....	1467
5.7.2.3 Tag warehouse.....	1467
5.7.2.4 My tags.....	1468
5.7.2.5 Tag models.....	1469
5.7.2.6 Model views.....	1473
5.7.2.7 Schemas.....	1473
5.7.2.8 Data import.....	1474
5.7.3 Tag Apps.....	1475
5.7.4 Manage workspaces.....	1475
6 E-MapReduce (EMR).....	1483
6.1 What is EMR?.....	1483
6.2 Introduction.....	1483
6.2.1 Prerequisites.....	1483
6.2.2 Introduction.....	1483
6.2.2.1 Software configuration.....	1483
6.2.2.2 Software environment.....	1483
6.2.2.3 Supported components.....	1484
6.2.2.4 Introduction to components.....	1484
6.2.3 Introduction.....	1486
6.2.3.1 Hardware architecture.....	1486
6.2.3.2 Cluster architecture.....	1487
6.2.3.3 Hardware requirements.....	1487
6.2.4 Introduction.....	1489
6.2.4.1 Deployment.....	1489
6.2.4.2 Deployment modes.....	1489
6.2.4.3 Supported services.....	1489
6.3 Introduction.....	1492
6.3.1 User operations.....	1492
6.3.2 Create a RAM role.....	1492
6.3.3 Log on to the E-MapReduce console.....	1492
6.3.4 Gateway.....	1493
6.3.4.1 Gateway.....	1493
6.3.4.2 Log on to an EMR gateway.....	1494
6.3.4.3 Service environment.....	1494
6.3.4.4 Security authentication.....	1495
6.3.5 Jobs.....	1496
6.3.5.1 Jobs.....	1496
6.3.5.2 Hadoop MapReduce Job Configuration.....	1496
6.3.5.3 Submit a Spark job.....	1496
6.3.5.4 Submit a Hive job.....	1497
6.3.5.5 Oozie.....	1497
6.3.5.5.1 Oozie.....	1497

6.3.5.5.2 Schedule a Hadoop MapReduce job.....	1498
6.3.5.5.3 Schedule a Spark job.....	1498
6.3.5.5.4 Schedule a Hive job.....	1499
6.3.6 Workflow.....	1499
6.3.6.1 Workflow.....	1499
6.3.6.2 Manage projects.....	1499
6.3.6.3 Edit a job.....	1502
6.3.6.4 Design a workflow.....	1504
6.3.7 Component endpoints.....	1506
6.4 Cluster O&M.....	1507
6.4.1 Cluster O&M.....	1507
6.4.2 Component O&M.....	1507
6.4.3 Basic operation environment and software environment O&M..	1508
7 Graph Analytics.....	1509
7.1 What is Graph Analytics?.....	1509
7.2 Quick Start.....	1509
7.2.1 Log on to Administration Console of Graph Analytics.....	1509
7.2.2 Create data sources.....	1511
7.2.3 Create OLEP models for tables.....	1514
7.2.4 Add OLEP table columns.....	1527
7.2.5 Configure object properties and business parameters.....	1529
7.2.6 Configure link properties and business parameters.....	1533
7.2.7 Configure event property parameters.....	1538
7.2.8 Log on to Analytics Workbench.....	1544
7.2.9 Create analyses.....	1545
7.2.10 View analyses.....	1548
7.3 Source tables.....	1548
7.3.1 Data sources.....	1548
7.3.1.1 Create data sources.....	1548
7.3.1.2 View data sources.....	1551
7.3.1.3 Modify a data source.....	1552
7.3.1.4 Delete data sources.....	1552
7.3.2 OLEP tables.....	1553
7.3.2.1 Create OLEP models for tables.....	1553
7.3.2.2 View an OLEP table.....	1566
7.3.2.3 Edit OLEP tables.....	1567
7.3.2.4 Remove OLEP tables.....	1568
7.3.3 OLEP table columns.....	1569
7.3.3.1 Add OLEP table columns.....	1569
7.3.3.2 Edit OLEP table columns.....	1571
7.3.3.3 Remove OLEP table columns.....	1573
7.4 Dictionaries.....	1575
7.4.1 Create a dictionary.....	1575
7.4.2 Modify a dictionary.....	1577
7.4.3 Delete a dictionary.....	1578

7.5 Object information.....	1580
7.5.1 Object groups.....	1580
7.5.1.1 Create an object group.....	1580
7.5.1.2 View object groups and objects.....	1581
7.5.1.3 Modify object groups and objects.....	1582
7.5.1.4 Delete object groups and objects.....	1585
7.5.2 Objects.....	1586
7.5.2.1 Create an object.....	1586
7.5.2.2 Configure object properties and business parameters.....	1587
7.5.2.3 Enable and disable an object.....	1591
7.5.2.4 Modify an object.....	1592
7.5.2.5 Delete an object.....	1593
7.6 Link information.....	1594
7.6.1 Link groups and links.....	1594
7.6.1.1 Create a link group.....	1594
7.6.1.2 View links and link groups.....	1595
7.6.1.3 Modify a link or link group.....	1596
7.6.1.4 Delete a link or link group.....	1600
7.6.2 First-degree links.....	1601
7.6.2.1 Create a first-degree link.....	1601
7.6.2.2 Configure link properties and business parameters.....	1603
7.6.3 Create a second-degree link.....	1608
7.6.4 Create a multi-degree link.....	1613
7.7 Event information.....	1616
7.7.1 Event groups.....	1616
7.7.1.1 Create an event group.....	1616
7.7.1.2 View an event group.....	1617
7.7.1.3 Modify an event group.....	1617
7.7.1.4 Delete an event group.....	1618
7.7.2 Events.....	1619
7.7.2.1 Create an event.....	1619
7.7.2.2 Configure event property parameters.....	1620
7.7.2.3 Enable and disable an event.....	1626
7.7.2.4 View an event.....	1626
7.7.2.5 Modify an event.....	1627
7.7.2.6 Delete an event.....	1628
7.8 View the business graph.....	1629
7.9 Advanced configurations.....	1629
7.9.1 Manage a system model.....	1629
7.9.2 Configure a search item.....	1633
7.9.3 System settings.....	1637
7.9.3.1 Configure components.....	1637
7.9.3.2 Technical parameters.....	1638
7.9.3.2.1 Path analysis settings.....	1638
7.9.3.2.2 Quick extension settings.....	1639

7.9.3.2.3 Maximum node settings.....	1640
7.9.3.3 Business parameters.....	1641
7.9.3.3.1 Add double-click link settings.....	1641
7.9.3.3.2 Double-click-disabled object settings.....	1643
7.9.3.3.3 Object grouping settings.....	1643
7.9.3.3.4 Configure lineage analysis.....	1645
7.9.3.3.5 Intimacy measurement settings.....	1646
7.9.3.3.6 Redirect URL settings.....	1647
7.9.3.4 Object icons.....	1647
7.9.3.4.1 Upload an object icon.....	1647
7.9.3.4.2 Modify an object icon.....	1647
7.9.3.4.3 Delete an object icon.....	1648
7.9.4 System labels.....	1648
7.9.4.1 Create a group.....	1648
7.9.4.2 Create a system label.....	1649
7.9.4.3 Modify a system label.....	1652
7.9.4.4 Delete a system label.....	1653
7.9.5 System operations and maintenance.....	1653
7.9.5.1 Audit logs.....	1653
7.9.6 View server clusters.....	1654
7.10 Import data.....	1655
7.10.1 Model list.....	1655
7.10.1.1 Model overview.....	1655
7.10.1.2 View models.....	1655
7.10.1.3 Create models.....	1656
7.10.1.4 Modify model names.....	1658
7.10.1.5 Download a model.....	1659
7.10.1.6 Delete a model.....	1659
7.10.2 Import data.....	1660
7.10.3 Data list.....	1662
7.10.3.1 View data.....	1662
7.10.3.2 Edit a data name.....	1663
7.10.3.3 Import data to Graph.....	1663
7.10.3.4 Delete data.....	1663
7.11 Search.....	1664
7.11.1 Search.....	1664
7.11.2 Simple search.....	1665
7.11.3 Advanced search.....	1666
7.11.4 View and analyze search results.....	1668
7.12 Graph.....	1670
7.12.1 Graph.....	1670
7.12.2 Analysis types.....	1671
7.12.3 Create analyses.....	1672
7.12.4 Add a node.....	1675
7.12.5 Delete nodes, links, and events.....	1677

7.12.6 Link extension.....	1678
7.12.7 Graphic operations.....	1680
7.12.7.1 Move canvases.....	1680
7.12.7.2 Zoom in and zoom out canvases.....	1681
7.12.7.3 Undo and redo operations.....	1682
7.12.7.4 View thumbnails.....	1683
7.12.7.5 Right-click menu.....	1684
7.12.8 Analyze.....	1689
7.12.8.1 Group Analysis.....	1689
7.12.8.2 Common neighbor analysis.....	1690
7.12.8.3 Lineage analysis.....	1692
7.12.8.4 Path analysis.....	1695
7.12.8.5 Backbone analysis.....	1698
7.12.8.6 Intimacy measurements.....	1700
7.12.9 Lock or unlock nodes.....	1701
7.12.10 Network analysis.....	1703
7.12.11 Closed-loop mining.....	1707
7.12.12 Layouts.....	1708
7.12.13 Flag and unflag nodes.....	1712
7.12.14 Labels.....	1712
7.12.14.1 Label types.....	1712
7.12.14.2 User labels.....	1713
7.12.14.3 Add user labels.....	1714
7.12.14.4 View labels.....	1716
7.12.14.5 Click likes and delete likes.....	1717
7.12.14.6 Edit user labels.....	1718
7.12.14.7 Delete user labels.....	1719
7.12.15 Save analysis.....	1720
7.12.16 Print graph areas.....	1720
7.12.17 Share analyses.....	1721
7.12.18 Behavior chronology.....	1723
7.12.18.1 Details.....	1723
7.12.18.2 Behavior analysis.....	1724
7.12.18.3 Chronology analysis.....	1725
7.12.19 Property statistics.....	1727
7.12.19.1 Details.....	1727
7.12.19.2 Statistics.....	1729
7.12.19.3 Property information.....	1734
7.12.19.4 Secondary filtering.....	1736
7.13 File Center.....	1738
7.13.1 View and manage all analyses.....	1738
7.13.2 View and manage your files.....	1739
7.13.3 My shared items.....	1741
7.13.3.1 Overview.....	1741
7.13.3.2 View and manage shared files.....	1742

7.13.3.3 Edit a shared file.....	1744
7.13.3.4 Publish a version.....	1745
7.13.3.5 Automatically merge files.....	1748
7.13.4 View and manage received shared files.....	1749
7.14 Intelligent Network.....	1750
7.14.1 Intelligent Network overview.....	1751
7.14.2 Patterns.....	1752
7.14.2.1 Create patterns.....	1752
7.14.2.2 View patterns.....	1758
7.14.2.3 Modify patterns.....	1760
7.14.2.4 Set private patterns to public patterns.....	1763
7.14.2.5 Delete a pattern.....	1764
7.14.3 Tasks.....	1764
7.14.3.1 Create a task.....	1764
7.14.3.2 Check the task.....	1767
7.14.3.3 Modify a task.....	1769
7.14.3.4 Execute the task and view the result.....	1772
7.14.3.5 Set a private task as a public task.....	1775
7.14.3.6 Delete a task.....	1776
7.15 Examples.....	1777
7.15.1 Tax industry case studies.....	1777
7.16 FAQ.....	1785
8 Machine Learning Platform for AI.....	1788
8.1 What is machine learning?.....	1788
8.2 Quick start.....	1789
8.2.1 Overview.....	1789
8.2.2 Log on to the Apsara Stack Machine Learning Platform for AI console.....	1790
8.2.3 Data preparation.....	1791
8.2.4 Data preprocessing.....	1792
8.2.5 Data visualization.....	1793
8.2.6 Algorithm modeling.....	1793
8.2.7 Model prediction evaluation.....	1794
8.2.8 Online model service (must be activated separately).....	1795
8.2.8.1 Deploy an online model service.....	1795
8.2.8.2 Create a service.....	1795
8.2.8.3 Add an existing service version.....	1796
8.2.8.4 Create a blue-green deployment.....	1796
8.2.9 DataWorks task scheduling.....	1797
8.3 Components.....	1798
8.3.1 Overview.....	1798
8.3.2 Data source and target.....	1799
8.3.3 Data preprocessing.....	1799
8.3.3.1 Sampling and filtering.....	1799
8.3.3.1.1 Random sampling.....	1799

8.3.3.1.2 Weighted sampling.....	1801
8.3.3.1.3 Filtering and mapping.....	1804
8.3.3.1.4 Stratified sampling.....	1805
8.3.3.2 Data merge.....	1807
8.3.3.2.1 Join.....	1807
8.3.3.2.2 Merge columns.....	1808
8.3.3.2.3 Merge rows (UNION).....	1809
8.3.3.3 Others.....	1811
8.3.3.3.1 Add ID column.....	1811
8.3.3.3.2 Split.....	1812
8.3.3.3.3 Missing value imputation.....	1814
8.3.3.3.4 Normalization.....	1820
8.3.3.3.5 Standardization.....	1822
8.3.3.3.6 KV to Table.....	1828
8.3.3.3.7 Table to KV.....	1832
8.3.4 Feature engineering.....	1836
8.3.4.1 Feature transformation.....	1836
8.3.4.1.1 PCA.....	1836
8.3.4.2 Feature importance evaluation.....	1838
8.3.4.2.1 Linear model feature importance.....	1838
8.3.4.2.2 Random forest feature importance.....	1840
8.3.5 Statistical analysis.....	1841
8.3.5.1 Data pivoting.....	1841
8.3.5.2 Whole table statistics.....	1846
8.3.5.3 Correlation coefficient matrix.....	1847
8.3.5.4 Covariance.....	1850
8.3.5.5 Empirical probability density chart.....	1852
8.3.5.6 Chi-square goodness of fit test.....	1858
8.3.5.7 Chi-square test of independence.....	1860
8.3.5.8 Scatter plot.....	1863
8.3.5.9 Two-sample T-test.....	1868
8.3.5.10 One-sample T-test.....	1872
8.3.5.11 Lorenz curve.....	1873
8.3.5.12 Normality test.....	1877
8.3.5.13 Percentile.....	1881
8.3.5.14 Pearson coefficient.....	1881
8.3.5.15 Histogram.....	1883
8.3.6 Machine learning.....	1883
8.3.6.1 Binary classification.....	1883
8.3.6.1.1 GBDT binary classification.....	1883
8.3.6.1.2 Linear SVM.....	1886
8.3.6.1.3 Logistic regression for binary classification.....	1888
8.3.6.1.4 PS-SMART binary classification.....	1890
8.3.6.2 Multiclass classification.....	1903
8.3.6.2.1 KNN.....	1903

8.3.6.2.2 Logistic regression for multiclass classification.....	1907
8.3.6.2.3 Random forest.....	1909
8.3.6.2.4 Naive Bayes.....	1912
8.3.6.2.5 PS-SMART multiclass classification.....	1913
8.3.6.3 K-means clustering.....	1927
8.3.6.4 Regression.....	1931
8.3.6.4.1 GBDT regression.....	1931
8.3.6.4.2 Linear regression.....	1936
8.3.6.4.3 PS linear regression.....	1943
8.3.6.4.4 PS-SMART regression.....	1952
8.3.6.5 Collaborative filtering (etrec).....	1967
8.3.6.6 Evaluation.....	1970
8.3.6.6.1 Regression model evaluation.....	1970
8.3.6.6.2 Clustering model evaluation.....	1972
8.3.6.6.3 Binary classification evaluation.....	1976
8.3.6.6.4 Confusion matrix.....	1978
8.3.6.6.5 Multiclass classification evaluation.....	1979
8.3.6.7 Prediction.....	1980
8.3.7 Deep learning (must be activated separately).....	1982
8.3.7.1 Activate deep learning.....	1982
8.3.7.2 Read OSS buckets.....	1982
8.3.7.3 TensorFlow 1.4.....	1983
8.3.8 Time series.....	1988
8.3.8.1 x13_arima.....	1988
8.3.8.2 x13_auto_arima.....	1995
8.3.9 Text analysis.....	2004
8.3.9.1 Word splitting.....	2004
8.3.9.2 Deprecated word filtering.....	2007
8.3.9.3 String similarity.....	2009
8.3.9.4 Convert row, column, and value to KV pair.....	2012
8.3.9.5 String similarity - Top N.....	2015
8.3.9.6 N-gram counting.....	2020
8.3.9.7 Text summarization.....	2022
8.3.9.8 Keyword extraction.....	2025
8.3.9.9 Sentence splitting.....	2031
8.3.9.10 Semantic vector distance.....	2033
8.3.9.11 Document similarity.....	2035
8.3.9.12 PMI.....	2038
8.3.9.13 Word frequency statistics.....	2043
8.3.9.14 TF-IDF.....	2045
8.3.9.15 PLDA.....	2047
8.3.9.16 Word2Vec.....	2050
8.3.10 Network analysis.....	2053
8.3.10.1 K-core.....	2053
8.3.10.2 Single-source shortest path.....	2056

8.3.10.3 PageRank.....	2059
8.3.10.4 Label propagation clustering.....	2062
8.3.10.5 Label propagation classification.....	2066
8.3.10.6 Modularity.....	2069
8.3.10.7 Maximum connected subgraph.....	2071
8.3.10.8 Vertex clustering coefficient.....	2073
8.3.10.9 Edge clustering coefficient.....	2076
8.3.10.10 Counting triangle.....	2078
8.3.10.11 Tree depth.....	2081
8.3.11 Tools.....	2083
8.3.11.1 SQL script.....	2083
8.4 Automatic parameter tuning with AutoML.....	2084
8.4.1 Automatic parameter tuning with AutoML.....	2084
8.4.2 Parameter tuning methods.....	2086
8.5 Terms and acronyms.....	2089
8.5.1 Terms.....	2089
8.5.2 Acronyms.....	2089
8.6 FAQ.....	2090
9 Dataphin.....	2092
9.1 What is Dataphin?.....	2092
9.2 Limits.....	2093
9.3 Quick start.....	2095
9.3.1 Instructions for the system administrator.....	2095
9.3.2 Instructions for quick start.....	2098
9.3.3 Log on to the Dataphin console.....	2098
9.3.4 Management Center.....	2099
9.3.5 Data warehouse planning.....	2099
9.3.6 Data ingestion.....	2100
9.3.7 Data modeling and development.....	2100
9.3.8 Scheduling center.....	2101
9.3.9 Data assets.....	2101
9.3.10 Theme-based data services.....	2101
9.4 Management Center.....	2102
9.4.1 Overview.....	2102
9.4.2 Initialize metadata.....	2102
9.4.3 Set the computing engine type.....	2103
9.4.4 Member management.....	2104
9.5 Data warehouse planning.....	2106
9.5.1 Overview.....	2106
9.5.2 Business units.....	2106
9.5.3 Global objects.....	2108
9.5.4 Project management.....	2108
9.5.5 Physical data sources.....	2111
9.5.6 Computing engine sources.....	2113
9.6 Data ingestion.....	2114

9.6.1 Manage sync task folders.....	2115
9.6.2 Manage sync tasks.....	2118
9.6.3 Create a sync task.....	2119
9.6.4 Configure a sync task.....	2120
9.6.5 Configure the scheduling policy.....	2126
9.6.6 Run sync tasks.....	2131
9.7 Data modeling and development.....	2131
9.7.1 Overview.....	2131
9.7.2 Data standardization: Dimensions.....	2132
9.7.3 Data standardization: Business processes.....	2136
9.7.4 Logical tables: Logical dimension tables.....	2137
9.7.5 Logical tables: Logical fact tables.....	2139
9.7.6 Logical tables: Scheduling configuration.....	2141
9.7.7 Data standardization: Atomic metrics and business filters.....	2143
9.7.8 Data standardization: Derived metrics.....	2146
9.8 Data distillation.....	2147
9.8.1 Instructions for data distillation.....	2147
9.8.2 Behavior Engine.....	2151
9.8.2.1 Define behavioral elements.....	2151
9.8.2.2 Define and design behavior rules.....	2155
9.8.2.3 View behaviors.....	2161
9.8.3 Tag Engine.....	2163
9.8.3.1 Define tags.....	2163
9.8.3.2 View tags.....	2167
9.8.4 Manage data distillation tasks.....	2168
9.9 Scheduling center.....	2171
9.9.1 Tasks.....	2171
9.9.2 Instances.....	2173
9.9.3 Logical table tasks.....	2175
9.9.4 Logical table task instances.....	2176
9.10 Monitoring and alerting.....	2178
9.10.1 Task monitoring settings.....	2178
9.10.1.1 Create task monitoring settings.....	2178
9.10.1.2 Manage task monitoring settings.....	2179
9.10.2 Alert records.....	2181
9.11 Data assets.....	2183
9.11.1 Overview.....	2183
9.11.2 Asset overview.....	2183
9.11.2.1 Global mode.....	2183
9.11.2.2 Flow mode.....	2185
9.11.2.3 Structure mode.....	2186
9.11.3 Map.....	2187
9.11.4 Administration.....	2188
9.12 Theme-based data service.....	2192
9.12.1 Ad hoc query.....	2192

10 Elasticsearch.....	2195
10.1 What is Elasticsearch?.....	2195
10.2 Planning and preparation.....	2196
10.2.1 Data types.....	2196
10.2.2 Connect to Elasticsearch.....	2196
10.3 Quick start.....	2199
10.3.1 Create a VPC.....	2199
10.3.2 Create a security group.....	2202
10.3.3 Create an ECS instance.....	2205
10.3.4 Log on to the Elasticsearch console.....	2207
10.3.5 Create an Elasticsearch instance.....	2208
10.3.6 Connect to an Elasticsearch instance.....	2212
10.4 Instance management.....	2213
10.4.1 Kibana console.....	2213
10.4.2 Restart an instance.....	2213
10.4.3 Refresh.....	2214
10.4.4 Basic information.....	2215
10.4.5 Elasticsearch cluster configurations.....	2216
10.4.5.1 Word splitting.....	2216
10.4.5.2 Configure synonyms.....	2217
10.4.5.3 YML configuration.....	2226
10.4.5.3.1 Configuration parameters.....	2227
10.4.5.3.2 Custom remote reindexing (whitelisting).....	2230
10.4.6 Cluster upgrade.....	2232
10.4.7 Plug-ins.....	2233
10.4.8 Security.....	2240
10.4.9 Snapshots.....	2241
10.4.9.1 Auto snapshot.....	2241
10.4.9.2 View snapshot status.....	2243
10.4.9.3 Restore data from snapshots.....	2245
10.5 Document operations.....	2249
10.5.1 Create a document.....	2249
10.5.2 Update a document.....	2250
10.5.3 Retrieve a document.....	2250
10.5.4 Search documents.....	2251
10.5.5 Complex searches.....	2252
10.5.6 Delete documents.....	2252
10.6 Snapshots and restoration.....	2253
10.6.1 Create a repository.....	2253
10.6.2 Obtain repository information.....	2254
10.6.3 Migrate a snapshot.....	2254
10.6.4 Create a snapshot for all running indexes.....	2255
10.6.5 Create a snapshot for a specific index.....	2255
10.6.6 Obtain snapshot information.....	2256
10.6.7 Delete a snapshot.....	2256

10.6.8 Monitor snapshot progress.....	2257
10.6.9 Cancel a snapshot.....	2259
10.6.10 Restore data from a snapshot.....	2260
10.6.11 Monitor the restoration operation.....	2261
10.6.12 Cancel a restoration task.....	2262
10.7 Elasticsearch test.....	2262
10.7.1 Use curl to connect to Elasticsearch through port 9200.....	2263
10.7.2 Use Python to connect to Elasticsearch through port 9200.....	2263
10.7.3 Use Java REST client to connect to Elasticsearch through port 9200.....	2264
11 DataHub.....	2266
11.1 What is DataHub?.....	2266
11.2 Limits.....	2267
11.3 Quick Start.....	2268
11.3.1 Overview.....	2268
11.3.2 Log on to the DataHub console.....	2269
11.3.3 Create projects.....	2270
11.3.4 Create a topic.....	2271
11.3.5 Upload local files.....	2271
11.3.6 Sample data.....	2272
11.4 Access Control.....	2273
11.4.1 Overview.....	2273
11.4.2 DataHub resources in RAM.....	2273
11.4.3 API.....	2274
11.4.4 Conditions.....	2275
11.4.5 Sample RAM authorization policy content.....	2276
11.4.5.1 AliyunDataHubFullAccess.....	2276
11.4.5.2 AliyunDataHubReadOnlyAccess.....	2276
11.5 Data Acquisition.....	2277
11.5.1 Overview.....	2277
11.5.2 Fluentd.....	2277
11.5.3 Logstash.....	2281
11.5.4 Oracle GoldenGate.....	2286
11.6 Data Archive.....	2297
11.6.1 Overview.....	2297
11.6.2 Archive to MaxCompute.....	2297
11.6.2.1 Create a DataConnector.....	2297
11.6.2.2 View archive details.....	2299
11.7 Performance monitoring.....	2299

1 MaxCompute

1.1 What is MaxCompute?

MaxCompute is a data processing platform developed by Alibaba Group to process large volumes of data. MaxCompute provides channels for upload and download, a range of computing and analysis features including SQL and MapReduce, and comprehensive security solutions.

MaxCompute is used to store and compute large volumes of structured data. It provides warehouse solutions for large amounts of data, as well as big data analysis and modeling services.

As data collection techniques are becoming increasingly diverse and comprehensive, industries are amassing larger volumes of data. The scale of data collection has increased from 100 GB to over 1 PB, far exceeding the processing capabilities of traditional software. Analysis tasks for large volumes of data require distributed computing instead of reliance on a single server. However, distributed computing models require skilled data analysts to be properly implemented. To use a distributed model, data analysts must understand their business needs and the underlying computing model.

MaxCompute is designed to provide an intuitive approach to analyze and process large amounts of data without the need for distributed computing knowledge. You can perform big data analysis without distributed computing knowledge. MaxCompute is widely implemented within Alibaba's businesses for scenarios such as data warehousing and BI analysis for large Internet enterprises, website log analysis, e-commerce transaction analysis, and mining user data for characteristics and interests.

MaxCompute provides the following features:

- **Data channel**
 - **Tunnel: provides highly-concurrent offline data upload and download services**
 - **MaxCompute Tunnel enables you to upload or download a large volume of**

data to or from MaxCompute. You must use a Java programming API to access MaxCompute Tunnel.

- **DataHub:** provides real-time data upload and download services. Data uploaded through DataHub is available immediately, while data uploaded through MaxCompute Tunnel is not.
- **Computing and analysis**
 - **SQL:** MaxCompute stores data in tables, and provides SQL query capabilities to manipulate the data. MaxCompute can be used as traditional database software, but is far more powerful and capable of processing petabytes of data. MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE. The SQL syntax used in MaxCompute is different from that of Oracle and MySQL. SQL statements from other database engines cannot be seamlessly integrated into MaxCompute. The fastest MaxCompute SQL is capable of responding in the scale of seconds, and is not capable of responding in milliseconds. SQL is easy to learn. You can transfer prior experience with database operation to MaxCompute SQL without the need to understand distributed computing.
 - **MapReduce:** First proposed by Google, MapReduce is a distributed data processing model that has gained extensive attention and been used in a wide range of business scenarios. This topic briefly describes the MapReduce modeling. You must have basic knowledge of distributed computing and relevant programming experience before using MapReduce. MapReduce provides a Java programming API.
 - **Graph:** It is an iteration-oriented graph computing framework provided in MaxCompute. Graph computing jobs use graphs to build models. A graph is a collection of vertices and edges that has values. You can edit and evolve a graph through iteration to obtain the final result. Typical applications include *PageRank*, *single source shortest path (SSSP) algorithm*, and *K-means clustering algorithm*.
 - **Non-structured data access and processing (integrated computing scenarios):** MaxCompute SQL cannot directly process external data (such as non-structured data from OSS). Data must be imported to MaxCompute tables through relevant tools before computation. The MaxCompute team introduces

the non-structured data processing framework to the MaxCompute system architecture to resolve this problem.

MaxCompute can process the following data sources by creating external tables:

- Internal data sources: OSS, Table Store, AnalyticDB, RDS, DFS, and TDDL.
- External data sources: HDFS, MongoDB, and Hbase.
- Non-structured data access and processing (inside MaxCompute): By reading and writing volumes, MaxCompute can store non-structured data, which otherwise must be stored in an external storage system.
- Spark on MaxCompute: It is a big data analytics engine designed by Alibaba Cloud to provide big data processing capabilities for Alibaba, government agencies, and enterprises. For more information, see [Spark on MaxCompute](#).
- Elasticsearch on MaxCompute: Elasticsearch on MaxCompute is an enterprise-class full-text retrieval system developed by Alibaba Cloud to retrieve large volumes of data. It provides near-real-time (NRT) search performance for government agencies and enterprises. For more information, see [Elasticsearch on MaxCompute](#).
- SDK: It is a toolkit provided for developers. For more information, see [Java SDK](#).
- Security solution: MaxCompute provides powerful security services to guarantee user data security.

1.2 Usage notes

You can selectively read topics in this document based on your requirements. This topic provides reading suggestions in the document based on user skill level.

Beginners

If you are a beginner, we recommend that you read the following topics:

- What is MaxCompute: The topic provides a general introduction of MaxCompute and its core features. You can obtain general knowledge about MaxCompute.
- Quick start: The topic provides step-by-step examples and instructions on how to perform basic MaxCompute operations, such as installing and configuring the client, creating tables, granting permissions, importing and exporting data, running SQL tasks, running user-defined functions (UDFs), and running MapReduce.

- **Basic concepts and common commands:** The topic introduces the basic concepts of MaxCompute and common MaxCompute commands. This topic helps familiarize yourself with MaxCompute operations.
- **Tools:** The topic describes how to download, configure, and use common MaxCompute tools to perform data analysis.

Data analysts

If you are a data analyst, we recommend that you read the following topics:

MaxCompute SQL: The topic describes how to query and analyze large amounts of data stored in MaxCompute. This topic covers the following operations:

- **Execute DDL statements CREATE, DROP, and ALERT to manage tables and partitions.**
- **Execute SELECT statements to select records in a table, and execute WHERE clauses to view the records that meet a specified filtering condition.**
- **Associate two tables through an EQUIJOIN operation.**
- **Execute GROUP BY statements to aggregate columns.**
- **Execute INSERT OVERVIEW or INSERT INTO statements to insert results into another table.**
- **Use built-in functions and UDFs to complete a variety of computations.**
- **Use UDTs to reference classes or objects of third-party languages in SQL statements to obtain data or call methods.**
- **Use UDJs to implement flexible cross-table and multi-table custom operations , and reduce the operations on the underlying details of the distributed system through MapReduce.**
- **Use the Select Transform feature to simplify the reference of script code.**
- **Collect table statistics and configure table lifecycles.**
- **Use regular expressions.**

Developers

If you are an experienced developer with basic understanding of distributed computing and need to perform data analysis that cannot be implemented with SQL, we recommend that you read the following topics on advanced MaxCompute functional modules:

- **MapReduce** is a Java programming model provided by MaxCompute. You can use the API to write MapReduce programs and process MaxCompute data.
- **MaxCompute Graph** is a processing framework designed that iteratively computes and models graphs. A graph consists of vertices and edges, both of which contain values. MaxCompute Graph iteratively edits and evolves graphs to obtain analysis results.
- **Eclipse plugin** provides an IDE to help you complete development of MapReduce, UDFs, and Graph.
- **Java SDK** is a toolkit provided to developers.
- **MaxCompute Tunnel** allows you to perform batch upload and download operations on offline data to and from MaxCompute.

Project owners or administrators

If you are a project owner or administrator, we recommend that you read the following topics:

Security solution: This topic describes how to authorize users, enable cross-project resource sharing, configure project data protection, and configure authorization policies.

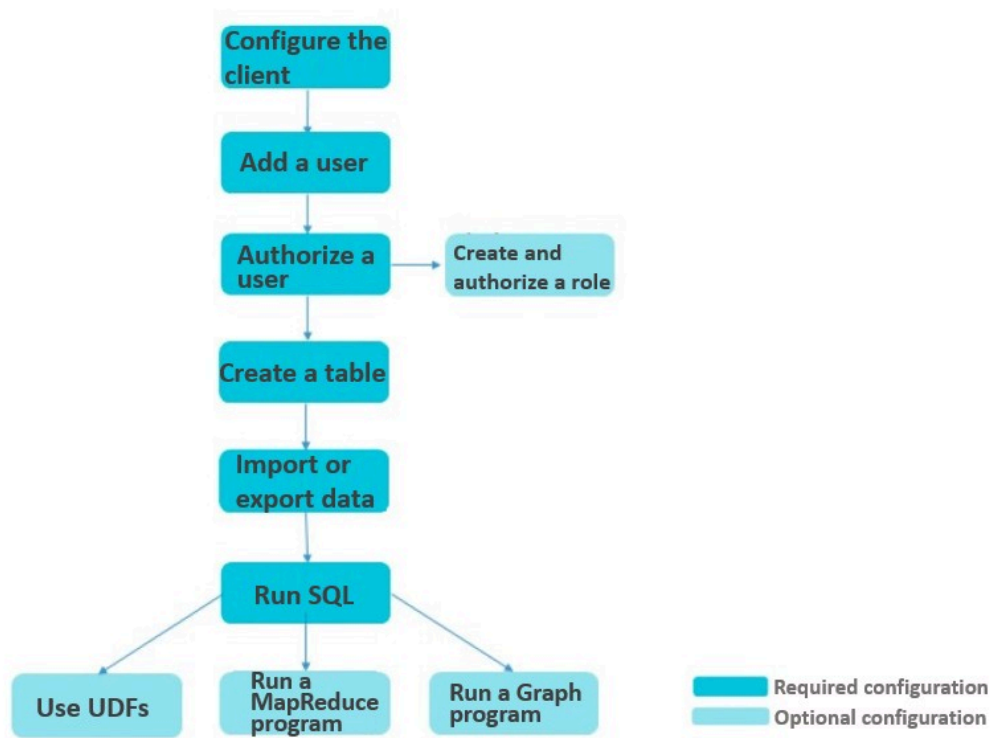
1.3 Quick start

1.3.1 Overview

This topic describes the operation process of MaxCompute. It aims to provide you with step-by-step instructions on basic MaxCompute features.

For a more detailed procedure, see [Figure 1-1: MaxCompute operation process](#).

Figure 1-1: MaxCompute operation process



- [Configure the client](#)

You must install and configure the [client](#) to access all MaxCompute functions.

- [Add a user](#)

Except for the project owner, all users must be manually added to a project and granted permissions before they can perform operations on the project.

- [Authorize a user](#)

After you add a user, you must authorize the user to perform operations on the project. A user can only perform operations on the project after the user is authorized.

- [Create and authorize a role](#)

It can be very time-consuming to individually authorize users if a project contains a large number of users. Project administrators can use roles to grant users a specified set of permissions. After you authorize a role, all users added to this role are granted the same permissions.

- [Create a table](#)

After you are added to a project and authorized, you can start to use MaxCompute. Table operations are the most basic operations in MaxCompute.

- [Import or export data](#)

You can use the SDK provided to import and export data to compile your own Java tools.

- [Run SQL](#)

This topic only describes the limits to a few common SQL statements. For more information about how to execute SQL statements, see [MaxCompute SQL](#).

- [Use UDFs](#)

MaxCompute provides three types of UDFs: UDFs, UDAFs, and UDTFs. These functions are collectively known as UDFs.

- [Run a MapReduce program](#)

After you install the MaxCompute client, you can run a MapReduce program.

- [Run a Graph program](#)

After you install the MaxCompute client, you can run a Graph program.

1.3.2 Configure the client

You must install and configure the [client](#) to use all of the features of MaxCompute.

Context

The client was developed in Java, so make sure that you have JRE 1.8 installed on your local PC. An Alibaba Cloud account is required to obtain the AccessKey ID and AccessKey Secret.



Note:

Before you configure the client, make sure that you have created a project, and obtained the AccessKey ID and AccessKey Secret.

Procedure

1. Download the [client](#) package to your local PC.
2. Extract the package to the folder where you want to store the client. The package contains the following four folders:

```
bin/  
conf/  
lib/  
plugins/
```

3. Edit the `odps_config.ini` file in the `conf` folder as follows:

```
project_name=my_project  
access_id=*****  
access_key=*****  
end_point= <MaxCompute endpoint>
```



Note:

- Set `access_id` and `access_key` to the AccessKey ID and AccessKey Secret of your Alibaba Cloud account.
 - `Project_name=my_project` specifies the project that you want to access. This is the default project that will be accessed each time you log on to the client. If this parameter is not configured, you must run the `use project_name` command after logging on to the client to access the project.
 - Set `end_point` to the service address of MaxCompute. The service address varies with the user account.
 - For more information about the client, see [MaxCompute client](#).
4. After the modifications, run the `odps` file in the `bin` directory (`./bin/odpscmd` in a Linux system or `./bin/odpscmd.bat` in a Windows system). You are now ready to execute SQL statements. An example is as follows:

```
create table tbl1(id bigint);  
insert overwrite table tbl1 select count(*) from tbl1;  
select 'welcome to MaxCompute!' from tbl1;
```



Note:

For information about more SQL statements, see [MaxCompute SQL](#).

1.3.3 Add and delete users

Other than the project owner, all other users must be added to a MaxCompute project and granted the corresponding permissions before they can perform any operations on the project. This topic describes how a project owner can add or delete users in a project.

If you are a project owner, we recommend that you read this topic in full. If you are a common user, we recommend that you submit an application to a project owner to join a project, and read the subsequent topics once you are added to the project.

Add users

Run the following command to add a user:

```
ADD USER <full_username>;
```

Run the following command on the client to add a user (bob@aliyun.com) to a project:

```
add user bob@aliyun.com;
```

If you are not sure whether the user is already in the project, run the following command to view the users in the project:

```
LIST USERS;
```



Note:

- After a user is added to a MaxCompute project, the user must be granted permissions by the project owner. Then, the user can perform operations authorized by the permissions.
- For more information about authorization, see [Grant and view permissions](#).

Delete users

Run the following command to delete a user:

```
REMOVE USER <full_username>;
```

Run the following command on the client to delete a user from a project:

```
remove user bob@aliyun.com;
```



Note:

- Before you delete a user, make sure that you have revoked all the permissions of the user. If you delete a user without revoking the permissions of the user, the permissions are retained. If the user is added to the project again, the user will have the permissions that were granted previously.
- For more information about how to add or delete users, see [Manage users in a project](#).

1.3.4 Grant and view permissions

1.3.4.1 Overview

After you add a user, you need to authorize the user. A user can only perform operations on the project after the user is authorized.

Authorization is a process of granting the permission to perform an operation (such as reading, writing, or viewing), on objects (such as tables, tasks, and resources) in MaxCompute.

This topic is intended for project administrators. If you are a regular MaxCompute user, verify that you have obtained the required permissions. You can quickly skim this topic.

MaxCompute provides two authorization mechanisms, [ACL authorization](#) and [policy authorization](#).

1.3.4.2 ACL authorization

This topic describes the commands for ACL authorization and provides examples.

ACL authorization in MaxCompute applies to the following objects: project, table, function, resource, instance, and task. Every object has different operation permissions. For more information, see [ACL authorization actions](#).

Command syntax:

```
GRANT privileges ON project_object TO project_subject
REVOKE privileges ON project_object FROM project_subject
privileges ::= action_item1, action_item2, ...
project_object ::= PROJECT project_name | TABLE schema_name |
INSTANCE inst_name | FUNCTION func_name |
RESOURCE res_name | JOB job_name
project_subject ::= USER full_username | ROLE role_name
```



Note:

You can skip the ROLE clause in the preceding command. It is described in the topics after this.

Example:

```
grant CreateTable on PROJECT $user_project_name to USER ALIYUN$bob@aliyun.com;
-- Grant bob@aliyun.com the permissions to create tables in the project named $user_project_name.
grant Describe on Table $user_table_name to USER ALIYUN$bob@aliyun.com;
;
-- Grant bob@aliyun.com the permissions to obtain information (Describe permission) in the table named $$user_table_name.
grant Execute on Function $user_function_name to USER ALIYUN$bob@aliyun.com;
-- Grant bob@aliyun.com the permissions to run the function named $user_function_name.
```

1.3.4.3 Policy authorization

This topic describes the commands for policy authorization and provides an example.

Policy authorization is a principal-based process. For more information, see [Authorization policies](#).

Command syntax:

```
GET POLICY;
PUT POLICY <policyFile>;
GET POLICY ON ROLE <roleName>;
PUT POLICY <policyFile> ON ROLE <roleName>;
```

Example:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "odps:*"
      ],
      "Resource": "acs:odps:*:projects/$user_project_name/tables/*",
      "Condition": {
        "StringEquals": {
          "odps:TaskType": [
            "DT"
          ]
        }
      }
    },
    {
      "Effect": "allow ",
      "Action": [
        "odps:List",
        "odps:Read",
        "odps:Describe",
        "odps:Select"
      ],
      "Resource": [
```

```
"acs:odps:*:projects/$user_project_name/tables/a",
"acs:odps:*:projects/$user_project_name"
],
"Condition": {
  "StringEquals": {
    "odps:TaskType": [
      "SQL"
    ]
  }
}
}
```

**Note:**

The preceding example disables the Tunnel feature of `$user_project_name`, and grants a user the permissions to perform list, read, describe, and select operations on the project and table `a` in the project.

1.3.4.4 View permissions

You can run a command to view user permissions in MaxCompute.

Run the following command to view the permissions of a user:

```
show grants for $user_name;
```

**Note:**

For more information about how to view user permissions, see [View permissions](#).

1.3.5 Create and authorize a role

If a project has a large number of users, the authorization process is time-consuming. Project administrators can use roles to categorize users with the same permissions. After you authorize a role, all users assigned with this role are granted the same permissions. This topic describes how to create a role and grant permissions to it.

One user can have multiple roles, and multiple users can have the same role.

Create a role

Run the following command to create a role:

```
CREATE ROLE <roleName>;
```

Example:

```
create role player;
```

Add a user to a role

Run the following command to add a user to a role:

```
GRANT <roleName> TO <full_username>;
```

Example:

```
grant player to bob@aliyun.com;
```

Delete a role

Run the following command to delete a role:

```
DROP ROLE <roleName>;
```

Example:

```
drop role player;
```



Note:

Before you delete a role, make sure that all users have been removed from the role.

Grant permissions to a role

The command for granting permissions to a role is similar to the command for granting permissions to a user. For more information about how to grant permissions to a user, see [Grant and view permissions](#). For more information about role authorization, see [Role management](#).

1.3.6 Create or delete a table

1.3.6.1 Create a table

This topic describes how to create a table.

Run the following command to create a table:

```
CREATE TABLE [IF NOT EXISTS] table_name
```

```
[(col_name data_type [COMMENT col_comment], ...)] [COMMENT table_comment]
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)] [
LIFECYCLE days]
[AS select_statement]
CREATE TABLE [IF NOT EXISTS] table_name
LIKE existing_table_name
```

Example:

```
create table test1 (key string);
-- Create a non-partitioned table.
create table test2 (key bigint) partitioned by (pt string, ds string
);
-- Create a partitioned table.
create table test3 (key boolean) partitioned by (pt string, ds string
) lifecycle 100;
-- Create a table with a lifecycle.
create table test4 like test3;
-- Table test3 has the same attributes (such as the field type and
partition type) as those of test4, except for lifecycle.
create table test5 as select * from test2;
-- Create table test5 without replicating the partition and lifecycle
information of test2 to it. Only data of test2 is copied to test5.
```

You can set partitions or lifecycles for MaxCompute tables. For more information about how to create a table, see [Create a table](#). For more information about how to modify partitions, see [Add a partition](#) and [Delete a partition](#). For more information about how to modify the lifecycle, see [Modify the lifecycle of a table](#).

1.3.6.2 Obtain table information

This topic describes how to obtain the table information.

Command syntax:

```
desc <table_name>;
```

Example:

```
desc test3;
-- Obtain the information about test3.
desc test4;
```

```
-- Obtain the information about test4.
```

1.3.6.3 Delete a table

This topic describes how to delete a table.

Run the following command to delete a table:

```
DROP TABLE [IF EXISTS] table_name;
```

Example:

```
drop table test2;
```



Note:

For more information, see [Delete a table](#).

1.3.7 Import or export data

You can compile your own Java tools by using the SDK provided by MaxCompute Tunnel to import or export data. For the sample code, see [Tunnel SDK examples](#).

1.3.8 Run SQL

1.3.8.1 Overview

This topic describes the limits to a few common SQL statements only. For more information about how to run SQL statements, see [MaxCompute SQL](#).

Note the following issues when using MaxCompute SQL:

- MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE.
- The SQL syntax of MaxCompute is different from that of Oracle or MySQL. You cannot seamlessly migrate SQL statements from other database engines to MaxCompute.
- MaxCompute SQL does not respond to queries in real time. It requires a few minutes to return query results, instead of seconds or milliseconds.

1.3.8.2 SELECT statement

This topic describes limits of the SELECT statement.

The following limits apply to the SELECT statement:

- The key of the GROUP BY statement can be the name of a column in the input table, or the expression composed of input table columns. However, it cannot be the output column of the SELECT statement.

```
select substr(col2, 2) from tbl group by substr(col2, 2);
-- Allowed: The key of the GROUP BY statement is an expression of
columns in the input table.
select col2 from tbl group by substr(col2, 2);
-- Not allowed: The key of the GROUP BY statement is not included in
columns of the SELECT statement.
select substr(col2, 2) as c from tbl group by c;
-- Not allowed: The key of the GROUP BY statement is the alias of a
column, or output column of the SELECT statement.
```

**Note:**

For SQL parsing, the GROUP BY operation is conducted before the SELECT operation, which means the GROUP BY statement can only use the column or expression of the input table as the key.

- DISTRIBUTE BY must be added in front of SORT BY.
- The key of ORDER BY/SORT BY/DISTRIBUTE BY must be the output column of SELECT statement, or the column alias.

```
select col2 as c from tbl order by col2 limit 100
-- Not allowed: The key of the ORDER BY statement is not the output
column of the SELECT statement, or the column alias.
select col2 from tbl order by col2 limit 100;
-- Allowed: If an output column of the SELECT statement does not
have an alias, the column name is used as the alias.
```

**Note:**

For SQL parsing, the ORDER BY, SORT BY, and DISTRIBUTE BY operations come after the SELECT operation. Therefore, they can only accept the output columns of the SELECT statement as the key.

For more information about the SELECT statement, see [SELECT statement](#).

1.3.8.3 INSERT statement

This topic describes the limits of the INSERT statement.

The following limits apply to INSERT statements:

- When an INSERT statement is used to insert data into a partition, the partition column cannot be included in the select list.

```
insert overwrite table sale_detail_insert partition (sale_date='2017', region='china') select shop_name, customer_id, total_price, sale_date, region from sale_detail;
-- An error is returned. The sale_date and region columns are partition columns and are not allowed in a INSERT statement for a static partition.
```

- When an INSERT statement is used to insert a dynamic partition, the dynamic partition column must be included in the select list.

```
insert overwrite table sale_detail_dypart partition (sale_date='2017', region) select shop_name, customer_id, total_price from sale_detail;
-- An error is returned. When a dynamic partition is specified for the insert, the dynamic partition columns must be included among the selected columns.
```

For more information about the INSERT statement, see [INSERT statement](#).

1.3.8.4 JOIN statements

This topic describes the limits of JOIN statements.

The following limits apply to the JOIN operation:

- MaxCompute SQL supports the following join operations: {LEFT OUTER|RIGHT OUTER|FULL > OUTER|INNER} JOIN.
- MaxCompute SQL supports a maximum of 16 parallel JOIN operations.
- The MAPJOIN function can be applied to up to 256 small tables at a time.

For more information about JOIN operations, see [JOIN statements](#).

1.3.8.5 Other limits

This topic describes the other application limits of MaxCompute SQL.

- MaxCompute SQL supports a maximum of 256 concurrent UNION operations.
- MaxCompute SQL supports a maximum of 256 concurrent INSERT OVERWRITE/ INTO operations.

1.3.9 Compile and use UDFs

1.3.9.1 Overview

This topic provides examples on how to compile and use MaxCompute UDFs.

MaxCompute provides three types of UDFs: UDFs, UDAFs, and UDTFs. These functions are collectively known as UDFs.

**Note:**

- UDFs only support Java APIs. To compile a UDF program, you can upload UDF code to your project by adding resources, and run the CREATE FUNCTION statement to create a UDF.
- This topic provides examples of UDF, UDAF, and UDTF code.

1.3.9.2 UDF example

This topic uses the convert-to-lowercase function as an example to demonstrate the process of creating a UDF. Specifically, follow these steps:

Procedure

1. Write code.

To archive function, write a program and compile in terms of MaxCompute UDF frame.

```
package org.alidata.odps.udf.examples; import com.aliyun.odps.udf.  
UDF;  
public final class Lower extends UDF { public String evaluate(String  
s) {  
if (s == null) { return null; } return s.toLowerCase();  
}  
}
```

Name the preceding JAR package *my_lower.jar*.

2. Add resources.

Specify the referenced UDF code before running UDF. User code must be added to MaxCompute by adding resources. Java UDFs must be compiled into the JAR package and added in MaxCompute as a JAR resource. The UDF framework loads the JAR package automatically and runs UDF.

Example of the command to add JAR resources:

```
add jar my_lower.jar;
```

**Note:**

If multiple resources have the same name, rename the JAR package and modify the name of relevant JAR packages in the example command below. You can also use the "f" option to override the existing JAR resources.

3. Register the UDF.

When your JAR package is uploaded, MaxCompute does not have any information about this UDF. Therefore, you must register a unique function name in MaxCompute, and specify to which function and under which JAR resources this function name corresponding.

An example of using commands to register the UDF is as follows:

```
CREATE FUNCTION test_lower AS org.alidata.odps.udf.examples.Lower  
USING my_lower.jar;
```

Example of the function used in SQL:

```
select test_lower('A') from my_test_table;
```

1.3.9.3 UDAF example

This topic provides an example of UDAF code for your reference.

UDAFs are registered in the same way as UDFs and are used in the same way as built-in aggregate functions.

The following UDAF code is for reference only:

```
package org.alidata.odps.udf.examples;  
import com.aliyun.odps.io.LongWritable; import com.aliyun.odps.io.Text  
;  
import com.aliyun.odps.io.Writable; import com.aliyun.odps.udf.  
Aggregator;  
import com.aliyun.odps.udf.UDFException;  
/**  
project: example_project  
table: wc_in2  
partitions: p2=1,p1=2  
columns: colc,colb,cola  
*/  
public class UDAFExample extends Aggregator {  
    @Override  
    public void iterate(Writable arg0, Writable[] arg1) throws UDFExcepti  
on { LongWritable result = (LongWritable) arg0;  
    for (Writable item : arg1) { Text txt = (Text) item;  
    result.set(result.get() + txt.getLength());  
    }  
}  
    @Override  
    public void merge(Writable arg0, Writable arg1) throws UDFException {  
    LongWritable result = (LongWritable) arg0;  
    LongWritable partial = (LongWritable) arg1; result.set(result.get() +  
    partial.get());  
    }  
    @Override  
    public Writable newBuffer() { return new LongWritable(0L);  
    }  
    @Override
```

```
public Writable terminate(Writable arg0) throws UDFException { return  
arg0;  
}  
}
```

1.3.9.4 UDTF example

This topic provides an example of UDTF code for your reference.

UDTFs are registered and used in the similar way to UDFs.

UDTF code example:

```
package org.alidata.odps.udtf.examples;  
import com.aliyun.odps.udf.UDTF;  
import com.aliyun.odps.udf.UDTFCollector; import com.aliyun.odps.udf.  
annotation.Resolve; import com.aliyun.odps.udf.UDFException;  
// TODO define input and output types, e.g., "string,string->string,  
bigint".  
@Resolve({"string,bigint->string,bigint"}) public class MyUDTF extends  
UDTF {  
@Override  
public void process(Object[] args) throws UDFException { String a = (  
String) args[0];  
Long b = (Long) args[1];  
for (String t: a.split("\\s+")) { forward(t, b);  
}  
}  
}
```

1.3.10 Compile and run a MapReduce job

This topic describes how to quickly run a MapReduce job after the MaxCompute client is installed.

Context

The following procedure uses a WordCount program as an example.

Before you compile and run a MapReduce program, make sure that you have completed the following operations:

- JDK 1.8 has been installed on your host.
- The MaxCompute client has been configured. For more information, see [Configure the client](#).

Procedure

1. Create input and output tables.

Example:

```
create table wc_in (key string, value string);
```

```
create table wc_out (key string, cnt bigint);
```

**Note:**

For more information about the table creation statement, see [Create a table](#).

2. Use the data transfer tool to upload data.**Example:**

```
odpscmd -e "tunnel upload kv.txt wc_in"
```

3. Compile a MaxCompute program and debug it.

- MaxCompute provides an Eclipse plugin to help you quickly develop MapReduce programs and debug them on the local machine.
- You need to create a MaxCompute project in Eclipse, and then compile a MapReduce program in this project. After successful local debugging, upload the compiled program to MaxCompute for base testing.

**Note:**

If the Java program requires the use of resources, you must use the `-resources` parameter to specify the resources.

1.3.11 Compile and run a Graph job

You can submit a Graph job in the same way that you would submit a MapReduce job. This topic provides an example of how to submit a Graph job.

Context

This example uses the SSSP algorithm. The operation procedure is as follows:

Procedure**1. Create input and output tables.****Example:**

```
create table sssp_in (v bigint, es string);  
create table sssp_out (v bigint, l bigint);
```

**Note:**

For more information about the table creation statement, see [Create a table](#).

2. Use the data transfer tool to upload data.

Example:

```
tunnel u -fd " " sssp.txt sssp_in;
```

3. Compile an SSSP.



Note:

During the Graph development process, you can locally compile and debug SSSP algorithm examples. You only need to package the SSSP code. You do not need to package the SDK into `odps-graph-example-sssp.jar`.

4. Add JAR resources.

Example:

```
add jar $LOCAL_JAR_PATH/odps-graph-example-sssp.jar odps-graph-example-sssp.jar
```



Note:

For more information about how to add resources, see [Add resources](#).

5. Run the SSSP.

Example:

```
jar -libjars odps-graph-example-sssp.jar -classpath $LOCAL_JAR_PATH/odps-graph-example-sssp.jar com.aliyun.odps.graph.examples.SSP 1 sssp_in sssp_out;
```



Note:

The MaxCompute client provides a jar command to run MaxCompute Graph jobs. This command is used in the same way as you would use the jar command in MapReduce. The Graph job execution command outputs the job instance ID, execution progress, and result summary. The command output is as follows:

ID = 20170730160742915gl205u3 2017-07-31 00:18:36 SUCCESS

Summary:

```
Graph Input/Output Total input bytes=211 Total input records=5
Total output bytes=161
Total output records=5 graph_input_[bsp.sssp_in]_bytes=211
graph_input_[bsp.sssp_in]_records=5 graph_output_[bsp.sssp_out]
_bytes=161 graph_output_[bsp.sssp_out]_records=5 Graph Statistics
Total edges=14
```

```
Total halted vertices=5 Total sent messages=28 Total supersteps=4
Total vertices=5
Total workers=1 Graph Timers
Average superstep time (milliseconds)=7 Load time (milliseconds)=8
Max superstep time (milliseconds) =14 Max time superstep=0
Min superstep time (milliseconds)=5 Min time superstep=2
Setup time (milliseconds)=277 Shutdown time (milliseconds)=20
Total superstep time (milliseconds)=30 Total time (milliseconds)=
344
OK
```

1.4 Basic concepts and common commands

1.4.1 Terms

This topic describes basic concepts of MaxCompute.

project

A basic organizational unit of MaxCompute. Like a database or schema in a traditional database system, a project is the basic unit of multi-user isolation and access control. A user can have permissions on multiple projects. After being authorized, a user can access objects that belong to a project, such as XXX, from another project.

You can run the Use Project command to enter a project. Example:

```
use my_project
-- Enter a project named my_project.
```



Note:

After you run the preceding command, you will enter a project named my_project and gain permissions to operate objects in the project. You will then be able to operate objects that belong to the project, regardless of which project you are currently in. The Use Project command is provided by the MaxCompute client. For more information, see [Common commands](#).

table

A data storage unit of MaxCompute. Logically, a table is a two-dimensional structure consisting of rows and columns, with each row representing a record and each column representing a field of the same data type. One record can be contained in one or more columns. The column names and types constitute the schema of this table. The data stored in table columns can be of any type that

is supported by MaxCompute. Tables are the input and output objects of all MaxCompute computing tasks. You can create, delete, and import data to tables.

Partitions of a table can be defined to process data more efficiently. You can specify some fields in the table as partition columns. Partitions within a table are comparable to directories within a file system. Each value of a partition column is called a partition in MaxCompute. You can group multiple fields of a table to a single partition column to create a multi-level partition. Multi-level partitions are similar to multi-level directories. If you specify the name of the partition that you want to access, MaxCompute only scans the specified partition instead of the entire table. This improves processing efficiency and reduces cost. For more information, see [Column and partition operations](#).

data type

Columns of a MaxCompute table must be of a certain type. MaxCompute supports the following data types:

Table 1-1: Basic data types

Type	New in MaxCompute 2.0?	Constant	Description	Value range
TINYINT	Yes	1Y,-127Y	The 8-bit signed integer type.	-128 to 127
SMALLINT	Yes	32767S,-100S	The 16-bit signed integer type.	-32,768 to 32,767
INT	Yes	1000,-15645787	The 32-bit signed integer type.	-2 ³¹ to 2 ³¹ -1
BIGINT	No	1000000000000L, -1L	The 64-bit signed integer type.	-2 ⁶³ + 1 to 2 ⁶³ -1

Type	New in MaxCompute 2.0?	Constant	Description	Value range
STRING	No	abc, bcd, alibaba, inc	The UTF-8 coded string. The character behaviors of other codes are not defined.	The size of all values in a string column cannot exceed 8 MB.
FLOAT	Yes	None.	The 32-bit binary floating point type.	/
BOOLEAN	No	True or False	The Boolean type.	True or False 308
DOUBLE	No	3.1415926 1E+7	The 64-bit binary floating point type.	-1.7976931348623157E+308 to 1.7976931348623157E+308
DATETIME	No	Datetime '2017-11-11 00:00:00'	The date and time type. The standard system time is UTC+8.	0001-01-01 00:00:00 000 to 9999-12-31 23:59:59 999 36
DECIMAL	No	3.5BD, 999999999999.99999999BD	The precise numeric type based on the decimal system.	Integer: 36-10 + 1 to 10 - 1 Fractional: -18 round to 10
VARCHAR	Yes	None.	The variable-length type. n specifies the length.	1 to 65,535
BINARY	Yes	None.	The binary data type.	A single binary column cannot exceed 8 MB.

Type	New in MaxCompute 2.0?	Constant	Description	Value range
TIMESTAMP	Yes	Timestamp '2017-11-11 00:00:00.123456789'	The timestamp type. This type is not time zone specific.	0001-01-01 00:00:00 0000000000 to 9999-12-31 23:59:59 9999999999

Note that if you want to use the new data types in MaxCompute 2.0, you must first execute the following statement to enable the new data type flag: `set odps.sql.type.system.odps2=true;` (at the session level) or `setproject odps.sql.type.system.odps2=true;` (at the project level). Otherwise, the following error may occur: `xxxx type is not enabled in current mode.` The data types listed in the preceding table can be NULL.



Note:

Only supported lowercase letters can be used in the preceding statements.



Notice:

Note the following points when you use the new data types in MaxCompute 2.0:

- After you execute the `set odps.sql.type.system.odps2=true;` statement, it brings the following major impacts:
 - The semantics of the INT keyword changes. INT in the SQL statement indicates a 32-bit integer.
 - The semantics of an integer constant changes. For example, in the `SELECT 1 + a;` statement,
 - If the new data type flag is not enabled, the integer constant is processed as BIGINT. If the length of the constant exceeds the range of values for a BIGINT value, the integer constant is processed as DOUBLE.
 - If the new data type flag is enabled, the integer constant is 1 of the 32-bit INT type.
 - Possible compatibility issues: The INT type may lead to inconsistencies in function prototypes during subsequent operations. For example, the

actions of peripheral tools and subsequent operations might be changed by new type tables generated after data is written to a disk.

- **Implicit type conversion rules change.**

If the new data type flag is enabled, some implicit types may not be converted. For example, errors may occur or precision may be reduced when converting the data type from STRING to BIGINT, from STRING to DATETIME, from DOUBLE to BIGINT, from DECIMAL to DOUBLE, or from DECIMAL to BIGINT. In this case, you can use the CAST function to convert the data type.

Functions and the INSERT statement are affected greatly by implicit type conversion. For example, the INSERT statement can be executed when the new data type flag is disabled, but will return an error when the flag is enabled.

- **Some operations and built-in functions that use new data types as parameters and response values are ignored when the new data type flag is disabled. When the new data type flag is enabled, they become valid.**

■ **Some built-in functions can only be used after the new data type flag is enabled. This includes most functions that use INT type parameters and subsequently suffer from BIGINT overload, such as YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE, SECOND, MILLISECOND, NANOSECOND**

, DAYOFMONTH, and WEEKOFYEAR. These functions can actually be implemented by using the DATEPART function.

■ UDF resolution changes.

- The resolution of the BIGINT keyword changes.
- The partition column types change.

■ If the new data type flag is disabled, the partition column type can only be STRING.

■ If the new data type flag is enabled, the partition column type can be STRING, VARCHAR, CHAR, TINYINT, SMALLINT, INT, or BIGINT.

■ If the new data type flag is disabled, partition fields in INSERT operations are processed as STRING.

- The behavior of the LIMIT statement changes.

For example, in the `SELECT * FROM t1 UNION ALL SELECT * FROM t2 limit 10;` statement,

■ If the new data type flag is disabled, it is `SELECT * FROM t1 UNION ALL SELECT * FROM (SELECT * FROM t2 limit 10) t2;`

■ If the new data type flag is enabled, it is `SELECT * FROM (SELECT * FROM t1 UNION ALL SELECT * FROM t2) t limit 10;`

Actions of the UNION, INTERSECT, EXCEPT, LIMIT, ORDER BY, DISTRIBUTE BY, SORT BY, and CLUSTER BY statements also change if the new data type flag is enabled.

- The resolution of the IN expression changes.

For example, in the `a in (1, 2, 3)` expression,

■ If the new data type flag is disabled, all the values in the parentheses () must be of the same type.

■ If the new data type flag is enabled, all the values in the parentheses () are implicitly converted to the same type.

- If the value of a constant is bigger than the maximum value of INT but smaller than the maximum value of BIGINT, it is converted to BIGINT. If the constant is greater than the maximum value of BIGINT, it is converted to DOUBLE. If `odps.sql.type.system.odps2` is not set to true, MaxCompute retains the conversion and will notify you that the INT data is being processed as the BIGINT type. If

`odps.sql.type.system.odps2` is not set to true, we recommend that you change these types to BIGINT to prevent confusion.

- VARCHAR constants can be expressed through implicitly converted STRING constants.
- STRING constants can be combined. For example, abc and xyz can be combined as abcxyz. Different parts can be written in different rows.
- Time values in milliseconds cannot be displayed. You can add `-dfp` in the Tunnel command to specify the time format to display milliseconds.

MaxCompute supports complex data types. The following table lists their definitions and constructors.

Table 1-2: Complicated data types

Type	Definition	Constructor
Array	<code>array< int >;</code> <code>array< struct< a:int, b:string >></code>	<code>array(1, 2, 3);</code> <code>array(array(1, 2);</code> <code>array(3, 4))</code>
Map	<code>map< string, string >;</code> <code>map< smallint, array< string >></code>	<code>map("k1", "v1", "k2", "v2");</code> <code>map(1S, array('a', 'b'), 2S,</code> <code>array('x', 'y'))</code>
Struct	<code>struct< x:int, y:int>;</code> <code>struct< field1:bigint,</code> <code>field2:array< int>, field3:</code> <code>map< int, int>></code>	<code>named_struct('x', 1, 'y', 2);</code> <code>named_struct('field1',</code> <code>100L, 'field2', array(1, 2), '</code> <code>field3', map(1, 100, 2, 200))</code>

resource

A concept used in MaxCompute. To accomplish tasks using user-defined functions (UDFs) or MapReduce, you must use resources.

- MaxCompute SQL UDF: After you write a UDF, you must compile it as a JAR package and upload the package to MaxCompute as a resource. When the UDF is run, MaxCompute automatically downloads this JAR package and obtains the

code to run the UDF. JAR packages are a type of MaxCompute resources. When you upload a JAR package, a resource is created in MaxCompute.

- **MaxCompute MapReduce:** After you write a MapReduce program, you must compile it as a JAR package and upload the package to MaxCompute as a resource. When you run a MapReduce job, the MapReduce framework automatically downloads this JAR package and obtains the code to run the MapReduce job.



Note:

- There are some limits on how MaxCompute UDFs and MapReduce access resources. For more information, see [Limits](#).
- You can also upload tables or text files to MaxCompute as different types of resources. You can read or use these resources when you run UDFs or MapReduce jobs. MaxCompute provides APIs for you to read and use resources. For more information, see the examples in resource use and [UDTF instructions](#).

MaxCompute supports the following resource types:

- **File**
- **Table:** tables in MaxCompute.
- **JAR:** compiled Java JAR packages.
- **Archive:** compressed files identified by the resource name extension. Supported file types include .zip, .tgz, .tar.gz, .tar, and .jar.
- **Py:** Python scripts used by Python UDFs.



Note:

For more information about resource operations, see [Resource operations](#).

UDF

MaxCompute is equipped with the SQL computing capabilities. You can use the built-in functions in MaxCompute SQL statements to implement certain computing or counting functions. If the built-in functions do not meet your requirements, you can use Java programming APIs provided by MaxCompute to develop UDFs. UDFs are classified into user-defined scalar functions (UDSFs), user-defined aggregate functions (UDAFs), and user-defined table-valued functions (UDTFs).

After you develop the UDF code, you need to compile the code to a JAR package, upload it to MaxCompute as a resource, and register this UDF in MaxCompute. To use a UDF in MaxCompute, you can simply specify its name and parameters in an SQL statement as you would when using the built-in MaxCompute functions.

**Note:**

For more information about UDF operations, see [Function operations](#).

task

A basic computing unit of MaxCompute. Both SQL and MapReduce features are implemented as tasks. MaxCompute parses most of the tasks that you submit, such as SQL DML statements and MapReduce tasks. MaxCompute generates a task execution plan based on the parsing results.

An execution plan consists of several mutually dependent stages. An execution plan can be logically defined as a directed graph. Vertices of the graph represent stages and edges of the graph represent dependencies between stages. MaxCompute executes stages based on the dependencies in the graph (execution plan). A stage has multiple threads, also known as workers. The workers in each stage cooperate to complete computations for the stage. Different workers within a stage each process different data but run on the same execution logic.

A computing task is converted to an instance during execution. You can perform operations such as obtaining status information and killing the instance.

Some MaxCompute tasks, such as SQL DDL statements, are not computing tasks. These tasks only need to read and modify metadata in MaxCompute. MaxCompute cannot generate execution plans for these tasks.

**Notice:**

MaxCompute does not convert all requests to tasks. For example, project, resource, UDF, and instance operations are not executed as tasks.

task instance

Some MaxCompute tasks are converted to instances during the execution process. An instance experiences two stages: Running and Terminated. Instances that are in the Running stage are also in the Running state, while instances that are in the Terminated stage can be in any of the Success, Failed, or Canceled states. You can

query or modify the status of a running task based on the instance ID provided by MaxCompute. For example:

```
status <instance_id>;  
-- Query the status of an instance.  
kill <instance_id>;  
-- Terminate an instance and change its status to Canceled.
```

resource quota

There are two types of resource quota: storage and computation. The storage quota is the upper limit of storage space that you configure for a project. When the storage usage approaches the storage quota, an alarm is triggered. There are two restrictions on computing resources: memory and CPU. The memory and CPU resources occupied by processes running simultaneously in the project cannot exceed the specified upper limit.

1.4.2 Common commands

1.4.2.1 Introduction

MaxCompute allows you to perform operations on objects, such as projects, tables, resources, and instances. You can use client commands or the SDK to perform operations on these objects.

This topic describes how to run these commands in the MaxCompute client.



Note:

- **For more information about how to install and configure the client, see [Configure the client](#).**
- **For more information about the SDK, see [SDK introduction](#).**

1.4.2.2 Project operations

This topic describes common project commands.

Create or delete a project

MaxCompute does not provide commands for creating or deleting projects. You can configure or operate your projects on the console.

Access a project

Command syntax:

```
use <project_name>;
```

Purpose: It is used to access the specified project. After you enter a project, you can directly operate all objects in the project.



Note:

If the specified project does not exist or you have not been added to the project, the system returns an exception and exits.

Example:

```
odps@ myproject> use my_project;  
-- my_project is a project that you have permission to access.
```



Note:

- The preceding command runs in the client.
- All command keywords, project names, table names, and column names in MaxCompute are case-insensitive.
- After you run the command, you can directly access objects in this project.

Example:

Run the following command to access the test_src table in the my_project project (assume test_src exists in my_project):

```
odps@ myproject>select * from test_src;
```

MaxCompute automatically searches for the table from my_project. If this table exists, its data is returned. Otherwise, the system returns an exception and exits.

If you are in my_project and want to access the test_src table in my_project2, you must specify the project name. Run the following command to access test_src in my_project2:

```
odps@ myproject>select * from my_project2.test_src;
```

Data of test_src in my_project2, not my_project, is returned.

View projects

Command syntax:

```
list projects;
```

Purpose: It is used to display all projects in MaxCompute.

Clear objects from a project

Run the following command to view objects in the recycle bin:

```
show recyclebin;
```

Purpose: It is used to list all objects in the project recycle bin.



Note:

Only the project owner can run this command.

Run the following command to clear all objects from the project recycle bin:

```
purge all;
```

Purpose: It is used to clear all objects from the project recycle bin to release storage space.



Note:

Only the project owner can run this command.

Run the following command to clear a table:

```
purge table tblname;
```

Purpose: It is used to clear all objects in a specified table from the recycle bin to release the storage space.



Note:

- If the specified table exists, the project owner and users who have write permissions on the table can run this command.
- If the table has been deleted using a DROP command, only the project owner can run this command.

1.4.2.3 Table operations

This topic describes common table operation commands.

Create Table

Command syntax:

```
CREATE TABLE [IF NOT EXISTS] table_name  
[(col_name data_type [COMMENT col_comment], ...)] [COMMENT table_comm  
ent]  
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)] [  
LIFECYCLE days]  
[AS select_statement];  
CREATE TABLE [IF NOT EXISTS] table_name  
LIKE existing_table_name;
```

Purpose: It is used to create a table.

Example:

```
CREATE TABLE IF NOT EXISTS sale_detail( shop_name STRING,  
customer_id STRING, total_price DOUBLE)  
PARTITIONED BY (sale_date STRING,region STRING);  
-- If the table named sale_detail does not exist, a partitioned table  
with this name is created.
```



Note:

- Table names and column names are case-insensitive.
- A table name or column name cannot contain special characters. It can contain only lowercase English letters (a to z), uppercase English letters (A to Z), numbers, or underscores (_). A name must start with an English letter and cannot exceed 128 bytes. If a name does not match any of the preceding rules, an error is returned.
- A comment must be a valid string within 1,024 bytes. Otherwise, an error is returned.
- For more information about this command, see [Create a table](#).

Drop Table

Command syntax:

```
DROP TABLE [IF EXISTS] table_name;
```

Purpose: It is used to delete a table.



Note:

If the command is run without the IF EXISTS option and the table does not exist, an exception is returned. If this option is specified, a success is returned regardless of whether the table exists.

Description:

table_name: the name of the table to be deleted.

Example:

```
DROP TABLE sale_detail;  
-- If the table exists, a success is returned.  
DROP TABLE IF EXISTS sale_detail;  
-- A success is returned regardless of whether the sale_detail table  
exists.
```

Describe Table**Command syntax:**

```
DESC <table_name>;
```

Purpose: It is used to return the information of the specified table. The following information is returned: Owner, Project, CreateTime, LastDDLTime, LastModifiedTime, InternalTable (indicates that the object is a table. This value is always YES), Size (table size in bytes), Native Columns (non-partition column information, including column name, type, and remarks), and Partition Columns (partition column information, including column name, type, and remarks).

Description:

table_name: the name of a table or view.

Example:

```
odps@ project_name>DESC sale_detail;  
-- Describe a partitioned table.  
+-----+  
+  
| Owner: ALIYUN$odpsuser@aliyun.com | Project: test_project |  
| TableComment: |  
+-----+  
+  
| CreateTime: 2017-01-01 17:32:13 |  
| LastDDLTime: 2017-01-01 17:57:38 |  
| LastModifiedTime: 2017-01-01 18:00:00 |  
+-----+  
+  
| InternalTable: YES | Size: 0 |  
+-----+  
+  
| Native Columns: |
```

```
+-----+
+
|Field | Type | Comment |
+-----+
+
|shop_name | string | |
|customer_id | string | |
|total_price | double | |
+-----+
+
|Partition Columns: |
+-----+
+
|sale_date | string | |
|region | string | |
+-----+
+
```

**Note:**

- The preceding command is run in the client.
- For a non-partitioned table, the Partition Columns option is not displayed.
- For a view, Internal Table is replaced by Virtual View, of which the value is always YES. Size is replaced by View Text, which is the view definition. For example: `select * > from src`. For more information about views, see [Create a view](#).

Show Tables**Command syntax:**

```
SHOW TABLES;
```

Purpose: It is used to list all tables in the current project.

Example:

```
odps@ project_name>show tables;
ALIYUN$odps_user@aliyun.com:table_name
.....
```

**Note:**

- The preceding command is run in the client.
- `odps_user@aliyun.com` is the user who creates the table.
- `table_name` is the name of the table.

Show Partitions

Command syntax:

```
SHOW PARTITIONS <table_name>;
```

Purpose: It is used to list all partitions of the specified table.

Description:

table_name: the name of the table to query. An error is returned if the specified table does not exist or is a non-partitioned table.

Example:

```
SHOW PARTITIONS table_name;  
partition_col1=col1_value1/partition_col2=col2_value1  
partition_col1=col1_value2/partition_col2=col2_value2
```

**Note:**

- The preceding command is run in the client.
- partition_col1 and partition_col2 indicate the partition columns of the table.
- col1_value1, col2_value1, col1_value2, and col2_value2 indicate the values of the corresponding partition columns.

1.4.2.4 Instance operations

This topic describes common commands for instance operations.

Show Instances

Command syntax:

```
SHOW INSTANCES [FROM startdate TO enddate] [number]
```

Purpose: It is used to return information about instances that you have created.

Description:

- **startdate To enddate:** specifies a time period. Information about instances created in the specified period is returned. The dates must be in the format of yyyy-mm-dd. This parameter is optional. If it is not specified, information about instances that you created in the last three days is returned.
- **number:** specifies the number of instances to be returned. Information about the specified number of latest instances is returned in chronological order. If it

is not specified, information about all instances that meet the requirements is returned.

Output items: include StartTime (in seconds), RunTime (s), Status (including Waiting, Success, Failed, Running, Cancelled, and Suspended), InstanceID, and corresponding SQL statement. The following figure shows the command output.

Figure 1-2: Command output

StartTime	RunTime	Status	InstanceID	Query
2015-04-28 13:57:55	1s	Success	20150428055754916grvd5vj4	select * from tab_pack_priv limit 20;
...
...

An instance can be in any of the following states:

- Running
- Success
- Waiting
- Failed: The job failed, but data in the target table is not modified.
- Suspended
- Cancelled

Status Instance

Command syntax:

```
STATUS <instance_id>;
```

Purpose: It is used to return the status of a specified instance, which can be Success, Failed, Running, or Canceled.



Note:

If the specified instance is not created, an exception is returned.

Description:

instance_id: the unique identifier of an instance. It specifies the instance of which the status is queried.

Example:

```
status 20171225123302267gk3u6k4y2; Success
```

```
-- View the status of the instance with ID of 20171225123302267gk3u6k4y2. The instance status is Success.
```



Note:

The preceding command runs in the client.

Kill Instance

Command syntax:

```
kill <instance_id>;
```

Purpose: It is used to terminate the specified instance, which must be running. Note that this is an abnormal process. A return from the command only means that the system has received the request. It does not necessarily mean that the job has been stopped. Therefore, you need to run the status command to view the instance status .

Description:

instance_id: the unique identifier of an instance. It must be the ID of a running instance. Otherwise, an error is returned.

Example:

```
kill 20171225123302267gk3u6k4y2;  
-- Terminate the instance with ID of 20171225123302267gk3u6k4y2.
```



Note:

The preceding command runs in the client.

Desc Instance

Command syntax:

```
desc instance <instance_id>;
```

Purpose: It is used to obtain the job information based on a specific instance ID. The obtained information includes the specific SQL database, owner, start time, end time, and status.

Description:

instance_id: the unique identifier of an instance.

Example:

```
desc instance 20170715103441522gond1qa2;  
ID 20170715103441522gond1qa2  
Owner ALIYUN$maojing.mj@alibaba-inc.com  
StartTime 2017-07-15 18:34:41  
EndTime 2017-07-15 18:34:42  
Status Terminated  
console_select_query_task_1436956481295 Success Query select * from  
mj_test;  
-- Query job information corresponding to the instance with the ID of  
20170715103441522gond1qa2.
```

**Note:**

The preceding command runs in the client.

1.4.2.5 Resource operations

This topic describes common resource operation commands.

Add a resource

Command syntax:

```
add file <local_file> [as alias] [comment 'cmt'][-f];  
add archive <local_file> [as alias] [comment 'cmt'][-f];  
add table <table_name> [partition <(spec)>] [as alias] [comment 'cmt']  
[-f];  
add jar <local_file.jar> [comment 'cmt'][-f];  
add py <local_file.py> [comment 'cmt'][-f];
```

The following table describes the parameters.

Table 1-3: Parameters

Parameter	Description
file/archive/table/jar/py	Indicates the resource type. For more information about resource types, see Resource in the "Basic concepts" topic.
local_file	Indicates the path of the local file. The file name is used as the resource name. A resource name is a unique identifier of a resource.
table_name	Indicates the name of a table in MaxCompute.
[PARTITION (spec)]	If the resource to be added is a partitioned table, MaxCompute only takes a partition, not the whole partitioned table, as a resource.

Parameter	Description
alias	Specifies a resource name. If this parameter is not specified, the file name is used as the resource name by default. JAR and PY resources do not support this function.
[comment 'cmt']	Adds a comment to the resource.
[-f]	If a resource with the same name exists, this operation overwrites the existing resource. If this parameter is not specified and a resource with the same name exists, the operation fails.

Example:

```
odps@ odps_public_dev>add table sale_detail partition (ds='20170602')
as sale.res comment 'sale detail on 201706 02' -f;
OK: Resource 'sale.res' have been updated.
-- Add a table resource with alias sale.res to MaxCompute.
```

**Note:**

The size of each resource file cannot exceed 64 MB. The total size of resources referenced by a single SQL or MapReduce task cannot exceed 512 MB.

Delete a resource

Command syntax:

```
DROP RESOURCE <resource_name>;
```

Description:

resource_name: the resource name specified when creating the resource.

View the resource list

Command syntax:

```
LIST RESOURCES;
```

Purpose: It is used to list all resources in the current project.

Example:


```
odps@ $project_name>list resources;
```

Resource Name	Comment	Last Modified Time	Type
1234.txt	2014-02-27 07:07:56	file	
mapred.jar	2014-02-27 07:07:57	jar	

1.4.2.6 Function operations

This topic describes common function operation commands.

Create a function

Command syntax:

```
CREATE FUNCTION <function_name> AS <package_to_class> USING<resource_list>;
```

The following table lists the parameters.

Table 1-4: Parameters

Parameter	Description
function_name	Indicates the UDF name. This is the name used in SQL statements to reference this function.
package_to_class	For a Java UDF, this name is a fully qualified class name, including names from the top-level package name to the UDF implementation class name. For a Python UDF, this name is the Python script name.classname. This name must be enclosed in quotation marks.
resource_list	Indicates the resource list that is used by the UDF. It must include the resource where the UDF code is located. If the user code reads resource files through the distributedcache API, this list must also include the list of resource files that the UDF reads. A resource list consists of multiple resource names separated by commas (.). The resource list must be enclosed in quotation marks.

Example:

Java UDF class org.alidata.odps.udf.examples.Lower is in my_lower.jar. Run the following command to create function my_lower:

```
CREATE FUNCTION test_lower AS 'org.alidata.odps.udf.examples.Lower'
```

```
USING 'my_lower.jar';
```

Python UDF MyLower is in pyudf_test.py. Run the following command to create function my_lower:

```
create function my_lower as 'pyudf_test.MyLower';using 'pyudf_test.py';
```



Note:

- **Similar to resource file names, function names must be unique.**
- **UDFs typically cannot overwrite system built-in functions. Only a project owner can overwrite built-in functions. If you use a UDF to overwrite a built-in function, the summary will include warning information after the SQL statement is executed.**

Delete a function

Command syntax:

```
DROP FUNCTION <function_name>;
```

Example:

```
DROP FUNCTION test_lower;
```

1.4.2.7 Tunnel operations

This topic describes some common tunnel operation commands.

Tunnel operations

- **Upload:** is used to upload files or directories (first-level directories). Data can only be uploaded to a table or a partition of a table. For a table with partitions, specify the target partition for the upload.

```
tunnel upload log.txt test_project.test_table/p1="b1",p2="b2";
```

```
tunnel upload log.txt test_table --scan=only;
```

- **Download:** is used to download a table or partition to a single file only. For a partitioned table, specify the target partition to be downloaded.

```
tunnel download test_project.test_table/p1="b1",p2="b2" log.txt;
```

- **Resume:** Dship supports resumable data transfer for files or directories after service interruption due to network or Tunnel service errors.

```
tunnel resume;
```

- **Show:** is used to display historical task information.

```
tunnel show history -n 5  
tunnel show log
```

- **Purge:** is used to clear the session directory. Sessions from the past three days are purged by default.

```
tunnel purge 5
```

Use of tunnel commands

Use the help subcommand to obtain the help information. Every command and selection supports short the command format as follows:

```
odps@ project_name>tunnel help;  
Usage: tunnel <subcommand> [options] [args]  
Type 'tunnel help <subcommand>' for help on a specific subcommand.
```

Available subcommands:

```
upload (u)  
download (d)  
resume (r)  
show (s)  
purge (p)  
help (h)
```

tunnel is a command for uploading data to / downloading data from MaxCompute.

The parameters are described as follows.

Table 1-5: Parameters

Parameter	Description
upload	Uploads data to MaxCompute tables.
download	Downloads data from MaxCompute tables.

Parameter	Description
resume	Resumes data upload in case of a failure. Currently, resume is only supported for data upload. Each data download or upload operation is called a session. Run the Resume command and specify the session ID to be resumed.
show	Displays historical running information.
purge	Clears the session directory.
help	Outputs tunnel help information.

Upload

Import data of local files to MaxCompute tables in the append mode. Subcommand format:

```
usage: tunnel upload [options] <path> <[project.]table[/partition]>
        upload data from local file
  -bs,-block-size <ARG>          block size in MiB, default 100
  -c,-charset <ARG>              specify file charset, default
ignore.
                                set ignore to download raw data
  -cp,-compress <ARG>            compress, default true
  -dbr,-discard-bad-records <ARG> specify discard bad records
                                action(true|false), default false
  -dfp,-date-format-pattern <ARG> specify date format pattern,
default
                                yyyy-MM-dd HH:mm:ss
  -fd,-field-delimiter <ARG>     specify field delimiter, support
                                unicode, eg \u0001. default ",",
                                if local file should have table
  -h,-header <ARG>               header,
                                default false
  -mbr,-max-bad-records <ARG>    max bad records, default 1000
  -ni,-null-indicator <ARG>     specify null indicator string,
default
                                ""(empty string)
  -rd,-record-delimiter <ARG>    specify record delimiter, support
                                unicode, eg \u0001. default "\n"
  -s,-scan <ARG>                specify scan file
                                action(true|false|only), default
true
  -sd,-session-dir <ARG>         set session dir, default /D:/
console/plugins/dship/
  -te,-tunnel_endpoint <ARG>     tunnel endpoint
  -threads <ARG>                 number of threads, default 1
  -tz,-time-zone <ARG>           time zone, default local timezone:
                                Asia/Shanghai
Example:
  tunnel upload log.txt test_project.test_table/p1="b1",p2="b2"
```

The parameters are described as follows.

Table 1-6: Parameters

Parameter	Description
-bs, block-size	Specifies the size of each data block uploaded using Tunnel. Default value: 100 MiB (1 MiB = 1,024 * 1,024 bytes).
-c, -charset	Specifies the encoding of local data files. The default value is UTF-8 without timing. The source data is downloaded by default.
-cp, -compress	Determines whether the local file is compressed to reduce traffic before being uploaded. Compression is enabled by default.
-dbr	Indicates whether to ignore dirty data (additional columns, missing columns, and unmatched types of column data). If the value is true, all data not complying with table definitions is ignored. If the value is false, an error is returned when dirty data is found, so that raw data in the target table is not polluted.
-dfp	Specifies the datetime type. The default format is yyyy-MM-dd HH:mm:ss.
-fd	Specifies the column delimiter used in the local data file. The default delimiter is comma (,).
-h	Indicates whether the data file includes a table header. If the value is true, Dship skips the table header and starts uploading data from the second row.
-mbr, -max-bad-records	If more than 1,000 rows of dirty data are uploaded, the upload operation is terminated by default. This parameter allows you to adjust the tolerated volume of dirty data.
-ni	Specifies the NULL data tag. The default value is "" (NULL string).
-rd	Specifies the row delimiter in the local data file. The default value is \n in a Linux system, or \r\n in a Windows system.

Parameter	Description
-s	Indicates whether to scan the local data file. The default value is false . If the value is true, the data is scanned first and can be imported only if the data format is correct. If the value is false, the data is directly imported without a scan. If the value is only , the local data is only scanned and not imported.
-sd, -session-dir	Specifies the path of the session directory. The default value is /D:/console/plugins/dship/lib /.. (the specific path of plugins/dship/lib).
-te	Specifies the endpoint of the tunnel.
-tz	Specifies the time zone. The default value is the local time zone: Asia/Shanghai.

Example:**Create the target table:**

```
CREATE TABLE IF NOT EXISTS sale_detail(  
    shop_name STRING,  
    customer_id STRING,  
    total_price DOUBLE)  
PARTITIONED BY (sale_date STRING,region STRING);
```

Add a partition:

```
alter table sale_detail add partition (sale_date='201705', region='hangzhou');
```

Prepare the data file data.txt, with the following contents:

```
shop9,97,100  
shop10,10,200  
shop11,11
```

The data of the third row of this file are not consistent with the definition in Table sale_detail. sale_detail defines three columns, but the data have only two columns.

```
odps@ project_name>tunnel u d:\data.txt      sale_detail/sale_date  
=201705,region=hangzhou -s false Upload session:      2017061016  
39224880870a002ec60c  
Start upload:d:\data.txt  
Total bytes:41 Split input to 1 blocks  
2017-06-10 16:39:22 upload block: '1'
```

```
ERROR: column mismatch -,expected 3 columns, 2 columns found, please
check      data or delimiter
```

In this case, data import fails because of the dirty data in data.txt. The system displays the session ID and error message. The data verification process is as follows:

```
odps@ odpstest_ay52c_ay52> select * from sale_detail where sale_date='
201705'; ID = 20170610084135370gyvc61z5
+-----+-----+-----+-----+-----+
shop_name | customer_id | total_price | sale_date | region |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

Download

Subcommand syntax:

```
odps@ project_name>tunnel help download
usage: tunnel download [options] <[project.]table[/partition]> <path>
      download data to local file
  -c,-charset <ARG>          specify file charset, default
ignore.
  -cp,-compress <ARG>        set ignore to download raw data
compress, default true
  -dfp,-date-format-pattern <ARG> specify date format pattern,
default
yyyy-MM-dd HH:mm:ss
  -e,-exponential <ARG>      When download double values, use
exponential express if necessary.
Otherwise at most 20 digits will be
reserved. Default false
  -fd,-field-delimiter <ARG> specify field delimiter, support
unicode, eg \u0001. default ","
if local file should have table
header,
  -h,-header <ARG>           default false
  -limit <ARG>                specify the number of records to
Download
  -ni,-NULL-indicator <ARG>  specify null indicator string,
default
""(empty string)
  -rd,-record-delimiter <ARG> specify record delimiter, support
unicode, eg \u0001. default "\n"
  -sd,-session-dir <ARG>     set session dir, defa /D
:/console/plugins/dship/
  -te,-tunnel_endpoint <ARG> tunnel endpoint
  -threads <ARG>             number of threads, default 1
  -tz,-time-zone <ARG>       time zone, default local timezone:
Asia/Shanghai
Example:
tunnel download test_project.test_table/p1="b1",p2="b2" log.txt
```

The parameters are described as follows.

Table 1-7: Parameters

Parameter	Description
-fd	Specifies the column delimiter used in the local data file. The default delimiter is a comma (,).
-rd	Specifies the row delimiter used in the local data file. The default delimiter is \r\n.
-dfp	Specifies the datetime format. The default format is yyyy-MM-dd HH:mm:ss.
-ni	Specifies the NULL data tag. The default value is "" (NULL string).
-c	Specifies the encoding of local data files. The default value is UTF-8.

Example:**Download data to result.txt.**

```
$ ./tunnel download sale_detail/sale_date=201705,region=hangzhou  
result.txt; Download session: 201706101658245283870a002ed0b9  
Total records: 2  
2017-06-10 16:58:24 download records: 2  
2017-06-10 16:58:24 file size: 30 bytes  
OK
```

The data verification process is as follows, and the contents of result.txt are:

```
shop9,97,100.0  
shop10,10,200.0
```

Resume**Repair and re-executes historical records (only valid for data uploads).****Subcommand syntax:**

```
usage: tunnel resume [session_id] [--force]  
        resume an upload session  
-f,--force force resume  
Example:  
        tunnel resume
```

Example:**The data.txt is changed to:**

```
shop9,97,100
```



```
shop10,10,200
```

Resume data uploading:

```
odps@ project_name>tunnel resume 201706101639224880870a002ec60c --
force;
start resume
201706101639224880870a002ec60c
Upload session: 201706101639224880870a002ec60c
Start upload:d:\data.txt
Resume 1 blocks
2017-06-10 16:46:42 upload block: '1'
2017-06-10 16:46:42 upload block complete, blockid=1
upload complete, average > speed is 0 KB/s
OK
```

201706101639224880870a002ec60c is the ID of the failed data upload session . The data verification process is as follows:

```
odps@ project_name>select * from sale_detail where sale_date='201705';
ID = 20170610084801405g0a741z5
+-----+-----+-----+-----+-----+
shop_name | customer_id | total_price | sale_date | region |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

Show

Display historical records. Subcommand syntax:

```
usage: tunnel show history [options]
```

Example:

```
odps@ project_name>tunnel show history;
usage: tunnel show history [options]
show session information
-n,-number <ARG> lines
Example:
tunnel show history -n 5
tunnel show log
```

Purge

Purge the session directory. By default, sessions from the past 3 days are purged.

Subcommand syntax:

```
usage: tunnel purge [n]
force session history to be purged.([n] days before,
default
3 days)
Example:
```

```
tunnel purge 5
```

1.4.2.8 Other operations

This topic describes common commands for other operations.

ALIAS command

The ALIAS command is used to read different resources (data) from the MaxCompute MapReduce reference manual or UDF code by using a fixed resource name.

Command syntax:

```
ALIAS <alias>=<real>;
```

Purpose: It is used to create an alias for resources.

Example:

```
ADD TABLE src_part PARTITION (ds='20171208') AS res_20171208; ADD
TABLE src_part PARTITION (ds='20171209') AS res_20171209;
ALIAS resName=res_20171208;
jar -resources resName -libjars work.jar -classpath ./work.jar com.
company.MainClass args ...;
-- Job 1
ALIAS resName=res_20171209;
jar -resources resName -libjars work.jar -classpath ./work.jar com.
company.MainClass args ...;
-- Job 2
```



Note:

In the preceding example, resource alias resName references different resource tables in two jobs. You can use the same code to read different data.

Set

Command syntax:

```
set ["<KEY>=<VALUE>"]
```

Purpose: It is used to configure built-in or custom system variables of MaxCompute.

MaxCompute supports the following system variables.

MaxCompute SQL and new versions of MapReduce support the following set commands:

```
set odps.stage.mapper.mem=
-- Set the memory size of every map worker. Unit: MB. Default value: 1
,024 MB.
set odps.stage.reducer.mem=
```

```
-- Set the memory size of every reduce worker. Unit: MB. Default value
: 1,024 MB.
set odps.stage.joiner.mem=
-- Set the memory size of every join worker. Unit: MB. Default value:
1,024 MB.
set odps.stage.mem =
-- Set the memory size of all workers of a specified MaxCompute task
. This command has a lower priority than the three preceding set
commands. Unit: MB. Default value: undefined.
set odps.stage.mapper.split.size=
-- Set the input data volume of each map worker (size of each slice in
the input file). You can use this command to indirectly control the
number of workers in each map stage. Unit: MB. Default value: 256 MB.
set odps.stage.reducer.num=
-- Set the number of workers in each reduce stage. Default value:
undefined.
set odps.stage.joiner.num=
-- Set the number of workers in each join stage. Default value:
undefined.
set odps.stage.num=
-- Set the worker concurrency in all stages of a specified MaxCompute
task. This command has a lower priority than the preceding three set
commands. Default value: undefined.
```

The old MapReduce versions of MaxCompute support the following set commands:

```
set odps.mapred.map.memory=
-- Set the memory size of each map worker. Unit: MB. Default value: 1,
024 MB.
set odps.mapred.reduce.memory=
-- Set the memory size of each reduce worker. Unit: MB. Default value
: 1,024 MB.
set odps.mapred.map.split.size=
-- Set the input data volume of each map worker (size of each slice in
the input file) to indirectly control the number of workers in each
map stage. Unit: MB. Default value: 256 MB.
set odps.mapred.reduce.tasks=
-- Set the number of workers in each reduce stage. Default value:
undefined.
```

SetProject

Command syntax:

```
setproject ["<KEY>=<VALUE>"];
```

Purpose: It is used to set project attributes. If **< KEY >=< VALUE >** is not specified, the current configurations of the project attribute are displayed.

The following table describes project attributes.

Table 1-8: Project attributes

Attribute	Configured by	Description	Value range
odps.table.drop.ignorenonexistent	All users	Indicates whether to report an error when you try to delete a table that does not exist . If the value is true, no error is reported.	true, false
odps.instance.priority.autoadjust	Project owner	Indicates whether to automatically adjust the priorities of tasks to give higher priority to smaller tasks. If the value is true, this function is enabled.	true, false
odps.instance.priority.level	Project owner	Sets the priorities of tasks in a project. The value 1 indicates the highest priority.	1-3
odps.security.ip.whitelist	Project owner	Specifies an IP address whitelist for the project.	List of IP addresses separated by commas (,)

Attribute	Configured by	Description	Value range
odps.table.lifecycle	Project owner	<p>Optional: The lifecycle clause is optional in a table creation statement . If no lifecycle is set for a table, the table does not expire.</p> <p>Mandatory: The lifecycle clause is mandatory.</p> <p>Inherit: If you do not set the lifecycle, odps.table.lifecycle.value will be the lifecycle of this table.</p>	optional, mandatory, inherit
odps.table.lifecycle.value	Project owner	Indicates the default lifecycle . This value is configurable. Default value: 37, 231.	1–37231
odps.instance.remain.days	Project owner	Indicates how long the information of the instance is retained. Unit: days .	3–30
odps.task.sql.outerjoin.ppd	Project owner	Indicates whether the filtering conditions in full outer join are pushed down.	true, false

Attribute	Configured by	Description	Value range
odps.function.strictmode	Project owner	Indicates whether to return NULL (false) or report an error (true) when built-in functions have dirty data.	true, false
odps.task.sql.write.str2null	Project owner	Indicates whether to consider empty strings as NULL (true).	true, false

Export project meta

Command syntax:

```
export <projectname> <local_path>;
```

Purpose: It is used to export the meta of a project to a local file. Meta is represented by statements acceptable by odpscmd. The meta file can be used to recreate a project.

Show Flags

Command syntax:

```
show flags;
```

Purpose: It is used to display parameters configured by the set command.



Note:

The use project_name command purges the set command configuration.

1.5 MaxCompute SQL

1.5.1 Overview

1.5.1.1 Scenarios

This topic describes the scenarios of MaxCompute SQL.

MaxCompute SQL offline computing is applicable to scenarios where large volumes of data (terabytes) need to be processed, but do not have high real-time requirements. In such scenarios, it takes a relatively long time to prepare and submit each

job. MaxCompute SQL is not well-suited for businesses that require to process thousands of transactions per second. MaxCompute SQL online computing provides near real-time (NRT) processing capabilities.

MaxCompute SQL uses the syntax that is similar to SQL syntax. It can be considered as a subset of standard SQL. However, MaxCompute SQL is not equivalent to a database. It does not have common database characteristics, such as transactions, primary key constraints, and indexes. The maximum length of SQL statements currently supported by MaxCompute is 2 MB.

1.5.1.2 Reserved words

Keywords of SQL statements are reserved words in MaxCompute. Do not use reserved words to name tables, columns, or partitions. Otherwise, an error is returned. Reserved words are case-insensitive.

Common reserved words are listed as follows. For a complete list of reserved words, see [Reserved words](#).

```
% & && ( ) * + - . / ; < <= <>
= > >= ? ADD ALL ALTER
AND AS ASC BETWEEN BIGINT BOOLEAN BY
CASE CAST COLUMN COMMENT CREATE DESC DISTINCT
DISTRIBUTE DOUBLE DROP ELSE FALSE FROM FULL
GROUP IF IN INSERT INTO IS JOIN
LEFT LIFECYCLE LIKE LIMIT MAPJOIN NOT NULL
ON OR ORDER OUTER OVERWRITE PARTITION RENAME
REPLACE RIGHT RLIKE SELECT SORT STRING TABLE
THEN TOUCH TRUE UNION VIEW WHEN WHERE
```

1.5.1.3 Partitioned table

Partition columns provide many benefits, such as higher SQL operating efficiency and lower costs. However, too many partitions can cause problems. Using partition columns as filtering conditions in WHERE clauses of SELECT statements can bring greater benefits. Some SQL partition statements run inefficiently. For example, a statement fails when a large volume of data (more than 2,048 MB) is generated in dynamic partitions in a single MaxCompute instance.

It is easy to underestimate the number of partitions generated when multi-level partitions are used. When a huge number of partitions are generated, you must evaluate the original data to determine if there are excessive partitions.

You can create up to six levels of partitions. For some MaxCompute commands, the syntax differs between partitioned and non-partitioned tables. For more information, see [DDL statements](#) and [DML statements](#).

For more information about the table creation statement, see [Create a table](#).

1.5.1.4 Type conversion

1.5.1.4.1 Explicit type conversion

Explicit conversion uses CAST to convert a value type to another one. This topic describes explicit type conversion.

The following table lists explicit type conversions supported by MaxCompute SQL.

Table 1-9: Explicit type conversion

From/To	Bigint	Double	String	Datetime	Boolean	Decimal
Bigint	–	Y	Y	N	N	Y
Double	Y	–	Y	N	N	Y
String	Y	Y	–	Y	N	Y
Datetime	N	N	Y	–	N	N
Boolean	N	N	N	N	–	N
Decimal	Y	Y	Y	N	N	–

Y indicates that the type can be converted. N indicates that the type cannot be converted.



Note:

- When double type values are converted to bigint, the fractional is truncated. For example, `cast(1.6 as bigint) = 1`.
- When a string that meets double type requirements is converted to bigint, the string is first converted to the double type before it is converted to the bigint type. Hence, the fractional is truncated. For example, `cast("1.6" as bigint) = 1`.
- When a string that meets bigint type requirements is converted to the double type, one decimal is retained. For example, `cast("1" as double) = 1.0`.
- To convert a constant string to the decimal type, enclose the constant string within a pair of quotation marks. If the value is not enclosed in quotation marks

, it is treated as a double type value. For example, `cast("1.234567890123456789" as decimal)`.

- **Unsupported explicit type conversion operations cause an exception.**
- **If a conversion fails during execution, the system returns an error and exits.**
- **The datetime data conversion uses the default format `yyyy-mm-dd hh:mi:ss`. For more information, see [Convert data between string and datetime types](#).**
- **Some types cannot be explicitly converted, but can be converted using built-in SQL functions. For example, the `to_char` function can be used to convert boolean type values to the string type. For more information, see [TO_CHAR](#). The `to_date` function can be used to convert string type values to the datetime type. For more information, see [TO_DATE](#).**
- **For more information about CAST, see [CAST](#).**
- **When the values of the decimal type are out of the value range, the cast string to decimal operation may return an error, such as most significant bit overflow or least significant bit overflow truncation.**

1.5.1.4.2 Implicit type conversion and its scope

Implicit type conversion is an automatic type conversion performed by MaxCompute based on the context and a predefined set of rules. This topic describes the rules of implicit type conversion.

The following table lists implicit type conversion rules supported by MaxCompute.

Table 1-10: Implicit type conversion 1

From/To	BOOLEAN	TINYINT	SMALLINT	INT	BIGINT	FLOAT
BOOLEAN	T	F	F	F	F	F
TINYINT	F	T	T	T	T	T
SMALLINT	F	F	T	T	T	T
INT	F	F	F	T	T	T
BIGINT	F	F	F	F	T	T
FLOAT	F	F	F	F	F	T
DOUBLE	F	F	F	F	F	F
DECIMAL	F	F	F	F	F	F
STRING	F	F	F	F	F	F

From/To	BOOLEAN	TINYINT	SMALLINT	INT	BIGINT	FLOAT
VARCHAR	F	F	F	F	F	F
TIMESTAMP	F	F	F	F	F	F
BINARY	F	F	F	F	F	F

Table 1-11: Implicit type conversion 2

From/To	DOUBLE	DECIMAL	STRING	VARCHAR	TIMESTAMP	BINARY
BOOLEAN	F	F	F	F	F	F
TINYINT	T	T	T	T	F	F
SMALLINT	T	T	T	T	F	F
INT	T	T	T	T	F	F
BIGINT	T	T	T	T	F	F
FLOAT	T	T	T	T	F	F
DOUBLE	T	T	T	T	F	F
DECIMAL	F	T	T	T	F	F
STRING	T	T	T	T	F	F
VARCHAR	T	T	T	T	F	F
TIMESTAMP	F	F	T	T	T	F
BINARY	F	F	F	F	F	T

T indicates that the type conversion can be performed, while F indicates that the type conversion cannot be performed.

**Note:**

- An unsupported implicit type conversion will cause an exception.
- If the conversion fails, an error is returned.
- Implicit type conversion is automatically performed by MaxCompute based on context. If the types do not match, we recommend that you perform explicit type conversion using cast.
- The rules of implicit type conversion are applied to different specific scopes. In certain scenarios, only part of the rules will take effect. For more information, see the scope of implicit type conversions.

Implicit type conversion with relational operators

Relational operators include equal to (=), not equal to (<>), less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), IS NULL, IS NOT NULL, LIKE, RLIKE, and IN. The implicit conversion rules of LIKE, RLIKE, and IN are different from those of the other relational operators. These three operators are described in a separate section. The rules described in this section do not apply to these three operators. The following table lists implicit conversion rules when different types of data are involved in relational calculations.

Table 1-12: Implicit type conversion with relational operators

From/To	BIGINT	DOUBLE	STRING	DATETIME	BOOLEAN	DECIMAL
BIGINT	–	DOUBLE	DOUBLE	N	N	DECIMAL
DOUBLE	DOUBLE	–	DOUBLE	N	N	DECIMAL
STRING	DOUBLE	DOUBLE	–	DATETIME	N	DECIMAL
DATETIME	N	N	DATETIME	–	N	N
BOOLEAN	N	N	N	N	–	N
DECIMAL	DECIMAL	DECIMAL	DECIMAL	N	N	–



Note:

- If implicit type conversion is not supported between two values to be compared, the relational operation cannot be completed and an error is returned.
- For more information about relational operators, see [Relational operators](#).

Implicit conversion with special relational operators

Special relational operators are LIKE, RLIKE, and IN.

LIKE and RLIKE are used as follows:

```
source like pattern;  
source rlike pattern;
```

Note the following points for the two relational operators in implicit type conversion:

- The source and pattern parameters of LIKE and RLIKE must be of the string type.
- Other types are not supported by this operation and cannot be implicitly converted to the STRING type.

- If the value of source or pattern is NULL, the operation returns NULL.

IN is used as follows:

```
key in (value1, value2,...)
```

The implicit conversion rules of IN are as follows:

- The data types in the value list specified by IN must be consistent.
- If keys and values are compared, the BIGINT, DOUBLE, and STRING types compared are converted to DOUBLE, whereas the DATETIME and STRING types compared are converted to DATETIME. Conversion between other types is not allowed.

Note the following points for the IN operator:

The memory used by the compiler increases with the number of parameters used by the IN operation. An IN operation with 5,000 parameters consumes 17 GB of memory with the GCC compiler. We recommend that you limit the number of parameters to around 1,024. In this case, memory consumption will peak at 1 GB and compilation will only take 39 seconds.

Implicit type conversion with arithmetic operators

Arithmetic operators include plus (+), minus (-), multiplier (*), divider (/), and percent (%). The implicit conversion rules are as follows:

- Only the STRING, BIGINT, DECIMAL, and DOUBLE types can be used in arithmetic operations.
- Before an arithmetic operation, STRING values are implicitly converted to DOUBLE values.
- When an arithmetic operation involves values of both the BIGINT and DOUBLE types, BIGINT values are implicitly converted to DOUBLE values.
- The DATETIME and BOOLEAN types cannot be used in arithmetic operations.



Note:

For more information about arithmetic operators, see [Arithmetic operators](#).

Implicit conversion with logical operators

Logical operators include AND, OR, and NOT. The implicit conversion rules are as follows:

- Only the BOOLEAN type can be used in logic operations.
- The other types are not supported by logical operations or implicit type conversions.



Note:

For more information about logical operators, see [Logical operators](#).

1.5.1.4.3 SQL built-in functions

MaxCompute SQL provides a variety of system functions, which can be used to calculate one or more columns of any row and output any type of data.

The implicit conversion rules as follows:

- In a call of a function, if the data type of an input parameter is not consistent with the data type defined in the function, the data type of the input parameter is converted to the function-defined data type.
- The parameters of each built-in SQL function on MaxCompute can have different requirements for implicit type conversion. For more information, see [Built-in functions](#).

1.5.1.4.4 CASE WHEN

This topic describes the implicit conversion rules of CASE WHEN.

The implicit conversion rules of case when are as follows:

- If the returned data types are only bigint and double, they are converted to the double type.
- If data of the string type is also returned, all data types are converted to string. If a data type cannot be converted to string (for example, boolean), an error is returned.
- Conversion between other types is not allowed.

1.5.1.4.5 Partition column

MaxCompute SQL supports partitioned tables. For the definition of partitioned tables, see [DDL statements](#) and [DML statements](#). MaxCompute supports partitions of the following types: tinyint, smallint, int, bigint, varchar, and string.

1.5.1.4.6 UNION ALL

The data type, number of column, and column names involved in UNION ALL operation must all be consistent. Otherwise, an error is returned.

1.5.1.4.7 Conversion between string and datetime types

MaxCompute supports conversion between string and datetime types.

The format used in conversion is `yyyy-mm-dd hh:mi:ss.ff3`.

Table 1-13: Value ranges of units

Unit	String (case-insensitive)	Value range
Year	yyyy	0001-9999
Month	mm	01-12
Day	dd	01-28,29,30,31
Hour	hh	00-23
Minute	mi	00-59
Second	ss	00-59
ms	ff3	00-999



Note:

- Leading zeros cannot be omitted. For example, 2017-1-9 12:12:12 is an invalid string and cannot be converted into datetime. It must be written as 2017-01-09 12:12:12.
- Only strings that meet the preceding format requirements can be converted into datetime. For example, `cast("2017-12-31 02:34:34" as datetime)` converts the "2017-12-31 02:34:34" string into datetime. Similarly, when datetime is converted into strings, the default conversion format is `yyyy-mm-dd hh:mi:ss`. If you attempt to convert the following examples (or similar strings), the operation will fail and cause an exception.

```
cast("2017/12/31 02/34/34" as datetime)
```

```
cast("20171231023434" as datetime)  
cast("2017-12-31 2:34:34" as datetime)
```

MaxCompute provides the `to_date` function, which converts a string type that does not meet the datetime format into datetime type. For more information, see [TO_DATE](#).

1.5.2 Operators

1.5.2.1 Relational operators

This topic describes relational operators in MaxCompute SQL operators.

Table 1-14: Relational operators

Operator	Description
A=B	If A or B is NULL, NULL is returned. If A is equal to B, TRUE is returned. Otherwise, FALSE is returned.
A<>B	If A or B is NULL, NULL is returned. If A is not equal to B, TRUE is returned. Otherwise, FALSE is returned.
A<B	If A or B is NULL, NULL is returned. If A is less than B, TRUE is returned. Otherwise, FALSE is returned.
A<=B	If A or B is NULL, NULL is returned. If A is less or equal to B, TRUE is returned. Otherwise, FALSE is returned.
A>B	If A or B is NULL, NULL is returned. If A is greater than B, TRUE is returned. Otherwise, FALSE is returned.
A>=B	If A or B is NULL, NULL is returned. If A is greater than or equal to B, TRUE is returned. Otherwise, FALSE is returned.
A IS NULL	If A is NULL, TRUE is returned. Otherwise, FALSE is returned.
A IS NOT NULL	If A is not NULL, TRUE is returned. Otherwise, FALSE is returned.

Operator	Description
A LIKE B	<p>If A or B is NULL, NULL is returned. A is a string and B is the pattern to be matched. If A matches B, TRUE is returned. Otherwise, FALSE is returned. The percent sign (%) is a wildcard character that matches an arbitrary number of characters. The underscore (_) is a wildcard character that matches a single character. To use these two characters as ordinary characters, use backslashes to escape them: \% and _.</p> <p>'aaa'like 'a ' = TRUE'aaa'</p> <p>like'a%' = TRUE'aaa'like</p> <p>'aab' = FALSE'a%b'like</p> <p>'a\%b' = TRUE'axb'like</p> <p>'a\%b' = FALSE</p>
A RLIKE B	<p>If A or B is NULL, NULL is returned. A is a string and B is a string constant regular expression. If A matches B, TRUE is returned. Otherwise, FALSE is returned. If B is NULL, the system returns an error and exits.</p>
A IN B	<p>B is a set. If A is NULL, NULL is returned. If A is in B, TRUE is returned. Otherwise, FALSE is returned. If B contains only one element NULL, that is, A IN (NULL), NULL is returned. If B contains NULL, the type of NULL is considered the same as the other elements in B. B must be a constant and have at least one element. All elements must be of the same type.</p>

Double type values have variable precision. We recommend that you do not use the equal sign (=) to compare two double type values. You can subtract between two values of the double type, and then take the absolute value of the result for comparison. When the absolute value is negligible, the two values of the double type are considered equal. For example:

```
abs(0.9999999999 - 1.0000000000) < 0.0000000001
-- 0.9999999999 and 1.0000000000 have 10 decimal digits, while 0.
0000000001 has 9 decimal digits.
-- 0.9999999999 is considered equal to 1.0000000000.
```



Note:

- ABS is a built-in function provided by MaxCompute to take the absolute value of its input. For more information, see [ABS](#).
- A value of the double type in MaxCompute can retain 16 valid digits.

1.5.2.2 Arithmetic operators

This topic describes arithmetic operators in MaxCompute SQL operators.

Table 1-15: Arithmetic operators

Operator	Description
A + B	If A or B is NULL, NULL is returned. Otherwise, the result of A + B is returned.
A - B	If A or B is NULL, NULL is returned. Otherwise, the result of A - B is returned.
A * B	If A or B is NULL, NULL is returned. Otherwise, the result of A * B is returned.
A / B	If A or B is NULL, NULL is returned. Otherwise, the result of A / B is returned. If both A and B are of the bigint type, the result is of the double type.
A % B	If A or B is NULL, NULL is returned. Otherwise, the result of A % B is returned.
+A	A is returned.
-A	If A is NULL, NULL is returned. Otherwise, -A is returned.



Note:

- Only values of the string, bigint, double, and decimal types can be used in arithmetic operations. Values of the datetime and boolean types are not allowed in these operations.
- Before the operation, values of the string type are converted to the double type by implicit type conversion.
- When values of the bigint and double types are involved in an operation, values of the bigint type are converted to the double type by implicit type conversion first. The returned result is a value of the double type.
- When both A and B are of the bigint type, the returned result of A / B is a value of the double type. The returned results of the other arithmetic operations are values of the bigint type.

1.5.2.3 Bitwise operators

This topic describes bitwise operators in MaxCompute SQL operators.

Table 1-16: Bitwise operators

Operator	Description
A & B	Returns the bitwise AND result of A and B. For example, 1 & 2 returns 0 and 1 & 3 returns 1. The bitwise AND result of NULL in combination with another value is always NULL. A and B must be of the bigint type.
A B	Returns the bitwise OR result of A and B. For example, 1 2 returns 3 and 1 3 returns 3. The bitwise OR result of NULL in combination with another value is always NULL. A and B must be of the bigint type.



Notice:

Bitwise operators only support bigint type data and do not support implicit type conversion.

1.5.2.4 Logical operators

This topic describes logical operators in MaxCompute SQL operators.

Table 1-17: Logical operators

Operator	Description
A and B	TRUE and TRUE = TRUE
	TRUE and FALSE = FALSE
	FALSE and TRUE = FALSE
	FALSE and NULL = FALSE
	FALSE and FALSE = FALSE
	NULL and FALSE = FALSE
	TRUE and NULL = NULL
	NULL and TRUE = NULL
	NULL and NULL = NULL
A or B	TRUE or TRUE = TRUE
	TRUE or FALSE = TRUE
	FALSE or TRUE = TRUE
	FALSE or NULL = NULL

Operator	Description
	NULL or FALSE = NULL
	TRUE or NULL = TRUE
	NULL or TRUE = TRUE
	NULL or NULL = NULL
NOT A	If expression A is NULL, NULL is returned.
	If expression A is TRUE, FALSE is returned.
	If expression A is FALSE, TRUE is returned.

**Note:**

Only data of the boolean type can be involved in logic operations. These operations do not support implicit type conversion.

1.5.3 DDL statements

1.5.3.1 Table operations

1.5.3.1.1 Create a table

This topic describes how to run a DDL statement to create a table.

Command syntax:

```
create table [if not exists] table_name
[(col_name data_type [comment col_comment], ...)] [comment table_comment]
[partitioned by (col_name data_type [comment col_comment], ...)] [
lifecycle days]
[as select_statement]
create table [if not exists] table_name like existing_table_name
```

**Note:**

- Table names and column names are case-insensitive.
- If the command is run without the IF NOT EXISTS option and another table with the same name exists, an error is returned. With this option, a success is returned regardless of whether a table with the same name exists, even if the structure of the existing table is different from that of the table to be created. The metadata of the existing table does not change.
- Supported data types are bigint, double, boolean, datetime, decimal, string, Array < T >, and Map < T1, T2 >.

- A table name or column name cannot contain special characters. It can contain only lowercase English letters (a to z), uppercase English letters (A to Z), numbers, or underscores (_). A name must start with an English letter and cannot exceed 128 bytes.
- PARTITIONED BY specifies partition fields of the table. Only the string type is supported. A partition name cannot contain double-byte characters. It must start with an English letter (uppercase or lowercase) and contain English letters or numbers. A name cannot exceed 128 bytes. Supported special characters are space, colon (:), underscore (_), dollar sign (\$), pound sign (#), period (.), exclamation point (!), and at symbol (@). Other characters such as \t, \n, and forward slash (/) are considered as undefined characters. After you use partition fields to define partitions for a table, a full table scan will not be triggered when you add partitions, update partition data, or read partition data. This improves processing efficiency.
- A comment is a valid string within 1,024 bytes.
- Lifecycle indicates the lifecycle of the table in days. The CREATE TABLE LIKE statement does not replicate the lifecycle attribute from the source table.
- Theoretically, a source table can have up to six levels of partitions. Use as few partitions as possible to avoid extreme table expansion on storage.
- You can configure the maximum number of table partitions for a project. The default number is 60,000.

Example:

Create a table named `sale_detail` to store sales records. Use the `sale_date` and `region` columns of the table as partition columns.

```
create table if not exists sale_detail( shop_name string,  
customer_id string,  
total_price double)  
partitioned by (sale_date string,region string);  
-- Create a partitioned table named sale_detail.
```

You can also run the `create table ... as select ..` statement to create a table and replicate data to it:

```
create table sale_detail_ctas1 as select * from sale_detail;
```



Note:

If there is data in `sale_detail`, all the data is replicated to `sale_detail_ctas1`. Note that `sale_detail` is a partitioned table. The `create table ... as select ...` statement does not replicate its partition attribute to `sale_detail_ctas1`. Partition columns in `sale_detail` become ordinary columns in `sale_detail_ctas1`. Therefore, `sale_detail_ctas1` is a non-partitioned table with five columns.

In the `create table ... as select ...` statement, if you use constants as column values in the `SELECT` clause, we recommend that you specify column aliases:

```
create table sale_detail_ctas2 as select shop_name,  
customer_id, total_price,  
'2017' as sale_date,  
'China' as region from sale_detail;
```

**Note:**

If you do not specify column aliases, the fourth and fifth columns of `sale_detail_ctas3` created in the following example will have names automatically generated by the system, such as `_c3` and `_c4`.

```
create table sale_detail_ctas3 as select shop_name,  
customer_id, total_price, '2017',  
'China'  
from sale_detail;
```

In this case, to reference the columns in `sale_detail_ctas3`, you must include the column names in grave accents, as shown in the following example. If you run `select c3, _c4 from sale_detail_ctas3`, the system returns an error and exits. MaxCompute SQL does not support column names starting with an underscore (`_`). You must enclose column names in grave accents. We recommend that you use aliases to avoid this problem.

```
select ` _c3`, ` _c4` from sale_detail_ctas3;
```

Run the following `create table ... like` statement to create a table with the same structure as the source table:

```
create table sale_detail_like like sale_detail;
```

**Note:**

The structure of `sale_detail_like` is exactly the same as that of `sale_detail`. Both tables have the same attributes, such as the column name, column comment, and

table comment, except for lifecycle. However, data in sale_detail is not replicated to sale_detail_like.

1.5.3.1.2 Delete a table

This topic describes how to run a DDL statement to delete a table.

Command syntax:

```
drop table [if exists] table_name;
```



Note:

If the command is run without the IF EXISTS option and the table does not exist, an exception is returned. With this option, a success is returned regardless of whether the table exists.

Example:

```
create table sale_detail_drop like sale_detail; drop table sale_detail_drop;
-- If the table exists, a success is returned. If not, an exception is returned.
drop table if exists sale_detail_drop2;
-- A success is returned regardless of whether sale_detail_drop2 exists.
```

1.5.3.1.3 Rename a table

This topic describes how to run a DDL statement to rename a table.

Command syntax:

```
alter table table_name rename to new_table_name;
```



Note:

- The rename operation only changes the table name, not the table data.
- If the table specified by new_table_name already exists, an error is returned.
- If the table specified by table_name does not exist, an error is returned.

Example:

```
create table sale_detail_rename1 like sale_detail;
```

```
alter table sale_detail_rename1 rename to sale_detail_rename2;
```

1.5.3.1.4 Modify the comment of a table

This topic describes how to run a DDL statement to modify the comment of a table.

Command syntax:

```
alter table table_name set comment 'tbl comment';
```



Note:

- **table_name** must be an existing table.
- A comment can contain a maximum of 1,024 bytes.

Example:

```
alter table sale_detail set comment 'new coments for table sale_detail';
```

You can run the desc command to view the modified comment in the table. For more information, see [Obtain table information](#).

1.5.3.1.5 Modify the lifecycle of a table

MaxCompute provides the lifecycle management function to release storage space and simplify the data clearance process. This topic describes how to run a DDL statement to modify the lifecycle of a table.

Command syntax:

```
alter table table_name set lifecycle days;
```



Note:

- The days parameter indicates the lifecycle of a table. Unit: days. It must be a positive integer.
- If the table specified by table_name is a non-partitioned table, and is not modified in the period specified by the days parameter since the last modification date, MaxCompute automatically clears the table (similar to the DROP TABLE operation). In MaxCompute, the LastDataModifiedTime value of a table is updated each time data in the table is modified. MaxCompute determines whether to clear a table based on its LastDataModifiedTime and lifecycle settings.

- If the table specified by `table_name` is a partitioned table, MaxCompute determines whether to clear each partition based on the `LastDataModifiedTime` value. Unlike non-partitioned tables, a partitioned table is not deleted after the last partition is reclaimed.
- You can configure a lifecycle for tables, but not for partitions.
- You can specify a lifecycle when creating a table.

Example:

```
create table test_lifecycle(key string) lifecycle 100;  
-- Create a table named test_lifecycle with a lifecycle of 100 days.  
alter table test_lifecycle set lifecycle 50;  
-- Change the lifecycle of the test_lifecycle table to 50 days.
```

1.5.3.1.6 Disable the lifecycle

In special cases, you may not want certain partitions to be automatically recycled by the lifecycle function. In such cases, you can disable the lifecycle function for the partition. This topic describes how to run a DDL statement to disable the lifecycle function.

Command syntax:

```
ALTER TABLE table_name partition[partition_spec] ENABLE|DISABLE  
LIFECYCLE;
```

Example:

```
ALTER TABLE trans PARTITION(dt='20141111') DISABLE LIFECYCLE;
```

1.5.3.1.7 Modify the LastDataModifiedTime value of a table

MaxCompute SQL supports the `TOUCH` operation, which allows you to modify the `LastDataModifiedTime` value of a table. This operation changes the `LastDataModifiedTime` value of a table to the current time. This topic describes how to run a DDL statement to modify the `LastDataModifiedTime` value of a table.

Command syntax:

```
alter table table_name touch;
```

**Note:**

- If the specified `table_name` does not exist, an error is returned.

- This operation modifies the `LastDataModifiedTime` value of the table. In this case, MaxCompute considers a change to the table data, and recalculates the lifecycle.

For more information about how to modify the `LastDataModifiedTime` value of a partition, see [Modify the `LastDataModifiedTime` value of a partition](#).

1.5.3.1.8 Clear data from a non-partitioned table

This topic describes how to run a DDL statement to clear data from a non-partitioned table.

Command syntax:

```
TRUNCATE TABLE table_name;
```



Note:

This statement is used to clear data from a specified non-partitioned table. To clear data from a partitioned table, run the `ALTER TABLE table_name DROP PARTITION (partition_spec) statement`.

1.5.3.1.9 Archive table data

This topic describes how to run a DDL statement to archive the data of a table.

If a project does not have enough space, you can use the table archiving feature in MaxCompute to compress data by about 50%. The archiving feature uses a compression algorithm with a higher compression ratio. It saves data as redundant array of independent disks (RAID) files. Data is no longer simply stored in three copies. Instead, six copies and three check blocks are maintained to increase the effective storage ratio from 1:3 to 1:1.5. The archive feature consumes only half of the usual physical space.

However, this feature comes at a price. If a data block or machine is damaged, the time required to restore the data is longer, and the read performance is affected. Therefore, this feature is suitable for compressing cold data for storage. For example, you can store large volumes outdated log data as RAID files for a long time.

Command syntax:

```
ALTER TABLE [table_name] <PARTITION(partition_name='partition_value')> ARCHIVE;
```

Example:

```
alter table my_log partition(ds='20170101') archive;
```

Command output:

```
Summary:
table name: test0128 /pt=a instance count: 1 run time: 21
before merge, file count: 1 file size: 456 file physical size: 1368
after merge, file count: 1 file size: 512 file physical size: 768
```

**Note:**

The output shows the changes in logical size and physical size during the archiving process. In the archiving process, multiple small files are automatically merged. After the archive operation is complete, you can run the `desc extended` command to check whether the data in the partition has been archived, and view the physical space usage:

```
desc extended my_log partition(ds='20170101');
+-----+
+
PartitionSize: 512 |
+-----+
+
CreateTime: 2017-01-28 07:05:20 |
LastDDLTime: 2017-01-28 07:05:20 |
LastModifiedTime: 2017-01-28 07:05:21 |
+-----+
+
```

1.5.3.1.10 Forcibly delete data from a table (partition)

If you need to forcibly and irrecoverably delete data from a table or partition to immediately release storage space, you can perform the deletion operation with the **PURGE** option. This topic describes how to run a DDL statement to forcibly delete data from a table (partition).

Command syntax:

```
DROP TABLE tblname PURGE;
```

```
ALTER TABLE tblname DROP PARTITION(part_spec) PURGE;
```

Example:

```
drop table my_log purge;  
alter table my_log drop partition (ds='20170618') purge;
```

1.5.3.2 View-based operation

1.5.3.2.1 Create a view

This topic describes how to run a DDL statement to create a view.

Command syntax:

```
create [or replace] view [if not exists] view_name  
[(col_name [comment col_comment], ...)]  
[comment view_comment]  
[as select_statement]
```

**Note:**

- To create a view, you must have read permissions on the table referenced by the view. Views in MaxCompute are not materialized views. View operations involve accessing data of referenced tables. Note that changes to your permission on the referenced table can result in changes to your permission on the view.
- A view can contain only one valid SELECT statement.
- A view can reference other views but cannot reference itself. Circular reference is not supported.
- You cannot write data to a view. For example, the INSERT INTO and INSERT OVERWRITE operations do not work on views.
- If the table referenced by a view changes, you may no longer be able to access the view. For example, a view becomes inaccessible after the table it references is deleted. You must maintain the mappings between referenced tables and views properly.
- If the CREATE VIEW statement is run without the IF NOT EXISTS option and the view already exists, an exception is returned. In this case, you can run the CREATE VIEW or REPLACE VIEW statement to recreate a view. The permissions on the recreated view remain unchanged.

Example:

```
create view if not exists sale_detail_view  
(store_name, customer_id, price, sale_date, region)  
comment 'a view for table sale_detail'
```

```
as select * from sale_detail;
```

1.5.3.2.2 Delete a view

This topic describes how to run a DDL statement to delete a view.

Command syntax:

```
drop view [if exists] view_name;
```



Note:

If the command is run without the IF EXISTS option and the view does not exist, an error is returned.

Example:

```
drop view if exists sale_detail_view;
```

1.5.3.2.3 Rename a view

This topic describes how to run a DDL statement to rename a view.

Command syntax:

```
alter view view_name rename to new_view_name;
```



Note:

If a view with the same name already exists, an error is returned.

Example:

```
create view if not exists sale_detail_view  
(store_name, customer_id, price, sale_date, region)  
comment 'a view for table sale_detail'  
as select * from sale_detail;  
alter view sale_detail_view rename to market;
```

1.5.3.3 Column and partition operations

1.5.3.3.1 Add a partition

This topic describes how to add a partition by using a DDL statement.

Command syntax:

```
alter table table_name add [if not exists] partition partition_spec
```

```
partition_spec:(partition_col1 = partition_col_value1, partition_col2  
= partition_col_value2, ...)
```

**Note:**

- If the command is run without the IF NOT EXISTS option and another partition with the same name exists, an error is returned.
- You can create up to 60,000 partitions in a single table in MaxCompute.
- To add a partition to a table that has multi-level partitions, you must specify all partition values.

Example:

The following examples add new partitions to the sale_detail table.

```
alter table sale_detail add if not exists partition (sale_date='201712  
' , region='hangzhou');  
-- A partition is added. This partition stores the sales records of  
the Hangzhou region in December 2017.  
alter table sale_detail add if not exists partition (sale_date='201712  
' , region='shanghai');  
-- A partition is added. This partition stores the sales records of  
the Shanghai region in December 2017.  
alter table sale_detail add if not exists partition(sale_date='  
20171011');  
-- The command specifies only the sale_date partition, so an error is  
returned.  
alter table sale_detail add if not exists partition(region='shanghai  
' );  
-- The command specifies only the region partition, so an error is  
returned.
```

1.5.3.3.2 Delete a partition

This topic describes how to run a DDL statement to delete a partition.

Command syntax:

```
alter table table_name drop [if exists] partition_spec;  
partition_spec:: (partition_col1 = partition_col_value1, partition_  
col2 = partition_col_value2, ...)
```

**Note:**

If the command is run without the IF EXIST option and the partition does not exist, an error is returned.

Example:

Run the following command to delete a partition from the sale_detail table.

```
alter table sale_detail drop partition(sale_date='201712',region='hangzhou');  
-- The sales records of Hangzhou in December 2017 are successfully deleted.
```

1.5.3.3.3 Add a column

This topic describes how to add a column by using a DDL statement.

Command syntax:

```
alter table table_name add columns (col_name1 type1, col_name2 type2  
...)
```



Note:

- A column can only be one of the following types: bigint, double, boolean, datetime, decimal, string, tinyint, smallint, int, float, varchar, binary, timestamp, array, map, or struct.
- You can create up to 1,200 columns in a single table in MaxCompute.

1.5.3.3.4 Change a column name

This topic describes how to run a DDL statement to change a column name.

Command syntax:

```
alter table table_name change column old_col_name rename to new_col_name;
```



Note:

- You must specify an existing column for old_col_name.
- You cannot name a column in the table new_col_name.

1.5.3.3.5 Modify the comment of a column or partition

This topic describes how to run a DDL statement to modify the comment of a column or partition.

Command syntax:

```
alter table table_name change column col_name comment 'comment';
```



Note:

- The comment cannot exceed 1,024 bytes.
- The data type and position of a column cannot be changed.

1.5.3.3.6 Modify the LastDataModifiedTime value of a partition

MaxCompute SQL supports the TOUCH operation, which allows you to modify the LastDataModifiedTime value of a partition. This operation changes the LastDataModifiedTime value of a partition to the current time. This topic describes how to run a DDL statement to modify the LastDataModifiedTime value of a partition.

Command syntax:

```
alter table table_name touch partition(partition_col='partition_col_value', ...);
```



Note:

- If the specified table_name or partition_col does not exist, an error is returned.
- If the specified partition_col_value does not exist, an error is returned.
- This operation modifies the LastDataModifiedTime value of the table. In this case, MaxCompute considers a change to the table or partition value, and recalculates the lifecycle.

For more information about how to modify the LastDataModifiedTime value of a table, see [Modify the LastDataModifiedTime value of a table](#).

1.5.3.3.7 Modify partition values

MaxCompute SQL provides the RENAME operation, which allows you to modify partition values of a table. This topic describes how to run a DDL statement to modify partition values.

Command syntax:

```
ALTER TABLE table_name PARTITION (partition_col1 = partition_col_value1, partition_col2 = partition_col_value2, . . .)  
RENAME TO PARTITION (partition_col1 = partition_col_newvalue1, partition_col2 = partition_col_newvalue2, ...);
```



Note:

- This command cannot modify the names of partition columns. It can only modify the values of the columns.

- To modify the values in one or more partitions in the case of multi-level partitions, you must specify values of partitions at each level.

1.5.4 DML statements

1.5.4.1 INSERT statement

1.5.4.1.1 Update the data of a table

This topic describes how to run an INSERT statement to update the data of a table.

The INSERT OVERWRITE and INSERT INTO statements are commonly used for data processing in MaxCompute SQL. They are used to save the computing results in the target table for the next computing. The INSERT INTO statement adds data to a table or partition. The INSERT OVERWRITE statement clears the original data before inserting data to a table or partition.

Command syntax:

```
insert overwrite|into table tablename [partition (partcol1=val1,  
partcol2=val2 ...)] select_statement  
from from_statement;
```



Note:

The INSERT syntax in MaxCompute is different from that in MySQL or Oracle. In MaxCompute, INSERT OVERWRITE or INSERT INTO must be followed by the keyword TABLE, not directly by the table name.

Example:

The following example calculates the sales of different regions in the sale_detail table.

```
create table sale_detail_insert like sale_detail;  
alter table sale_detail_insert add partition(sale_date='2017', region  
='china');  
insert overwrite table sale_detail_insert partition (sale_date='2017  
, region='china') select shop_name, customer_id, total_price from  
sale_detail;
```



Note:

When data is updated using an INSERT operation, the mapping between the source and target tables depends on the column sequence in the SELECT clause, instead of the mapping of column names between both tables.

The following statement is also valid:

```
insert overwrite table sale_detail_insert partition (sale_date='2017', region='china')
select customer_id, shop_name, total_price from sale_detail;
-- When the sale_detail_insert table is created, the column sequence
is shop_name string, customer_id string, and total_price bigint.
-- When data in sale_detail is inserted to sale_detail_insert, the
insertion sequence is customer_id, shop_name, and total_price.
-- In this case, data in sale_detail.customer_id is inserted into
sale_detail_insert.shop_name.
-- Data in sale_detail.shop_name is inserted into sale_detail_insert.
customer_id.
```

When data is inserted into a partitioned table, the partition columns cannot appear in the SELECT list.

```
insert overwrite table sale_detail_insert partition (sale_date='2017', region='china')
select shop_name, customer_id, total_price, sale_date, region from sale_detail;
-- An error is returned, because partition columns (sale_date and region)
cannot appear in an INSERT statement for a static partition.
```

1.5.4.1.2 Output data to multiple objects

This topic describes how to run the INSERT statement to output data to multiple objects.

MaxCompute SQL allows you to insert data to different result tables or partitions by using one SQL statement.

Command syntax:

```
from from_statement
insert overwrite | into table tablename1 [partition (partcol1=val1, partcol2=val2 ...)] select_statement1
[insert overwrite | into table tablename2 [partition ...] select_statement2]
```



Note:

- A SQL statement typically supports up to 256 outputs. A syntax error is returned if more than 256 outputs are specified.
- In a MULTI INSERT statement, you can specify a target partition in a partitioned table or specify a non-partitioned table only once.
- The INSERT OVERWRITE and INSERT INTO operations cannot be performed simultaneously on different partitions in a partitioned table. Otherwise, an error is returned.

Example:

```
create table sale_detail_multi like sale_detail;
from sale_detail
insert overwrite table sale_detail_multi partition (sale_date='2016',
region='china' ) select shop_name, customer_id, total_price
insert overwrite table sale_detail_multi partition (sale_date='2017',
region='china' ) select shop_name, customer_id, total_price;
-- A success is returned. Data of the sale_detail table is inserted
into the sale records of the China region in 2016 and 2017 in the
sales table.
from sale_detail
insert overwrite table sale_detail_multi partition (sale_date='2017',
region='china' ) select shop_name, customer_id, total_price
insert overwrite table sale_detail_multi partition (sale_date='2017',
region='china' ) select shop_name, customer_id, total_price;
-- An error is returned. The same partition appears more than once.
from sale_detail
insert overwrite table sale_detail_multi partition (sale_date='2016',
region='china' )
select shop_name, customer_id, total_price
insert into table sale_detail_multi partition (sale_date='2017',
region='china' ) select shop_name, customer_id, total_price;
-- An error is returned. The INSERT OVERWRITE and INSERT INTO
operations cannot be performed simultaneously on different partitions
in a partitioned table.
```

1.5.4.1.3 Output data to a dynamic partition

This topic describes how to use the INSERT statement to output data to a dynamic partition.

When you run the INSERT OVERWRITE statement on a partitioned table, you can specify the partition values in the statement. Another flexible method is to specify partition column names instead of setting partition values. In the meantime, specify the partition values in the corresponding columns of a SELECT clause.

Command syntax:

```
insert overwrite table tablename partition (partcol1, partcol2 ...)
select_statement from from_statement;
```

**Note:**

- When you run a SQL dynamic partition statement in a distributed environment, a single process can output up to 512 dynamic partitions. If the number of dynamic partitions exceeds this limit, an exception is returned.
- Currently, a SQL dynamic partition statement can generate up to 2,000 dynamic partitions. If the number of dynamic partitions exceeds this limit, an exception is returned.

- The dynamic partition values cannot be NULL. Otherwise, an exception is returned.
- If a target table has multi-level partitions, you can specify some partitions as static partitions in an INSERT statement. However, the static partitions must be high-level partitions.

Example:

```
create table total_revenues (revenue bigint) partitioned by (region
string); insert overwrite table total_revenues partition(region)
select total_price as revenue, region from sale_detail;
```

**Note:**

In the preceding example, you do not know which partitions are generated before running the SQL statement. The partitions generated are determined by the value of the region field after the execution of the SELECT statement. This is why the partitions are called dynamic partitions.

Other examples:

```
create table sale_detail_dypart like sale_detail;
insert overwrite table sale_detail_dypart partition (sale_date, region
) select * from sale_detail;
-- A success is returned.
insert overwrite table sale_detail_dypart partition (sale_date='
2017', region) select shop_name,customer_id,total_price,region from
sale_detail;
-- A success is returned. The table has multi-level partitions.
Specify a primary partition.
insert overwrite table sale_detail_dypart partition (sale_date='2017
', region) select shop_name,customer_id,total_price from sale_detail;
-- An error is returned. The inserted dynamic partition must be in the
SELECT list.
insert overwrite table sales partition (region='china', sale_date)
select shop_name,customer_id,total_price,region from sale_detail;
-- An error is returned. You cannot specify only low-level partitions
when dynamically inserting high-level partitions.
```

1.5.4.2 SELECT statement

1.5.4.2.1 SELECT operation

This topic describes how to use the SELECT statement.

Command syntax:

```
select [all | distinct] select_expr, select_expr, ... from table_refe
rence
[where where_condition] [group by col_list]
[order by order_condition]
```

```
[distribute by distribute_condition [sort by sort_condition] ] [limit number]
```

Note the following when using select statements:

- A **SELECT** operation reads data from a table. You can specify names of the columns to read or use ***** to represent all columns in the statement.

Example:

```
select * from sale_detail;  
-- Read all columns in the sale_detail table.  
select shop_name from sale_detail;  
-- Read only the shop_name name in the sale_detail table.
```



Note:

Currently, the **SELECT** statement can only return up to 1,000 rows of results. If the **SELECT** statement serves as a clause, it does not have such a restriction. The **SELECT** clause returns all results to the outer query. To obtain more than 1,000 rows of results through the **SELECT** operation, you must use Tunnel to download the entire table or a temporary table returned by a **SELECT** operation. For more information, see [Use Tunnel](#).

- You can use a **WHERE** clause to apply filtering conditions.

Example:

```
select * from sale_detail where shop_name like 'hang%';
```

The following table lists filter conditions supported by the **WHERE** clause.

Table 1-18: Filter conditions

Filter condition	Description
>, <, =, >=, <=, <>	/
like, rlike	/

Filter condition	Description
in, not in	If a sub-query is added following condition 'in/not in', then it can only return one column result and the quantity of returned values cannot exceed 1,000.

You can specify a partition range in WHERE clause of SELECT statement to avoid a full table scan.

Example:

```
select sale_detail.* from sale_detail
where sale_detail.sale_date >= '2015' and sale_detail.sale_date <= '2017';
```



Note:

WHERE clauses of MaxCompute SQL statements do not support queries with between conditions, and can have no more than 256 conditions.

- Nested subqueries are supported in table_reference.

Example:

```
select * from (select region from sale_detail) t where region = 'shanghai';
```

- **distinct:** If there are repeated rows, then use distinct in front of the field to remove the duplicated value, and only one value will be returned. Or use all to return all repeated values. Without the distinct option, the statement returns all duplicate values, same as the result obtained with the ALL option.

Example:

```
select distinct region from sale_detail;
select distinct region, sale_date from sale_detail;
-- The distinct option applies to multiple columns. The option takes effect on all columns of a select option, rather than a single column.
```

- **GROUP BY:** a grouping query clause, usually used with aggregate functions . When the SELECT statement contains an aggregate function, the key of the GROUP BY statement can be the name of a column in the input table or the

expression composed of input table columns. However, it cannot be the output column of the SELECT statement.

Example:

```
select region from sale_detail group by region;
-- Runs successfully with the name of a column in the input table
directly used as the group by column
select sum(total_price) from sale_detail group by region;
-- Runs successfully with the table grouped by the region value and
returns the total sales of each group
Select region, sum (total_price) from sale_detail group by region;
-- Runs successfully with the table grouped by the region value and
returns the region value (unique in the group) and total sales of
each group
select region as r from sale_detail group by r;
-- Runs with the alias of the Select column and returns an error
select 'China-' + region as r from sale_detail group by 'China-' +
region;
-- Requires a complete expression of the column
Select region, total_price from sale_detail group by region;
-- Returns an error; all columns not using an aggregate function in
the Select statement must exist in group by
select region, total_price from sale_detail group by region,
total_price;
-- Runs successfully
```



Note:

The preceding limits are imposed for the following reason: SQL parses the GROUP BY operation prior to the SELECT operation, and therefore the GROUP BY statement can only use the column or expression of the input table as the key. For more information about aggregate functions, see [Aggregate functions](#).

- **ORDER BY:** Order all data globally based on specified columns. To order records in descending order, use the DESC keyword. For global sorting, ORDER BY must be used together with LIMIT. In the ORDER BY operation, NULL is considered smaller than any value. This is consistent with MySQL, but is not consistent with Oracle. Unlike GROUP BY, ORDER BY must be followed by the aliases of the SELECT columns. If the SELECT operation is performed on a column and the column alias is not specified, the column name is used as the column alias.

Example:

```
select * from sale_detail order by region;
-- Returns an error because order by is not used together with limit
select * from sale_detail order by region limit 100;
select region as r from sale_detail order by region;
-- An error is returned because ORDER BY is not followed by a column
alias.
```

```
select region as r from sale_detail order by r;
```

**Note:**

The number in [limit number] is a constant that limits the number of returned rows. If a SELECT statement is run without the LIMIT option, it can return at most 5000 rows on screen. The screen display limit may vary with projects and can be set in the console.

- **DISTRIBUTE BY:** Shard data based on hash values of specified columns, where the alias of SELECT output columns must be used.

Example:

```
select region from sale_detail distribute by region;
-- Runs successfully because the column name is an alias
select region as r from sale_detail distribute by region;
-- Returns an error because DISTRIBUTE BY is not followed by a
column alias
select region as r from sale_detail distribute by r;
```

- **Sort by:** for partial sorting, DISTRIBUTE BY must be added in front of the statement. sort by is used to partially sort the results of distribute by. It must use the alias of the SELECT output column.

Example:

```
select region from sale_detail distribute by region sort by region;
select region as r from sale_detail sort by region;
-- The statement returns an error and exits because it does not
follow a DISTRIBUTE BY statement.
```

- **ORDER BY and GROUP BY** cannot be used together with DISTRIBUTE BY/SORT BY, and must use the alias of SELECT output columns.

**Note:**

- The key of ORDER BY, SORT BY, or DISTRIBUTE BY must be the output column of a SELECT statement, that is, the column alias.
- In MaxCompute SQL parsing, ORDER BY, SORT BY, and DISTRIBUTE BY come after the SELECT operation. Therefore, they can only accept the output columns of the SELECT statement as keys.

1.5.4.2.2 Subquery

This topic describes how to use the SELECT statement for subquery operations.

A common SELECT statement reads data from multiple tables, for example, select column_1, column_2 ... from table_name. The query object can be another SELECT operation, which is a subquery.

Command syntax:

```
select * from (select shop_name from sale_detail) a;
```



Notice:

A subquery must have an alias.

Example:

```
create table shop as select * from sale_detail;  
select a.shop_name, a.customer_id, a.total_price from  
(select * from shop) a join sale_detail on a.shop_name = sale_detail.  
shop_name;
```



Note:

In a FROM clause, a subquery can be used as a table, which supports a JOIN operation with other tables or subqueries.

1.5.4.3 UNION statements

1.5.4.3.1 UNION ALL

This topic describes how to run the SELECT statement to perform the UNION ALL operation.

Command syntax:

```
select_statement union all select_statement
```



Note:

Combine two or more datasets returned from SELECT operations into one data set. If repeated rows exist in the result, all rows that meet the condition are returned, with duplicated rows retained.

MaxCompute SQL does not support combination of two top-level query results. To combine them, they must be rewritten into a subquery format.

Example of incorrect format before rewriting:

```
Select * From sale_detail where region = 'Hangzhou'
union all
select * from sale_detail where region = 'shanghai';
```

Example of correct format after rewriting:

```
select * from (
select * from sale_detail where region = 'hangzhou' union all
select * from sale_detail where region = 'shanghai') t;
```

**Notice:**

- **For a UNION ALL operation, all subqueries must have the same number of columns, column names, and column types. If the column names are inconsistent, use a column alias.**
- **Generally, MaxCompute allows a UNION ALL operation for a maximum of 256 subqueries. A syntax error is returned if the limit is exceeded.**

1.5.4.4 JOIN statement

1.5.4.4.1 JOIN operation

This topic describes how to use the JOIN statement.

In MaxCompute, JOIN supports multiple connections, but not Cartesian products (JOIN without the ON condition).

Command syntax:

```
join_table:
table_reference join table_factor [join_condition]
| table_reference {left outer|right outer|full outer|inner} join
table_reference join_condition
table_reference: table_factor
join_table
table_factor: tbl_name [alias]
table_subquery alias
( table_references )
join_condition:
on equality_expression ( and equality_expression )*
```

**Note:**

'Equality_expression' is an equality expression.

Note the following when using the JOIN statement:

- **LEFT OUTER JOIN:** Returns all records in the left table (shop in the example below), even if these records have no matching rows in the right table (sale_detail in the example below).

Example:

```
select a.shop_name as ashop, b.shop_name as bshop from shop a left
outer join sale_detail b on a.shop_name=b.shop_name;
-- Both the shop and sale_detail tables have the shop_name column
. Aliases are assigned to the columns in the SELECT clause to
distinguish the columns.
```

- **RIGHT OUTER JOIN:** Returns all records in the right table (sale_detail in the example below), even if these records have no matching rows in the left table (shop in the example below).

Example:

```
select a.shop_name as ashop, b.shop_name as bshop from shop a right
outer join sale_detail b on a.shop_name=b.shop_name;
-- Both the shop and sale_detail tables have the shop_name column
. Aliases are assigned to the columns in the SELECT clause to
distinguish the columns.
```

- **FULL OUTER JOIN:** Returns all records in both the left and right tables.

Example:

```
select a.shop_name as ashop, b.shop_name as bshop from shop a full
outer join sale_detail b on a.shop_name=b.shop_name;
```

- **inner join:** if there is at least one matching record in the table, 'inner join' will return the record. The keyword 'inner' can be omitted.

Example:

```
select a.shop_name from shop a inner join sale_detail b on a
.shop_name=b.shop_name; select a.shop_name from shop a join
sale_detail b on a.shop_name=b.shop_name;
```

- **Connection condition:** only equal conditions connected by 'and' are allowed, and at most 16 JOIN operations are supported. MAPJOIN allows unequal connections or multiple conditions connected by the OR operator.

Example:

```
select a.* from shop a full outer join sale_detail b on a.shop_name=
b.shop_name full outer join sale_detail c on a.shop_name=c.shop_name
;
-- A JOIN operation supports a maximum of 16 connections.
select a.* from shop a join sale_detail b on a.shop_name <> b.
shop_name;
```

```
-- An error is returned because unequal JOIN conditions are not supported.
```

1.5.4.4.2 MAPJOIN HINT

This topic describes how to use the MAPJOIN statement to join a large table with one or more small tables.

The MAPJOIN operation is faster than common JOIN operations.

The basic principle of MAPJOIN is that when the volume of data is small, SQL loads all of the specified small tables in to the program memory through the JOIN operation to accelerate the execution process.

Example:

```
select /* + mapjoin(a) */ a.shop_name, b.customer_id, b.total_price  
from shop a join sale_detail b  
on a.shop_name = b.shop_name;
```



Notice:

Note the following points when you use the MAPJOIN statement:

- The left table of a LEFT OUTER JOIN clause must be a large table.
- The right table of a RIGHT OUTER JOIN clause must be a large table.
- Both the left and right tables of an INNER JOIN clause can be large tables.
- MAPJOIN cannot be used in a FULL OUTER JOIN clause.
- MAPJOIN supports small tables as subqueries.
- When MAPJOIN is used and a small table or subquery must be referenced, the alias must be referenced.
- MAPJOIN can use non-equivalent JOIN conditions and combine multiple conditions by using OR statements.
- MaxCompute can specify up to 256 small tables in a MAPJOIN statement.
- If MAPJOIN is used, the total memory occupied by all of the small tables cannot exceed 512 MB. However, you can use the `odps.sql.mapjoin.memory.max` parameter to adjust this limit to 2,048 MB.

The limit here refers to the original size of data. If you run the desc command to obtain the compressed size, you must multiply it by the compression ratio.

MaxCompute SQL does not support complex JOIN conditions, such as non-equivalent expressions and the OR logic, in the ON conditions of common JOIN operations. However, MAPJOIN supports such operations.

```
select /*+ mapjoin(a) */ a.total_price, b.total_price
from shop a join sale_detail b
on a.total_price < b.total_price or a.total_price + b.total_price <
500;
```

1.5.4.5 EXPLAIN statement

This topic describes the EXPLAIN statement in DML statements of MaxCompute SQL.

MaxCompute SQL provides the EXPLAIN operation, which displays the description of the ultimate execution plan structure of DML statements. An execution plan is the program that is ultimately used to execute SQL semantics.

Command syntax:

```
EXPLAIN <DMLquery>;
```



Note:

The execution result of an EXPLAIN statement includes the following:

- Dependencies between all the jobs of this DML statement.
- Dependencies between all the tasks of each job.
- All operator dependency structures in a task.

Example:

```
EXPLAIN
SELECT abs(a.key), b.value FROM src a JOIN src1 b ON a.value = b.value
;
```

The EXPLAIN statement output includes the following:

- The first part is the dependency between jobs.

Command output:

```
job0 is root job
```



Note:

Because this query only needs one job (job0), only one line of information is needed.

- The second part is the dependency between tasks.

Command output:

```
In Job job0:
root Tasks: M1_Stg1, M2_Stg1
J3_1_2_Stg1 depends on: M1_Stg1, M2_Stg1
```



Note:

- Job0 contains three tasks, among which M1_Stg1 and M2_Stg1 are executed first, and J3_1_2_Stg1 is executed after the first two tasks are finished.
 - Naming rules for tasks: MaxCompute provides four task types: MapTask, ReduceTask, JoinTask, and LocalWork. The first letter of a task name indicates the type of the current task (for example, M2Stg1 is a MapTask). The number immediately following the first letter represents the current Task ID, which is unique among all tasks in the current query. The numbers separated by underscores (_) represent the immediate dependencies of the current task. For example, J3_1_2_Stg1 means that the current task (ID 3) is dependent on tasks with ID 1 and ID 2.
- The third part is the operator structure in the tasks, where each operator string describes the execution semantics of a task.

Command output:

```
In Task M1_Stg1:
Data source: yudi_2.src #### "Data source" describes the input
content of the current task TS: alias: a #### TableScanOperator
RS: order: + #### ReduceSinkOperator keys:
a.value values:
a.key partitions:
a.value
In Task J3_1_2_Stg1:
JOIN: a INNER JOIN b #### JoinOperator
SEL: Abs(UDFToDouble(a._col0)), b._col5 #### SelectOperator FS:
output: None #### FileSinkOperator
In Task M2_Stg1:
Data source: yudi_2.src1 TS: alias: b
RS: order: + keys:
b.value values:
b.value partitions:
```

b.value

The meanings of the operators are shown as below.

Table 1-19: Operators

Operator	Description
TableScanOperator	Describes the logic of FROM statement blocks in a query statement. The input table name (alias) is displayed in the EXPLAIN results.
SelectOperator	Describes the logic of SELECT statement blocks in a query statement. The columns passed to the next operator, separated by commas, are displayed in the EXPLAIN results. If the result is a reference to a column, it is displayed as < alias >.< column_name >. If the result is an expression, it is displayed as a function, for example, func1(arg1_1, arg1_2, func2(arg2_1, arg2_2)). If the result is a constant, the value is displayed directly.
FilterOperator	Describes the logic of WHERE statement blocks in a query statement. A WHERE condition, which complies with a display rule similar to that of selectOperator, is displayed in the EXPLAIN results.
JoinOperator	Describes the logic of JOIN statement blocks in a query statement. The tables involved in the JOIN operation and the mode of JOIN operation are displayed in the EXPLAIN results.
GroupByOperator	Describes the logic of the AGGREGATE operation . This structure is displayed if an aggregate function is used in a query. The content of the aggregate function is displayed in the EXPLAIN results.
ReduceSinkOperator	Describes the logic of the data distribution operation between tasks. If the result of the current task is transferred to another task, ReduceSinkOperator must be used to distribute data at the end of the current task. The output sorting method, the distributed keys, values, and columns used to calculate the hash value are displayed in the EXPLAIN results.

Operator	Description
FileSinkOperator	Describes the final data storage operation. If there is an INSERT statement block in the query statement, the name of the target table is displayed in the EXPLAIN results.
LimitOperator	Describes the logic of LIMIT statement blocks in a query statement. The limit value is displayed in the EXPLAIN results.
MapjoinOperator	Describes JOIN operations in large tables, similar to JoinOperator.

**Note:**

- If a query is complex and has too many EXPLAIN results, the API restriction is triggered, and incomplete results are displayed. In this case, the query can be split, and the EXPLAIN operation can be performed on each part to show the structure of the job.
- The maximum number of partitions in a query is 10,000. Inputting too many partitions leads to over-length Data source content. To circumvent this limit, you can filter out most partitions by adding a query filter.

1.5.4.6 GROUPING SETS

1.5.4.6.1 Overview

For scenarios where you need to aggregate and analyze data of multiple dimensions, you must execute multiple UNION ALL clauses. For example, you wanted to aggregate column a, aggregate column b, and aggregate columns a and b together. The GROUPING SETS clause is a better choice in such cases.

GROUPING SETS is an extension to the GROUP BY clause in the SELECT statement. You can group results in various ways by using GROUPING SETS without executing multiple SELECT statements. This can produce better execution plans and result in higher performance from the MaxCompute engine.

**Notice:**

Many examples in this topic are demonstrated using MaxCompute Studio. We recommend that you install MaxCompute Studio before you proceed with subsequent operations.

1.5.4.6.2 Example

The following example is for your reference.

1. Prepare data.

```
create table requests LIFECYCLE 20 as
select * from values
  (1, 'windows', 'PC', 'Beijing'),
  (2, 'windows', 'PC', 'Shijiazhuang'),
  (3, 'linux', 'Phone', 'Beijing'),
  (4, 'windows', 'PC', 'Beijing'),
  (5, 'ios', 'Phone', 'Shijiazhuang'),
  (6, 'linux', 'PC', 'Beijing'),
  (7, 'windows', 'Phone', 'Shijiazhuang')
as t(id, os, device, city);
```

2. Use GROUPING SETS.

```
SELECT os,device, city ,COUNT(*)
FROM requests
GROUP BY os, device, city GROUPING SETS((os, device), (city), ());
```

A similar output is displayed.

Figure 1-3: Command output

	os	device	city	_c3
1	ios	Phone	\N	1
2	linux	PC	\N	1
3	linux	Phone	\N	1
4	windows	PC	\N	3
5	windows	Phone	\N	1
6	\N	\N	Beijing	4
7	\N	\N	Shijiazhuang	3
8	\N	\N	\N	7



Note:

You can also execute multiple SELECT statements to obtain the same result.

```
SELECT NULL, NULL, NULL, COUNT(*)
FROM requests
UNION ALL
SELECT os, device, NULL, COUNT(*)
FROM requests GROUP BY os, device
UNION ALL
SELECT null, null, city, COUNT(*)
FROM requests GROUP BY city;
```

However, the GROUPING SETS method is simpler and more efficient.



Notice:

Expressions not used in GROUPING SETS use NULL as placeholders. You can execute UNION statements on grouping sets.

1.5.4.6.3 CUBE and ROLLUP

CUBE and ROLLUP are special GROUPING SETS functions. CUBE lists all possible combinations of specified columns as grouping sets. ROLLUP aggregates data by level to produce grouping sets.

Example:

```
GROUP BY CUBE(a, b, c)
```

```
GROUPING SETS((a,b,c),(a,b),(a,c),(b,c),(a),(b),(c),())
```

The preceding two clauses are equivalent.

```
GROUP BY ROLLUP(a, b, c)
```

```
GROUPING SETS((a,b,c),(a,b),(a))
```

The preceding two clauses are equivalent.

1.5.4.6.4 GROUPING and GROUPING_ID

NULL is used as placeholders in grouping sets, but it can also be a value that is manually entered. In the code, however, placeholder NULLs are indistinguishable from value NULLs. The GROUPING function is provided to address this issue.

GROUPING allows you to specify the name of a column as a parameter. If the specified lines are aggregated based on a column whose name is used as a parameter in this function, 0 is returned, indicating that NULL is an entered value. Otherwise, 1 is returned, indicating that NULL is a placeholder.

GROUPING_ID can be used to specify the names of one or more columns as parameters. The GROUPING results in these columns are formed into integers by using BitMap.

Example:

```
SELECT a,b,c ,COUNT(*),  
GROUPING(a) ga, GROUPING(b) gb, GROUPING(c) gc, GROUPING_ID(a,b,c)  
groupingid  
FROM VALUES (1,2,3) as t(a,b,c)
```

```
GROUP BY CUBE(a,b,c);
```

A similar output is displayed.

Figure 1-4: Command output

	a	b	c	_c3	ga	gb	gc	groupingid
1	\N	\N	\N	1	1	1	1	7
2	\N	\N	3	1	1	1	0	6
3	\N	2	\N	1	1	0	1	5
4	\N	2	3	1	1	0	0	4
5	1	\N	\N	1	0	1	1	3
6	1	\N	3	1	0	1	0	2
7	1	2	\N	1	0	0	1	1
8	1	2	3	1	0	0	0	0

1.5.5 SELECT TRANSFORM

1.5.5.1 Overview

SELECT TRANSFORM implements features that MaxCompute SQL does not provide . **SELECT TRANSFORM** allows you to start a specified child process and enter data of a required format into the child process through standard input (stdin). Then, you can parse the standard output (stdout) of the child process to obtain the final output. This process does not require you to compile UDFs.

SELECT TRANSFORM simplifies the reference of script code and supports programming languages such as Java, Python, Shell, and Perl. It is suitable for ad hoc data analysis. MaxCompute Select Transform is fully compatible with Hive syntax, features, and actions, including input/output row format and reader/writer . Most Hive scripts can be added directly to the **SELECT TRANSFORM** statement. Others can be used after a few changes.

Command syntax:

```
SELECT TRANSFORM(arg1, arg2 ...)
(ROW FORMAT DELIMITED (FIELDS TERMINATED BY field_delimiter (ESCAPED
  BY character_escape)?)? (LINES SEPARATED BY line_separator)? (NULL
  DEFINED AS null_value)?)?
USING 'unix_command_line'
(RESOURCES 'res_name' (' 'res_name') *)?
( AS col1, col2 ...)?
(ROW FORMAT DELIMITED (FIELDS TERMINATED BY field_delimiter (ESCAPED
  BY character_escape)?)? (LINES SEPARATED BY line_separator)? (NULL
  DEFINED AS null_value)?)?
```

Description:

- **SELECT TRANSFORM:** The **SELECT TRANSFORM** keyword can be replaced with the **MAP** or **REDUCE** keyword while maintaining the same semantic meaning.

However, we recommend that you use **SELECT TRANSFORM** because its syntax is simpler.

- **(arg1, arg2 ...):** arguments in the **TRANSFORM** clause. Their format is similar to those of items in the **SELECT** clause. In the default format, the results of expressions for each argument are combined by using `\t` after they are implicitly converted into strings. The arguments are then entered into the specified child process.



Note:

The default format is configurable. For more information, see **ROW FORMAT**.

- **USING:** specifies the command used to start a child process. Note the following points about the **USING** clause.
 - In most MaxCompute SQL statements, the **USING** clause can only specify resources. However, in the **SELECT TRANSFORM** statement, the **USING** clause can specify commands to ensure compatibility with Hive syntax.
 - The format of the **USING** clause is similar to the syntax of a Shell script. However, a Shell script is not actually expected to start the child process. The child process is created based on the command input. Because of this, a number of Shell functions, such as input and output redirection, pipe, and loop, are unavailable. A Shell script can be used as to start a child process if necessary.
- **RESOURCES:** specifies the resources that the specified child process can access. You can use one of the following methods to specify resources:
 - Use the **RESOURCES** clause. Example: `using 'sh foo.sh bar.txt' Resources 'foo.sh', 'bar.txt'.`
 - Add the `set odps.sql.session.resources=foo.sh,bar.txt;` clause before SQL statements.



Notice:

This clause takes effect globally once it is specified. All **SELECT TRANSFORM** statements will be able to access the resources specified by this clause.

- **ROW FORMAT:** specifies the input or output format. Two **ROW FORMAT** clauses are used in the syntax: the first one specifies the input format, and the second

one specifies the output format. \t is used to separate columns, \n is used to separate rows, and NULL is represented by \N .



Notice:

- For field_delimiter, character_escape, and line_separator, only one character can be accepted. If you specify a string, the first character in the string takes priority over the others.
- There are a variety of Hive syntaxes to specify formats. MaxCompute supports syntaxes such as inputRecordReader, outputRecordReader, and Serdeinput. To use these formats, you must enable Hive compatibility by adding the `set odps.sql.hive.compatible=true;` clause before SQL statements. If you specify a syntax such as inputRecordReader or outputRecordReader supported by Hive, statements may be executed at lower speeds.

- AS: specifies output columns.



Note:

- You can specify data types in the AS clause, as in `as(col1:bigint, col2:boolean)`. By default, strings are returned if you do not specify data types, as in `as(col1, col2)`.
- The output is obtained by parsing the stdout of the child process. If the specified data types do not include STRING, the system implicitly calls the CAST function. Runtime exceptions may occur when the CAST function is called.
- You cannot specify data types for only some of the columns, as in `as(col1, col2:bigint)`.
- If you skip the AS clause, the field preceding the first \t in the stdout is a key, and all the following parts are a value. This is equivalent to `as(key, value)`.

1.5.5.2 SELECT TRANSFORM examples

1.5.5.2.1 Call Shell scripts

In this example, a Shell script is used to generate 50 lines of data starting from 1 to 50. The output of the data field is as follows:

```
SELECT TRANSFORM(script) USING 'sh' AS (data)
FROM (
    SELECT 'for i in `seq 1 50`; do echo $i; done' AS script
```

```
) t
;
```

The Shell commands are used as the input of the TRANSFORM clause.



Note:

In addition to language extensions, SELECT TRANSFORM also provides simple features of AWK, Python, Perl, and Shell to compile scripts in commands. You do not need to compile script files or upload resources separately.

You can upload script files for complex cases, as in the following example Python script call.

1.5.5.2.2 Call Python scripts

This topic provides an example of how to use SELECT TRANSFORM to call Python scripts.

1. Compile a Python script file. In this example, the file name is myplus.py.

```
#!/usr/bin/env python
import sys
line = sys.stdin.readline()
while line:
    token = line.split('\t')
    if (token[0] == '\\N') or (token[1] == '\\N'):
        print '\\N'
    else:
        print int(token[0]) + int(token[1])
    line = sys.stdin.readline()
```

2. Add the Python script file as a resource to MaxCompute.

```
add py ./myplus.py -f;
```



Note:

You can also add resources from the DataWorks console.

3. Execute the SELECT TRANSFORM statement to call the resource.

```
Create table testdata(c1 bigint,c2 bigint); -- Create a test table.
insert into Table testdata values (1,4),(2,5),(3,6); -- Insert test
data into the test table.
-- Execute the SELECT TRANSFORM statement:
SELECT
TRANSFORM (testdata.c1, testdata.c2)
USING 'python myplus.py' resources 'myplus.py'
AS (result bigint)
FROM testdata;
-- Or
set odps.sql.session.resources=myplus.py;
SELECT
```

```
TRANSFORM (testdata.c1, testdata.c2)
USING 'python myplus.py'
AS (result bigint)
FROM testdata;
```

4. A similar output is displayed:

```
+-----+
| cnt |
+-----+
| 5   |
| 7   |
| 9   |
+-----+
```

Python scripts are not subject to any format requirements and do not require a Python framework to be run in MaxCompute. In MaxCompute, Python commands can be used as the input of the TRANSFORM clause. For example, you can call Shell scripts by running Python commands.

```
SELECT TRANSFORM('for i in xrange(1, 50): print i;') USING 'python'
AS (data);
```

1.5.5.2.3 Call Java scripts

Java scripts are called in a similar manner to Python scripts. In this example, you need to compile a Java script file, export it as a JAR package, and then run the add command to add the JAR package as a resource to MaxCompute. The resource will be called by using SELECT TRANSFORM.

1. Compile a Java script file and export it as a JAR package. In this example, the name of the JAR package is Sum.jar.

```
package com.aliyun.odps.test;
import java.util.Scanner;
public class Sum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (sc.hasNext()) {
            String s = sc.nextLine();
            String[] tokens = s.split("\t");
            if (tokens.length < 2) {
                throw new RuntimeException("illegal input");
            }
            if (tokens[0].equals("\\N") || tokens[1].equals("\\N")) {
                System.out.println("\\N");
            }
            System.out.println(Long.parseLong(tokens[0]) + Long.parseLong(
tokens[1]));
        }
    }
}
```

```
}
```

2. Add the JAR package as a resource to MaxCompute.

```
add jar . /Sum.jar -f;
```

3. Execute the SELECT TRANSFORM statement to call the resource.

```
Create table testdata(c1 bigint,c2 bigint); -- Create a test table.
insert into Table testdata values (1,4),(2,5),(3,6); -- Insert test
data into the test table.
-- Execute the SELECT TRANSFORM statement:
SELECT TRANSFORM(testdata.c1, testdata.c2)
    USING 'java -cp Sum.jar com.aliyun.odps.test.Sum' resources 'Sum.
jar'
from testdata;
-- Or
set odps.sql.session.resources=Sum.jar;
SELECT TRANSFORM(testdata.c1, testdata.c2)
    USING 'java -cp Sum.jar com.aliyun.odps.test.Sum'
FROM testdata;
```

4. A similar output is displayed:

```
+-----+
|  cnt  |
+-----+
|   5   |
|   7   |
|   9   |
+-----+
```

You can use the preceding method to run most Java utilities.

Although UDTF frameworks are provided for Java and Python, it is easier to compile code by using SELECT TRANSFORM. SELECT TRANSFORM is a simpler process because it is not subject to any format requirements and can be called offline. The paths for Java and Python offline scripts can be obtained from the JAVA_HOME and PYTHON_HOME environment variables.

1.5.5.2.4 Call scripts of other languages

In addition to language extensions, SELECT TRANSFORM also supports commonly used Unix command and script interpreters, such as AWK and Perl.

An example of calling AWK:

```
SELECT TRANSFORM(*) USING "awk '{print $2}'" as (data) from testdata
;
```

An example of calling Perl:

```
SELECT TRANSFORM (testdata.c1, testdata.c2) USING "perl -e 'while($
input = <STDIN>){print $input;}'" FROM testdata;
```



Notice:

PHP and Ruby are not deployed in the MaxCompute cluster and cannot be called.

1.5.5.2.5 Call scripts in series

SELECT TRANSFORM allows you to call scripts in series. For example, you can use **DISTRIBUTE BY** and **SORT BY** to pre-process data.

```
SELECT TRANSFORM(key, value) USING 'cmd2' from
(
  SELECT TRANSFORM(*) USING 'cmd1' from
  (
    SELECT * FROM data distribute by col2 sort by col1
  ) t distribute by key sort by value
) t2;
```

More often, you can use either the map or reduce keywords to produce the same results.

```
@a := select * from data distribute by col2 sort by col1;
@b := map * using 'cmd1' distribute by col1 sort by col2 from @a;
reduce * using 'cmd2' from @b;
```

1.5.5.3 Performance advantages

The performance of SELECT TRANSFORM and UDTF varies depending on the specific scenario. In general, SELECT TRANSFORM performs better. However, UDTF performs better as the volume of data increases. Because the development of transform is easier, SELECT TRANSFORM is more suitable for ad hoc data analysis.

The advantages of UDTFs and SELECT TRANSFORM are listed in the following sections.

Advantages of UDTFs

- **Output and input follow specified data types and do not require conversion.**
- **Processes are not suspended if the operating system pipe is empty or fully occupied. The operating system pipe has a 4 KB buffer.**

- **Constant parameters do not need to be transmitted.**

Advantages of SELECT TRANSFORM

- **Supports child and parent processes and can utilize multiple server cores when high CPU usage and low throughput is needed.**
- **Calls underlying systems to read and write data to be transmitted, giving it a higher performance than Java.**
- **Supports tools such as AWK and can run native code.**

1.5.6 UNION, INTERSECT, and EXCEPT

This topic describes SQL syntax, descriptions and examples of UNOIN ALL, UNION DISTINCT, INTERSECT ALL, INTERSECT DISTINCT, EXCEPT ALL, and EXCEPT DISTINCT.

Syntax:

```
select_statement UNION ALL select_statement;  
select_statement UNION [DISTINCT] select_statement;  
select_statement INTERSECT ALL select_statement;  
select_statement INTERSECT [DISTINCT] select_statement;  
select_statement EXCEPT ALL select_statement;  
select_statement EXCEPT [DISTINCT] select_statement;  
select_statement MINUS ALL select_statement;  
select_statement MINUS [DISTINCT] select_statement;
```

Purpose: It is used to return the union of two data sets, the intersection of two data sets, or the complement of the second dataset in the first dataset.

Description:

- **UNION:** returns the union of two datasets. It combines the two datasets into one dataset.
- **INTERSECT:** returns the intersection of two datasets. It outputs the records contained in both datasets.
- **EXCEPT:** returns the complement of the second dataset in the first dataset. It outputs the records that are contained in the first dataset, but not in the second dataset.
- **MINUS:** equivalent to EXCEPT.

Examples:

- **UNOIN ALL example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4) t(a, b)  
UNION ALL
```

```
SELECT * FROM VALUES (1, 2), (1, 4) t(a, b);
```

Returned result: two datasets are combined.

a	b
1	2
1	4
1	2
1	2
3	4

• **UNION DISTINCT example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4) t(a, b)
UNION
SELECT * FROM VALUES (1, 2), (1, 4) t(a, b);
```

Returned result: equivalent to SELECT DISTINCT * FROM (< the result of UNION ALL >) t;.

a	b
1	2
1	4
3	4

• **INTERSECT ALL example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (5, 6) t(a, b)
INTERSECT ALL
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (5, 7) t(a, b);
```

Returned result: deduplication is skipped in INTERSECT ALL. It seems that there is a hidden serial number behind the same row and each row can be displayed separately.

a	b
1	2
1	2
3	4

• **INTERSECT DISTINCT example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (5, 6) t(a, b)
INTERSECT
```

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (5, 7) t(a, b);
```

Returned result: `SELECT DISTINCT * FROM (< the result of INTERSECT ALL >) t;`

a	b
1	2
3	4

• **EXCEPT ALL example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (3, 4), (5, 6), (7, 8)
t(a, b)
EXCEPT ALL
SELECT * FROM VALUES (3, 4), (5, 6), (5, 6), (9, 10) t(a, b);
```

Returned result: deduplication is skipped in EXCEPT ALL. There is a hidden serial number behind the same row and each row can be displayed separately.

a	b
1	2
1	2
3	4
7	8

• **EXCEPT DISTINCT example:**

```
SELECT * FROM VALUES (1, 2), (1, 2), (3, 4), (3, 4), (5, 6), (7, 8)
t(a, b)
EXCEPT
SELECT * FROM VALUES (3, 4), (5, 6), (5, 6), (9, 10) t(a, b);
```

Returned result: equivalent to `Select distinct * FROM left_branch limit t all select distinct * FROM right_branch;`

a	b
1	2
7	8



Note:

- Sorting may be skipped in the preceding operations.
- The left and right branches in the preceding operations must have the same number of columns. In addition, if data types in the left and right branches are

not consistent, they may be implicitly converted. Due to compatibility issues, implicit conversion is not carried out between STRING and no-STRING types for the preceding operations.

- Up to 256 branches are allowed in the preceding operations. An error is returned if more branches are used.
- If the UNION statement is followed by the CLUSTER BY, DISTRIBUTE BY, SORT BY, ORDER BY or LIMIT clause and you add `set odps.sql.type.system.odps2=false;`, the SET statement is applicable to the last `select_statement`; of the UNION statement. If you add `set odps.sql.type.system.odps2=true;`, the SET statement is applicable to all `select_statements` of the UNION statement.

Example:

```
set odps.sql.type.system.odps2=true;
SELECT explode(array(3, 1)) AS (a) UNION ALL SELECT explode(array(0
, 4, 2)) AS (a) ORDER
BY a LIMIT 3;
```

Returned result:

```
+-----+
|  a    |
+-----+
|  0    |
|  1    |
|  2    |
+-----+
```

1.5.7 Built-in functions

1.5.7.1 Mathematical functions

1.5.7.1.1 ABS

This topic describes the ABS function.

Function declaration:

```
double abs(double number)
bigint abs(bigint number)
decimal abs(decimal number)
```

Purpose: It is used to return absolute values.

Description:

number: double, bigint or decimal type. When the input is of the bigint type, a value of the bigint type is returned; when the input is of the double type, a value

of the double type is returned. If the input is of the string type, it is implicitly converted into a value of the double type before this computation. If the input is of another type, an error is returned.

Returned value: double, bigint, or decimal type, depending on the type of the input. If the input is NULL, NULL is returned.



Note:

When the input is of the bigint type and is out of the maximum range of the bigint type, the returned value is of the double type. In this case, the precision may be diminished.

Example:

```
abs(null) = null
abs(-1) = 1
abs(-1.2) = 1.2
abs("-2") = 2.0
abs(122320837456298376592387456923748) = 1.2232083745629837e32
```

The following example shows the usage of a complete ABS function in SQL. Other built-in functions (except window functions and aggregation functions) are in similar usage to this function and are not shown here.

```
select abs(id) from tbl1;
-- Take the absolute value of the id field in tbl1.
```

1.5.7.1.2 ACOS

Function declaration:

```
double acos(double number)
decimal acos(decimal number)
```

Purpose: It is used to calculate the arccosine of a number.

Description:

number: double or decimal type. Value range: -1 to 1. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. Value range: 0 to π . If number is NULL, NULL is returned.

Example:

```
acos("0.87") = 0.5155940062460905  
acos(0) = 1.5707963267948966
```

1.5.7.1.3 ASIN

Function declaration:

```
double asin(double number)  
decimal asin(DECIMAL number)
```

Purpose: It is used to calculate the arcsine of a number.

Description:

number: double or decimal type. Value range: -1 to 1. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. Value range: $-\pi/2$ to $\pi/2$. If number is NULL, NULL is returned.

Example:

```
asin(1) = 1.5707963267948966  
asin(-1) = -1.5707963267948966
```

1.5.7.1.4 ATAN

Function declaration:

```
double atan(double number)
```

Purpose: It is used to calculate the arctangent of a number.

Description:

number: double type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double type. Value range: $-\pi/2$ to $\pi/2$. If number is NULL, NULL is returned.

Example:

```
atan(1) = 0.7853981633974483;
```

```
atan(-1) = -0.7853981633974483
```

1.5.7.1.5 CEIL

Command syntax:

```
bigint ceil(double value)  
bigint ceil(decimal value)
```

Purpose: It is used to return the smallest integer that is equal to or greater than the input value.

Description:

value: double or decimal. If the value is of the string or bigint type, it is implicitly converted to the double type. For all other input types, an error is returned.

Returned value: bigint type. If the input is NULL, NULL is returned.

Example:

```
ceil(1.1) = 2  
ceil(-1.1) = -1
```

1.5.7.1.6 CONV

Command syntax:

```
string conv(string input, bigint from_base, bigint to_base)
```

Purpose: It is used to convert a number from one numeric base number system to another.

Description:

- **input:** an integer of the string type to be converted. It accepts values of the bigint and double types by means of implicit conversion.
- **from_base, to_base:** a number system value in decimal form. Value range: 2, 8, 10, and 16. It accepts values of the string and double types by means of implicit conversion.

Returned value: string type. If any input is NULL, NULL is returned. The conversion process runs at a 64-bit precision. An error is returned when overflow occurs. If the input is a negative value (beginning with '-'), an error is returned. If the input is a decimal, it is converted to an integer before hex conversion. The decimal part is left out.

Example:

```
conv('1100', 2, 10) = '12'  
conv('1100', 2, 16) = 'c'  
conv('ab', 16, 10) = '171'  
conv('ab', 16, 16) = 'ab'
```

1.5.7.1.7 COS

Command syntax:

```
double cos(double number)  
decimal cos(decimal number)
```

Purpose: It is used to return the cosine of a number. The input must be a radian value.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted to a value of the double type. For all other input types, an error is returned.

Returned value: double or decimal type. If the input is NULL, NULL is returned.

Example:

```
cos(3.1415926/2) = 2.6794896585028633e-8  
cos(3.1415926) = -0.99999999999999986
```

1.5.7.1.8 COSH

Command syntax:

```
double cosh(double number)  
decimal cosh(decimal number)
```

Purpose: It is used to return the hyperbolic cosine of a number.

Description:

number: double or decimal. If the input is of the string or bigint type, it is implicitly converted to a value of the double type. For all other input types, an error is returned.

Returned value: double or decimal. If the input is NULL, NULL is returned.

1.5.7.1.9 COT

Function declaration:

```
double cot(double number)
decimal cot(decimal number)
```

Purpose: It is used to return the cotangent of a number. The input must be a radian value.

Description:

number: double or decimal. If the input is of the string or bigint type, it is implicitly converted a value of the double type. For all other input types, an error is returned.

Returned value: double or decimal type. If the input is NULL, NULL is returned.

1.5.7.1.10 EXP

Function declaration:

```
double exp(double number)
decimal exp(decimal number)
```

Purpose: It is used to return the exponent value of number.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.11 FLOOR

Function declaration:

```
bigint floor(double number)
bigint floor(decimal number)
```

Purpose: It is used to return the round-down integer that is less than or equal to number.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: bigint type. If number is NULL, NULL is returned.

Example:

```
floor(1.2) = 1  
floor(1.9) = 1  
floor(0.1) = 0  
floor(-1.2) = -2  
floor(-0.1) = -1  
floor(0.0) = 0  
floor(-0.0) = 0
```

1.5.7.1.12 LN

Function declaration:

```
double ln(double number)  
decimal ln(decimal number)
```

Purpose: It is used to return the natural logarithm of a number.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. If the input is NULL, negative, or zero, NULL is returned.

1.5.7.1.13 LOG

Function declaration:

```
double log(double base, double x)  
decimal log(decimal base, DECIMAL x)
```

Purpose: It is used to return the logarithm of x to base.

Description:

- **base:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.
- **x:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: logarithm value of the double or decimal type. If either base or x is NULL, negative, or zero, NULL is returned. If base is 1 (which leads to division by zero), NULL is returned.

1.5.7.1.14 POW

Command syntax:

```
double pow(double x, double y)
decimal pow(decimal x, decimal y)
```

Purpose: It is used to return the yth power of x, that is, x^y .

Description:

- **x:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.
- **y:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. If x or y is NULL, NULL is returned.

1.5.7.1.15 RAND

Command syntax:

```
double rand(bigint seed)
```

Purpose: It is used to return a random number of the double type from 0 to 1 based on the seed.

Description:

Seed: optional, bigint type. It is the seed of a random number, and determines the start value of the random number sequence.

Returned value: double type.

Example:

```
select rand() from dual;
```

```
select rand(1) from dual;
```

1.5.7.1.16 ROUND

Function declaration:

```
double round(double number, [bigint decimal_places])  
decimal round(decimal number, [bigint decimal_places])
```

Purpose: It is used to return a number rounded to the specified decimal place.

Description:

- **number:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. If the input is of another type, an error is returned.
- **decimal_place:** a constant of the bigint type. It indicates the specified decimal place to which the result is to be rounded off. For all other input types, an error is returned. If it is omitted, the number is rounded to the ones place. The default value is 0.

Returned value: double or decimal type. If number or decimal_places is NULL, NULL is returned.



Note:

decimal_places can be negative. Negative numbers are counted from the decimal point to left and the decimal part is left out; if the value of decimal_places is greater than the length of the integer part, 0 is returned.

Example:

```
round(125.315) = 125.0  
round(125.315, 0) = 125.0  
Round (125.315, 1) = 125.3  
round(125.315, 2) = 125.32  
round(125.315, 3) = 125.315  
round(-125.315, 2) = -125.32  
round(123.345, -2) = 100.0  
round(null) = null  
round(123.345, 4) = 123.345  
round(123.345, -4) = 0.0
```

1.5.7.1.17 SIN

Function declaration:

```
double sin(double number)
```

```
decimal sin(decimal number)
```

Purpose: It is used to return the sine of a number. The input must be a radian value.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.18 SINH

Function declaration:

```
double sinh(double number)  
decimal sinh(decimal number)
```

Purpose: It is used to return the hyperbolic sine of a number.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.19 SQRT

Function declaration:

```
double sqrt(double number)  
decimal sqrt(decimal number)
```

Purpose: It is used to return the square root of a number.

Description:

number: double or decimal type. It must be greater than 0. If it is less than 0, an error is returned. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.20 TAN

Function declaration:

```
double tan(double number)
decimal tan(decimal number)
```

Purpose: It is used to return the tangent of a number. The input must be a radian value.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.21 TANH

Function declaration:

```
double tanh(double number)
decimal tanh(decimal number)
```

Purpose: It is used to return the hyperbolic tangent of a number.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned.

Returned value: double or decimal type. If number is NULL, NULL is returned.

1.5.7.1.22 TRUNC

Function declaration:

```
double trunc(double number[, bigint decimal_places])
decimal trunc(decimal number[, bigint decimal_places])
```

Purpose: It is used to truncate 'number' to the specified decimal place.

Description:

- **number:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned.

- **decimal_places**: a constant of the bigint type. It indicates the decimal place to which a number is to be truncated. Numbers of other types are implicitly converted into values of the bigint type. If it is omitted, the result is truncated to the ones place by default.

Returned value: double or decimal type. If number or decimal_places is NULL, NULL is returned.



Note:

- The truncated part is supplemented with 0.
- **decimal_places** can be negative. Negative numbers are truncated from the decimal point to the left and the decimal part is left out. If the value of decimal_places is greater than the length of the integer part, 0 is returned.

Example:

```
trunc(125.815) = 125.0
trunc(125.815, 0) = 125.0
trunc(125.815, 1) = 125.800000000000001
trunc(125.815, 2) = 125.81
trunc(125.815, 3) = 125.815
trunc(-125.815, 2) = -125.81
trunc(125.815, -1) = 120.0
trunc(125.815, -2) = 100.0
trunc(125.815, -3) = 0.0
trunc(123.345, 4) = 123.345
trunc(123.345, -4) = 0.0
```

1.5.7.1.23 Additional mathematical functions

MaxCompute 2.0 provides additional mathematical functions. You must add the following SET statement before SQL statements contained in the UNHEX function:

```
set odps.sql.type.system.odps2=true;
```



Note:

You must submit and execute the SET statement and the SQL statements of the new functions simultaneously.

The mathematical functions described in subsequent topics are new in MaxCompute 2.0.

1.5.7.1.24 LOG2

Function declaration:

```
Double log2(DOUBLE number)  
Double log2(DECIMAL number)
```

Purpose: It is used to return the logarithm of number to base 2.

Description:

number: double or decimal type.

Returned value: double type. If the input is 0 or NULL, NULL is returned.

Example:

```
log2(null) = null  
log2(0) = null  
log2(8) = 3.0
```

1.5.7.1.25 LOG10

Function declaration:

```
Double log10(Double number)  
Double log10(Decimal number)
```

Purpose: It is used to return the logarithm of number to base 10.

Description:

number: double or decimal type.

Returned value: double type. If the input is 0 or NULL, NULL is returned.

Example:

```
log10(null) = null  
log10(0) = null  
log10(8) = 0.9030899869919435  
log10('abc') = null
```

1.5.7.1.26 BIN

Command syntax:

```
string bin(bigint number)
```

Purpose: It is used to return the binary format of a number.

Description:

number: bigint.

Returned value: string type. If the input is 0, 0 is returned. If the input is NULL, NULL is returned.

Example:

```
bin(0) = '0'  
bin(null) = 'null'  
bin(12) = '1100'
```

1.5.7.1.27 HEX

Function declaration:

```
STRING hex(BIGINT number)  
STRING hex(STRING number)  
STRING hex(BINARY number)
```

Purpose: It is used to convert an integer or character into hexadecimal format.

Description:

number: If this value is of the bigint type, the hexadecimal format of the number is returned. If this value is of the string type, the hexadecimal value of the string is returned.

Returned value: string type. If the input is 0, 0 is returned. If the input is NULL, NULL is returned.

Example:

```
hex(0) = '0'  
hex('abc') = '616263'  
hex(17) = '11'  
hex('17') = '3137'  
hex(null) = 'null'
```

1.5.7.1.28 UNHEX

Function declaration:

```
BINARY unhex(STRING number)
```

Purpose: It is used to return the regular character string represented in the hexadecimal format.

Description:

number: a hexadecimal string.

Returned value: binary type. If the input is 0, a failure is returned. If the input is NULL, NULL is returned.

Example:

```
unhex('616263') = 'abc'  
unhex(616263) = 'abc'
```

1.5.7.1.29 RADIANS

Command syntax:

```
double radians(double number)
```

Purpose: It is used to convert degrees into radians.

Description:

number: double type

Returned value: double type. If the input is NULL, NULL is returned.

Example:

```
radians(90) = 1.5707963267948966  
radians(0) = 0.0  
radians(null) = null
```

1.5.7.1.30 DEGREES

Function declaration:

```
DOUBLE degrees(DOUBLE number)  
DOUBLE degrees(DECIMAL number)
```

Purpose: It is used to convert radians into degrees.

Description:

number: double or decimal type.

Returned value: double type. If the input is NULL, NULL is returned.

Example:

```
degrees(1.5707963267948966) = 90.0  
degrees(0) = 0.0
```

```
degrees(null) = null
```

1.5.7.1.31 SIGN

Function declaration:

```
DOUBLE sign(DOUBLE number)  
DOUBLE sign(DECIMAL number)
```

Purpose: It is used to indicate the sign of the input data. 1.0 indicates positive and -1.0 indicates negative. 0.0 indicates 0.

Description:

number: double or decimal type.

Returned value: double type. If the input is 0, 0.0 is returned. If the input is NULL, NULL is returned.

Example:

```
sign(-2.5) = -1.0  
sign(2.5) = 1.0  
sign(0) = 0.0  
sign(null) = null
```

1.5.7.1.32 E

Function declaration:

```
DOUBLE e()
```

Purpose: It is used to return the value of e (Euler's number).

Returned value: double type.

Example:

```
e() = 2.718281828459045
```

1.5.7.1.33 PI

Function declaration:

```
DOUBLE pi()
```

Purpose: It is used to return the value of π .

Returned value: double type.

Example:

```
pi() = 3.141592653589793
```

1.5.7.1.34 FACTORIAL

Function declaration:

```
BIGINT factorial(INT number)
```

Purpose: It is used to return the factorial of number.

Description:

number: int type. Value range: 0 to 20.

Returned value: bigint type. If the input is 0, 1 is returned. If the input is NULL or any value outside the range of 0 to 20, NULL is returned.

Example:

```
factorial(5) = 120 --5! = 5*4*3*2*1 = 120
```

1.5.7.1.35 CBRT

Command syntax:

```
double cbrt(double number)
```

Purpose: It is used to return the cube root of a number.

Description:

number: double type.

Returned value: double type. If the input is NULL, NULL is returned.

Example:

```
cbrt(8) = 2  
cbrt(null) = null
```

1.5.7.1.36 SHIFTLEFT

Function declaration:

```
INT shiftleft(TINYINT|SMALLINT|INT number1, INT number2)  
BIGINT shiftleft(BIGINT number1, INT number2)
```

Purpose: It is used to shift left a value by a given number of places (<<).

Description:

- **number1:** an integer of the tinyint, smallint, int, or bigint type.
- **number2:** an integer of the int type.

Returned value: int or bigint type.

Example:

```
shiftright(1,2) = 4
-- Shift left the binary value of 1 by two places (1<<2, 0001 changed
to 0100)
shiftright(4,3) = 32
-- Shift left the binary value of 4 by three places (4<<3, 0100
changed to 100000)
```

1.5.7.1.37 SHIFTRIGHT

Function declaration:

```
INT shiftright(TINYINT|SMALLINT|INT number1, INT number2)
BIGINT shiftright(BIGINT number1, INT number2)
```

Purpose: It is used to shift right a value by a given number of places (>>).

Description:

- **number1:** an integer of the tinyint, smallint, int, or bigint type.
- **number2:** an integer of the int type.

Returned value: int or bigint type.

Example:

```
shiftright(4,2) = 1
-- Shift right the unsigned binary value of 4 by two places (4>>2,
0100 changed to 0001)
shiftright(32,3) = 4
-- Shift right the unsigned binary value of 32 by two places (32>>3,
100000 changed to 0100)
```

1.5.7.1.38 SHIFTRIGHTUNSIGNED

Function declaration:

```
INT shiftrightunsigned(TINYINT|SMALLINT|INT number1, INT number2)
BIGINT shiftrightunsigned(BIGINT number1, INT number2)
```

Purpose: It is used to shift right an unsigned value by a given number of places (>>>).

Description:

- **number1**: an integer of the tinyint, smallint, int, or bigint type.
- **number2**: an integer of the int type.

Returned value: int or bigint type.

Example:

```
shiftrightunsigned(8,2) = 2
-- In this example, shift right the unsigned binary value of 8 (1000
in binary) by two places and return 2 (0010 in binary).
shiftrightunsigned(-14,2) = 1073741820
-- Shift right the unsigned binary value of -14 by two places (-14>>>
2, 11111111 11111111 11111111 11110010 changed to 00111111 11111111
11111111 11111100)
```

1.5.7.2 String processing functions

1.5.7.2.1 CHAR_MATCHCOUNT

Command syntax:

```
bigint char_matchcount(string str1, string str2)
```

Purpose: It is used to return the number of characters in str1 that appear in str2 (repeated characters are not counted).

Description:

str1 and str2: string type. Both must be valid UTF-8 strings. If invalid characters are found during matching, a negative value is returned.

Returned value: bigint type. If any input is NULL, NULL is returned.

Example:

```
char_matchcount('abd', 'aabc') = 2
-- The a and b characters in str1 appear in str2.
```

1.5.7.2.2 CHR

Command syntax:

```
string chr(bigint ascii)
```

Purpose: It is used to convert an ASCII code into the corresponding character.

Description:

ascii: ASCII value of the bigint type. If the input is of the string, double, or decimal type, it is implicitly converted into a value of the bigint type before this computation. If the input is of another type, an error is returned.

Returned value: string type. The parameter value range is from 0 to 255. A value out of range will cause an error. If the input is NULL, NULL is returned.

1.5.7.2.3 CONCAT

Command syntax:

```
string concat(string a, string b...)
```

Purpose: It is used to join input strings into a single string.

Description:

a, b...: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type. For all other input types, an error is returned.

Returned value: string type. If there is no input or if any input is NULL, NULL is returned.

Example:

```
concat('ab', 'c') = 'abc'  
concat() = null  
concat('a', null, 'b') = null
```

1.5.7.2.4 INSTR

Function declaration:

```
bigint instr(string str1, string str2[, bigint start_position[, bigint  
nth_appearance]])
```

Purpose: It is used to calculate the position of substring str2 in string str1.

Description:

- **str1:** string type. It indicates a string to be searched. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.
- **str2:** string type. It indicates a substring to be searched out. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value

of the string type before this computation. For all other input types, an error is returned.

- **start_position**: bigint type. If it is of another type, an error is returned. It indicates which character in str1 the search will start with. The default start position is the first character, marked as 1.
- **nth_appearance**: bigint type. If it is greater than 0, it indicates the position where the substring matches the string for the nth_appearance time. If it is of another type or if it is less than or equal to 0, an error is returned.

Returned value: bigint type.



Note:

- If str2 is not found in str1, 0 is returned.
- If any input is NULL, NULL is returned.
- If str2 is NULL, the matching will always be successful. Therefore, 1 is returned for instr('abc', '').

Example:

```
instr('Tech on the net', 'e') = 2
instr('Tech on the net', 'e', 1, 1) = 2
instr('Tech on the net', 'e', 1, 2) = 11
instr('Tech on the net', 'e', 1, 3) = 14
```

1.5.7.2.5 IS_ENCODING

Function declaration:

```
boolean is_encoding(string str, string from_encoding, string
to_encoding)
```

Purpose: It is used to determine whether an input string can be converted from a specified character set (from_encoding) to another character set (to_encoding). It can be used to determine whether the input is garbled. from_encoding is usually set to utf-8, and to_encoding is set to gbk.

Description:

- **str**: string type. If the input is NULL, NULL is returned. Null is considered to belong to any character set.
- **from_encoding, to_encoding**: string type. They indicate the source and the destination character sets respectively. If the input is NULL, NULL is returned.

Returned value: boolean type. If a string is converted successfully, true is returned. Otherwise, false is returned.

Example:

```
is_encoding('test', 'utf-8', 'gbk') = true
is_encoding('test', 'utf-8', 'gbk') = true
-- These two traditional Chinese characters are in GBK stock in China.
is_encoding('test', 'utf-8', 'gb2312') = false
-- The grapheme inventory of 'GB2312' does not contain these two
Chinese characters.
```

1.5.7.2.6 KEYVALUE

Function declaration:

```
KEYVALUE(String srcStr, String split1, String split2, String key)
KEYVALUE(String srcStr, String key) //split1 = ";", split2 = ":"
```

Purpose: It is used to split the source string into key-value pairs by split1, separate key-value pairs by split2, and return the value of the corresponding key.

Description:

- **srcStr:** the source string to be split.
- **key:** string type. After the source string is split by 'split1' and 'split2', return the corresponding value according to the specification of the 'key' value.
- **split1 and split2:** strings used as separators. The source string is split by the two separators. If these two parameters are not specified in the expression, split1 is a semicolon (;) and split2 is a colon (:) by default. If a string that has been split by split1 has multiple split2 values, the returned result is undefined.

Returned value: string type.

- If 'split1' or 'split2' is NULL, return NULL.
- If 'srcStr' and 'key' are NULL or if there is no matched 'key', return NULL.
- If multiple 'key-value' matches, return the value corresponding to the first matched key.

Example:

```
keyvalue('0:1\;1:2', 1) = '2'
-- The source string is "0:1\;1:2". Because split1 and split2 are not
specified, split1 is a semicolon (;) and split2 is a colon (:) by
default. After split1 split, the key-value pair is:
0:1\,1:2
After split2 split, it becomes:
0 1/
1 2
```

```

Returns the value(2) of the key corresponding to 1.
keyvalue("\;decreaseStore:1\;xcard:1\;isB2C:1\;tf:21910\;cart:1\;
shipping:2\;pf:0\;market:shoes\;instPayAmount:0\;", "\;",";", "tf") = "
21910"
-- The source string is "\;decreaseStore:1\;xcard:1\;isB2C:1\;tf:21910
\;cart:1\;shipping:2\;pf:0\;market:shoes\;instPayAmount:0\;". After
the source string is split by split1 "\;", the key-value pairs are as
follows:
decreaseStore:1, xcard:1, isB2C:1, tf:21910, cart:1, shipping:2, pf:0
, market:shoes, instPayAmount:0
If split2 is ":", after split it becomes:
decreaseStore 1
xcard 1
isB2C 1
tf 21910
cart 1
shipping 2
pf 0
market shoes
instPayAmount 0
For the key parameter whose value is "tf", the returned value of the
corresponding value parameter is 21910.

```

1.5.7.2.7 LENGTH

Function declaration:

```
bigint length(string str)
```

Purpose: It is used to return the length of a string.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: bigint type. If a string is NULL, NULL is returned. If a string is not UTF-8 encoded, -1 is returned.

Example:

```
length('hi! China') = 6
```

1.5.7.2.8 LENGTHB

Function declaration:

```
bigint lengthb(string str)
```

Purpose: It is used to return the length of a string. Unit: byte.

Description:

str: string type. If the input is of the bigint, double, decimal, or datetime type, it is implicitly converted into a value of the string type before this computation. If the input is of another type, an error is returned.

Returned value: bigint type. If the input is NULL, NULL is returned.

Example:

```
lengthb('hi! china') = 10
```

1.5.7.2.9 MD5

Function declaration:

```
string md5(string value)
```

Purpose: It is used to calculate the MD5 value of the input string value.

Description:

value: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. If the input is of another type, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

1.5.7.2.10 PARSE_URL

Function declaration:

```
STRING PARSE_URL(STRING url, STRING part[,STRING key])
```

Purpose: It is used to parse a URL and extract information by key.

Description:

- If URL or part is NULL, NULL is returned. If URL is invalid, an error is returned.
- **part:** string type. It supports HOST, PATH, QUERY, REF, PROTOCOL, AUTHORITY, FILE, and USERINFO, and is case insensitive. If it is none of the preceding values, an error is returned.
- If part is QUERY, the value in query string that corresponds to the key value is extracted. Otherwise, the parameter key is ignored.

Returned value: string type.

Example:

```
url = file://username:password@example.com:8042/over/there/index.dtb?
type=animal&name=narwhal#nose
parse_url('url', 'HOST') = "example.com"
parse_url('url', 'PATH') = "/over/there/index.dtb"
parse_url('url', 'QUERY') = "type=animal&name=narwhal"
parse_url('url', 'QUERY', 'name') = "narwhal"
parse_url('url', 'REF') = "nose"
parse_url('url', 'PROTOCOL') = "file"
parse_url('url', 'AUTHORITY') = "username:password@example.com:8042"
parse_url('url', 'FILE') = "/over/there/index.dtb? type=animal&name=
narwhal"
parse_url('url', 'USERINFO') = "username:password"
```

1.5.7.2.11 REGEXP_EXTRACT

Command syntax:

```
string regexp_extract(string source, string pattern[, bigint
occurrence])
```

Purpose: It is used to return part of the source string that matches the regular expression and the occurrence of the matches.

Description:

- **source:** string type. It indicates a string to be searched.
- **pattern:** string type. If pattern is NULL or if there is no specified group in pattern, an error is returned.
- **occurrence:** bigint type. It must be a number that is greater than or equal to 0. Otherwise, an error is returned. The default value is 1 if it is not specified. If it is 0, a substring which meets all pattern requirements is returned.

Returned value: string type. If any input is NULL, NULL is returned.

Example:

```
regexp_extract('foothebar', 'foo(. *?)( bar)', 1) = the
regexp_extract('foothebar', 'foo(. *?)( bar)', 2) = bar
regexp_extract('foothebar', 'foo(. *?)( bar)', 0) = foothebar
regexp_extract('8d99d8', '8d(\\d+)d8') = 99
-- If the regular expression is submitted at the MaxCompute client,
two backslashes (\\) are needed to be used as the escape character.
regexp_extract('foothebar', 'foothebar')
```

```
-- An error is returned because no part is specified in the pattern.
```

1.5.7.2.12 REGEXP_INSTR

Function declaration:

```
bigint regexp_instr(string source, string pattern[,bigint start_position[, bigint nth_occurrence[, bigint return_option]])
```

Purpose: It is used to return the start or end position of the substring that matches the pattern in the source string from start_position for the nth_occurrence time.

Description:

- **source:** string type. It indicates a string to be searched.
- **pattern:** a constant of the string type. If pattern is null, an error is returned.
- **start_position:** a constant of 'bigint' type. It is the start position for the search. When it is not specified, it is 1 by default. If it is of another type or less than or equal to 0, an error is returned.
- **nth_occurrence:** a constant of the bigint type. When it is not specified, it is 1 by default, indicating the position where a substring matches pattern in search for the first time. If it is of another type or if it is less than or equal to 0, an error is returned.
- **return_option:** a constant of the bigint type. The value is either 0 or 1. If it is of another type or the value is not supported, an error is returned. 0 indicates that the start position of the matched substring is returned, and 1 indicates that the end position of the matched substring is returned.

Returned value: bigint type. It is the start or end position of the matched substring in source string according to the type specified by return_option. If any input is NULL, NULL is returned.

Example:

```
regexp_instr("i love www.taobao.com", "o[[:alpha:]]{1}", 3, 2) = 14
```

1.5.7.2.13 REGEXP_SUBSTR

Function declaration:

```
string regexp_substr(string source, string pattern[, bigint start_position[, bigint nth_occurrence]])
```

Purpose: It is used to return the string that matches pattern in the source string from position start_position for the nth_occurrence time.

Description:

- **source:** string type. It indicates a string to be searched.
- **pattern:** a constant of the string type. It indicates a pattern to be matched. If pattern is null, an error is returned.
- **start_position:** a constant of the bigint type. It must be greater than 0. If it is another type or if it is less than or equal to 0, an error is reported. When it is not specified, it is regarded as 1 by default, so the matching starts from the first character of 'source'.
- **nth_occurrence:** a constant of the bigint type. It must be greater than 0. If it is another type or is less than or equal to 0, an error is returned. If it is not specified, it is regarded as 1 by default, indicating that the string in the first match is returned.

Returned value: string type. If any input is NULL, NULL is returned. If there is no matching, NULL is returned.

Example:

```
regexp_substr ("I love aliyun very much", "a[[:alpha:]]{5}") = "aliyun"  
regexp_substr('I have 2 apples and 100 bucks!', '[:blank:][:alnum:]]*', 1, 1) = " have"
```

```
regexp_substr('I have 2 apples and 100 bucks!', '[:,blank:][::a\lnum:]]*', 1, 2) = "2"
```

1.5.7.2.14 REGEXP_COUNT

Command syntax:

```
bigint regexp_count(string source, string pattern[, bigint start_position])
```

Purpose: It is used to return the number of occurrences that a string pattern appears in the source string, starting from start_position.

Description:

- **source:** string type. It indicates a string to be searched. For all other input types, an error is returned.
- **pattern:** string type. It indicates a pattern to be matched. If the pattern is NULL or of another type, an error is returned.
- **start_position:** bigint start_position must be a number that is greater than 0. Otherwise, an error is returned. If start_position is not specified, the default value is 1 which means starting from the first character of the source string.

Returned value: bigint type. If any input is NULL, NULL is returned. If there is no matching, 0 is returned.

Example:

```
regexp_count('abababc', 'a.c') = 1  
regexp_count('abcde', '[:,alpha:]]{2}', 3) = 1
```

1.5.7.2.15 SPLIT_PART

Function declaration:

```
string split_part(string str, string delimiter, bigint start[, bigint end])
```

Purpose: It is used to split a string with the specified delimiter, and return the string between the specified start segment and end segment (inclusive).

Description:

- **str:** string type. It indicates a string to be split. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. If the input is of any other type, an error is returned.

- **delimiter:** a constant of the string type. It indicates the delimiter used to split a string. It can be a character or a string. If it is neither a character nor a string, an error is returned.
- **start:** a constant of the bigint type. It must be greater than 0. If it is not a constant or is of a different type, an error is returned. It indicates the start number (starting from 1) of the segment to be returned. If end is not specified, the segment specified by start is returned.
- **end:** a constant of the bigint type. It must be greater than or equal to the value of start; otherwise, an error is returned. It indicates the end number of the segment to be returned. If it is not a constant or is of a different type, an error is returned. If end is not specified, the last segment is returned.

Returned value: string type. If any input is NULL, NULL is returned. If delimiter is NULL, the original string is returned.



Note:

- If delimiter does not exist in str, and start is set to 1, the entire str is returned. If the input is NULL, NULL is returned.
- If start is set to a value greater than the number of segments (for example, the string has 6 segments but the start value is greater than 6), NULL is returned.
- If end is set to a value greater than the number of segments, the string between start and the last segment is returned.

Example:

```
split_part('a,b,c,d', ',', 1) = 'a'
split_part('a,b,c,d', ',', 1, 2) = 'a,b'
split_part('a,b,c,d', ',', 10) = ''
```

1.5.7.2.16 REGEXP_REPLACE

Function declaration:

```
string regexp_replace(string source, string pattern, string replace_string[, bigint occurrence])
```

Purpose: It is used to search a source string for substrings that match a given pattern, replace them with the specified replace_string, and return the result.

Description:

- **source:** string type. It indicates a string to be replaced.

- **pattern:** a constant of the string type. It indicates a pattern to be matched. If pattern is null, an error is returned.
- **replace_string:** string type. It is used to replace the matched pattern.
- **occurrence:** a constant of the bigint type. It must be greater than or equal to 0. This parameter indicates the number of times at which the substring matches the pattern for replacement with replace_string. If the input value is 0, all matched substrings are replaced. If it is of another type or less than 0, an error is returned. It can be omitted. The default value is 0.

Returned value: string type. When the referenced group does not exist, the replace operation is not performed. When the input parameters source, pattern, and occurrence are NULL, NULL is returned. If replace_string is NULL and the pattern is matched, NULL is returned. If replace_string is NULL but the pattern is not matched, the original string is returned.



Note:

When the referenced group does not exist, the action is not defined.

Example:

```
regexp_replace("123.456.7890", "([[:digit:]]{3})\\.[[:digit:]]{3}\\.[[:digit:]]{4})", "(\\1)\\2-\\3", 0) = "(123)456-7890"
regexp_replace("abcd", "(.)", "\\1 ", 0) = "a b c d "
regexp_replace("abcd", "(.)", "\\1 ", 1) = "a bcd"
regexp_replace("abcd", "(.)", "\\2", 1) = "abcd"
-- Only a group is defined in pattern and the referenced second group
is not existent.
-- Please avoid this. The result to reference nonexistent group is not
defined.
regexp_replace("abcd", "(. *)\\.\\$", "\\2", 0) = "d"
regexp_replace("abcd", "a", "\\1", 0) = "bcd"
-- No group definition is in pattern, so '\\1' references a nonexistent
group.
-- Try to avoid this. The result of referencing a nonexistent group is
not defined.
```

1.5.7.2.17 SUBSTR

Function declaration:

```
string substr(string str, bigint start_position[, bigint length])
```

Purpose: It is used to return a substring of 'length' from 'str' starting from 'start_position'.

Description:

- **str**: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other types of inputs, an error is returned.
- **start_position**: bigint type. The start position is 1. If start_position is a negative value, the counting starts from the end to the start of the string and the last character is -1. If the input is of another type, an error is returned.
- **length**: bigint type. It indicates the length of the substring, which is greater than 0. If it is of another type or less than or equal to 0, an error is returned.

Returned value: string type. If any input is NULL, NULL is returned.



Note:

If the length is omitted, the substring from start to end is returned.

Example:

```
substr("abc", 2) = "bc"  
substr("abc", 2, 1) = "b"  
substr("abc",-2,2) = "bc"  
substr("abc",-3) = "abc"
```

1.5.7.2.18 TOLOWER

Function declaration:

```
string tolower(string source)
```

Purpose: It is used to convert 'source' into a lowercase string and return the value.

Description:

source: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
tolower("aBcd") = "abcd"
```

```
tolower("Haha Cd") = "haha cd"
```

1.5.7.2.19 TOUPPER

Function declaration:

```
string toupper(string source)
```

Purpose: It is used to convert 'source' into an uppercase string and return the value.

Description:

source: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other types of inputs, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
toupper("aBcd") = "ABCD"  
toupper("HahaCd") = "HAHACD"
```

1.5.7.2.20 TO_CHAR

Function declaration:

```
string to_char(boolean value)  
string to_char(bigint value)  
string to_char(double value)  
string to_char(decimal value)
```

Purpose: It is used to convert the input of the boolean, bigint, decimal, or double type into a value of the string type.

Description:

value: boolean, bigint, or double type. For all other types of inputs, an error is returned. For more information about the formatted output of data of the datetime type, see [Date processing functions — TO_CHAR](#).

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
to_char(123) = '123'  
to_char(true) = 'TRUE'  
to_char(1.23) = '1.23'
```

```
to_char(null) = 'null'
```

1.5.7.2.21 TRIM

Function declaration:

```
string trim(string str)
```

Purpose: It is used to remove the spaces from both ends of 'str'.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other types of inputs, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

1.5.7.2.22 LTRIM

Function declaration:

```
string ltrim(string str)
```

Purpose: It is used to remove the left spaces for input string str.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
select ltrim(' abc ') from dual;
-- Returned result:
+-----+
| _c0 |
+-----+
| abc |
+-----+
```

1.5.7.2.23 RTRIM

Function declaration:

```
string rtrim(string str)
```

Purpose: It is used to remove the rightmost spaces from the input string 'str'.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
select rtrim('a abc ') from dual;
-- Returned result:
+-----+
| _c0 |
+-----+
| a abc |
+-----+
```

1.5.7.2.24 REVERSE

Function declaration:

```
STRING REVERSE(string str)
```

Purpose: It is used to return a reverse string.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
select reverse('abcedfg') from dual;
-- Returned result:
+-----+
| _c0 |
+-----+
| gfdecba |
+-----+
```

1.5.7.2.25 SPACE

Function declaration:

```
STRING SPACE(bigint n)
```

Purpose: It is used to return a string with 'n' consecutive space characters.

Description:

n: bigint type. The length cannot exceed 2 MB. If the input is NULL, an error is returned.

Returned value: string type.

Example:

```
select length(space(10)) from dual;
-- 10 is returned.
select space(4000000000000) from dual;
-- An error is returned as the length exceeds 2 MB.
```

1.5.7.2.26 REPEAT

Function declaration:

```
STRING REPEAT(string str, bigint n)
```

Purpose: It is used to return string 'str' that has been repeated n times.

Description:

- **str:** string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.
- **n:** bigint type. The length cannot exceed 2 MB. If it is NULL, an error is returned.

Returned value: string type.

Example:

```
select repeat('abc',5) from lxw_dual;
-- abcabcabcabcab is returned.
```

1.5.7.2.27 ASCII

Function declaration:

```
Bigint ASCII(string str)
```

Purpose: It is used to return the ASCII code of the first character of string 'str'.

Description:

str: string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.

Returned value: bigint type.

Example:

```
select ascii('abcde') from dual;  
-- 97 is returned.
```

1.5.7.2.28 URL_ENCODE

Function declaration:

```
STRING URL_ENCODE(String input[, String encoding])
```

Purpose: It is used to encode the input string in the application/x-www-form-urlencoded MIME format:

- a-z and A-Z remain unchanged.
- ":", "-", "*", and "_" remain unchanged.
- Spaces are converted into "+".
- The rest of the characters are converted into byte values according to the specified encoding. If encoding is not specified, UTF-8 is used by default. In this case, each byte value is represented in the %xy format, where xy represents the hexadecimal form of the character.

Description:

- **input:** string type.
- **encoding:** specifies an encoding format. If it is not specified, UTF-8 is used by default.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
url_encode('Example for url_encode:// (fdsf)') = "%E7%A4%BA%E4%BE%  
8Bfor+url_encode%3A%2F%2F+%28fdsf%29"
```

```
url_encode('Example for url_encode :// dsf(fasfs)', 'GBK') = "Example+for+url_encode+%3A%2F%2F+dsf%28fasfs%29"
```

1.5.7.2.29 URL_DECODE

Function declaration:

```
STRING URL_DECODE(STRING input[, STRING encoding])
```

Purpose: It is used to convert an input string from the application/x-www-form-urlencoded MIME format into a normal string. This is the inverse function of URL_ENCODE:

- a–z and A–Z remain unchanged.
- ":", "-", "*", and "_" remain unchanged.
- "+" is converted into a space.
- The %xy formatted sequence is converted into byte values. Consecutive byte values are interpreted as the corresponding strings based on the input encoding.
- Other characters remain unchanged.
- The final returned value of the function is a UTF-8 string.

Description:

- **input:** string type.
- **encoding:** specifies an encoding format. If it is not specified, UTF-8 is used by default.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
url_decode('%E7%A4%BA%E4%BE%8Bfor+url_encode%3A%2F%2F+%28fdsf%29')= "
Example for url_encode:// (fdsf)"
url_decode('Exaple+for+url_encode+%3A%2F%2F+dsf%28fasfs%29', 'GBK') =
"Exaple for url_encode :// dsf(fasfs)" ````
```

1.5.7.2.30 Additional string processing functions

MaxCompute 2.0 provides additional string processing functions. You must add the following SET statement before SQL statements contained in the LPAD, RPAD, and TRANSLATE functions:

```
set odps.sql.type.system.odps2=true;
```



Note:

You must submit and execute the SET statement and the SQL statements of the new functions simultaneously.

The string processing functions described in subsequent topics are new in MaxCompute 2.0.

1.5.7.2.31 CONCAT_WS

Command syntax:

```
string concat_ws(string SEP, string a, string b...)
```

Purpose: It is used to join input strings starting from the second with the first string as the separator.

Description:

- **SEP:** delimiter of the string type. If it is not specified, an error is returned.
- **a, b...:** string type. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type. For all other input types, an error is returned.

Returned value: string type. If there is no input or if any input is NULL, NULL is returned.

Example:

```
concat_ws(':', 'name', 'bob') = 'name:bob'  
concat_ws(':', 'avg', null, '34') = 'null'
```

1.5.7.2.32 LPAD

Function declaration:

```
string lpad(string a, int len, string b)
```

Purpose: It is used to pad the left side of string a with string b until the new padded string has len bits.

Description:

- **len:** int type.
- **a, b:** string type.

Returned value: string type. If len is less than the number of bits in a, a is truncated from the left to obtain a string with the number of bits specified by len. If len is 0, NULL is returned.

Example:

```
lpad('abcdefgh',10,'12')='12abcdefgh'  
lpad('abcdefgh',5,'12')='abcde'  
lpad('abcdefgh',0,'12')  
-- NULL is returned.
```

1.5.7.2.33 RPAD

Function declaration:

```
string rpad(string a, int len, string b)
```

Purpose: It is used to pad the right side of string 'a' with string 'b' until the new padded string has 'len' places.

Description:

- len: int type.
- a, b: string type.

Returned value: string type. If len is smaller than the number of characters in a, a is truncated from the left to obtain a string with the number of characters specified by len. If len is 0, NULL is returned.

Example:

```
rpadd('abcdefgh',10,'12')='abcdefgh12'  
rpadd('abcdefgh',5,'12')='abcde'  
rpadd('abcdefgh',0,'12')  
-- NULL is returned.
```

1.5.7.2.34 REPLACE

Function declaration:

```
string replace(string a, string OLD, string NEW)
```

Purpose: It is used to replace the part of string a that is exactly the same as string OLD with string NEW, and return string a.

Description:

All parameters are of the string type.

Returned value: string type. If any input is NULL, NULL is returned.

Example:

```
replace('ababab','abab','12') = '12ab'  
replace('ababab','cdf','123') = 'ababab'  
replace('123abab456ab',null,'abab') = 'null'
```

1.5.7.2.35 SOUNDEX

Function declaration:

```
string soundex(string a)
```

Purpose: It is used to convert an ordinary string into a soundex string.

Description:

All parameters are of the string type.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
soundex('hello') = 'H400'
```

1.5.7.2.36 SUBSTRING_INDEX

Function declaration:

```
string substring_index(string a, string SEP, int count))
```

Purpose: It is used to return the substring in 'a' that comes before the 'count' (nth) delimiter ('SEP'). If 'count' is a positive value, it starts from the left of the string. If 'count' is a negative value, it starts from the right of the string.

Description:

- a, SEP: string type.
- count: int type.

Returned value: string type. If the input is NULL, NULL is returned.

Example:

```
substring_index('https://help.aliyun.com', '.', 2) = 'https://help.  
aliyun'  
substring_index('https://help.aliyun.com', '.', -2) = 'aliyun.com'
```

```
substring_index('https://help.aliyun.com', null, 2) = 'null'
```

1.5.7.2.37 TRANSLATE

Function declaration:

```
string translate(string|varchar str1, string|varchar str2, string|  
varchar str3)
```

Purpose: It is used to replace str2 in str1 with str3.

Returned value: STRING type. If any input is NULL, NULL is returned.

Example:

```
translate('MaxComputer', 'puter', 'pute')='MaxCompute'  
translate('aaa', 'b', 'c')='aaa'  
translate('MaxComputer', 'puter', null)=null
```

1.5.7.3 Date processing functions

1.5.7.3.1 DATEADD

Function declaration:

```
datetime dateadd(datetime date, bigint delta, string datepart)
```

Purpose: It is used to modify date based on delta and datepart.

Description:

- **date:** This value must be a string type date. If the input is of the string type, it is implicitly converted into a value of the datetime type before this computation. For all other types of inputs, an error is returned.
- **delta:** bigint type. It indicates the scope of modification. If the input is of the string or double type, it is implicitly converted into a value of the bigint type before this computation. If the input is of another type, an error is returned. If delta is greater than 0, the delta is added to the value. If delta is less than 0, the delta is subtracted from the value.
- **datepart:** a constant of the string type. This field is set based on the string-datetime conversion convention. yyyy indicates year and mm indicates month. For rules of type conversion, see [Conversion between the string type and datetime type](#). In addition, the extended date format is also supported: year, month or mon, day, and hour. If the parameter value is not a constant or of an unsupported format or another type, an error is returned.

Returned value: datetime type. If any input is NULL, NULL is returned.



Note:

- When delta is added to or subtracted from the value, carrying and borrowing are base-10 for year, base-12 for month, base-24 for hour, and base-60 for minute and second. If delta is measured in months, the following calculation is applied: If the month in the datetime value does not cause the day value to become invalid after delta is added, the day value is kept. Otherwise, the day value is adjusted to the last day of the resulting month.
- This field is set based on the string-datetime conversion convention. `yyyy` indicates the year and `mm` indicates the month. Unless otherwise specified, all built-in functions related to the datetime type follow this convention. Unless otherwise specified, the datepart of all built-in functions related to the datetime type also supports the extended date format: year, month or mon, day, and hour.

Example:

```
If trans_date = 2017-02-28 00:00:00:
dateadd(trans_date, 1, 'dd') = 2017-03-01 00:00:00
-- Add one day. The result is beyond the last day of February. The
actual value is the first day of next month.
dateadd(trans_date, -1, 'dd') = 2017-02-27 00:00:00
-- Subtract one day.
dateadd(trans_date, 20, 'mm') = 2018-10-28 00:00:00
-- 20 months are added. The month overflows, and 1 is added to the
year.
trans_date = 2017-02-28 00:00:00, dateadd(transdate, 1, 'mm') = 2017-
03-28 00:00:00
trans_date = 2017-01-29 00:00:00, dateadd(transdate, 1, 'mm') = 2017-
02-28 00:00:00
-- February has 28 days only, so the last day of the month is returned
.
trans_date = 2017-03-30 00:00:00, dateadd(transdate, -1, 'mm') = 2017-
02-28 00:00:00
```

The values of `trans_date` used only serve as examples. The datetime examples in this document use simple formats. In MaxCompute SQL, a constant cannot be of the datetime type. The following syntax is incorrect:

```
select dateadd(2017-03-30 00:00:00, -1, 'mm') from tbl1;
```

If you must use a constant of the datetime type, use the following method:

```
select dateadd(cast("2017-03-30 00:00:00" as datetime), -1, 'mm') from
tbl1;
```

```
-- The String type constant is converted to datetime type by explicit conversion.
```

1.5.7.3.2 DATEDIFF

Function declaration:

```
bigint datediff(datetime date1, datetime date2, string datepart)
```

Purpose: It is used to calculate the difference between date1 and date2 based on the specified datepart.

Description:

- **date1 and date2:** minuend and subtrahend of the datetime type respectively. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other input types, an error is returned.
- **datepart:** A constant of the string type. It supports the extended date format. If datepart is not in the specified format or is of another type, an error is returned.

Returned value: bigint type. If any input is NULL, NULL is returned. If date1 is less than date2, the returned value may be negative.



Note:

The lower unit part is truncated based on 'datepart' in the computation process and then the result is calculated.

Example:

```
If start = 2017-12-31 23:59:59 and end = 2018-01-01 00:00:00:  
datediff(end, start, 'dd') = 1  
datediff(end, start, 'mm') = 1  
datediff(end, start, 'yyyy') = 1  
datediff(end, start, 'hh') = 1  
datediff(end, start, 'mi') = 1  
datediff(end, start, 'ss') = 1  
datediff('2017-05-31 13:00:00', '2017-05-31 12:30:00', 'ss') = 1800  
datediff('2017-05-31 13:00:00', '2017-05-31 12:30:00', 'mi') = 30
```

1.5.7.3.3 DATEPART

Function declaration:

```
bigint datepart(datetime date, string datepart)
```

Purpose: It is used to extract the value of the specified datepart in date.

Description:

- **date:** datetime type. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other input types, an error is returned.
- **datepart:** a constant of the string type. It supports the extended date format. If datepart is not in the specified format or is of another type, an error is returned.

Returned value: bigint type. If any input is NULL, NULL is returned.

Example:

```
datepart('2017-06-08 01:10:00', 'yyyy') = 2017  
datepart('2017-06-08 01:10:00', 'mm') = 6
```

1.5.7.3.4 DATETRUNC

Function declaration:

```
datetime datetrunc (datetime date,string datepart)
```

Purpose: It is used to return the value of a date after the specified datepart is truncated.

Description:

- **date:** datetime type. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other input types, an error is returned.
- **datepart:** a constant of the string type. It supports the extended date format. If datepart is not in the specified format or is of another type, an error is returned.

Returned value: datetime type. If any input is NULL, NULL is returned.

Example:

```
datetrunc('2017-12-07 16:28:46', 'yyyy') = 2017-01-01 00:00:00  
datetrunc('2017-12-07 16:28:46', 'month') = 2017-12-01 00:00:00  
datetrunc('2017-12-07 16:28:46', 'DD') = 2017-12-07 00:00:00
```

1.5.7.3.5 GETDATE

Function declaration:

```
datetime getdate()
```

Purpose: It is used to obtain the current system time. Use UTC+8 as the standard time of MaxCompute.

Returned value: the current date and time of the datetime type.



Note:

In a MaxCompute SQL task (executed in a distributed manner), 'getdate' always returns a fixed value. The returned result is any time in MaxCompute. The time returned is precise to the second. In later versions, the time will be precise to the millisecond.

1.5.7.3.6 ISDATE

Function declaration:

```
boolean isdate(string date, string format)
```

Purpose: It is used to determine whether a date string can be converted into a date value based on the corresponding format string. If the conversion can be performed, true is returned. Otherwise, false is returned.

Description:

- **date:** This value must be a string type date. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other input types, an error is returned.
- **format:** a constant of the string type. The extended date format is not supported. If it is of another type or an unsupported format, an error is returned. If there are redundant format strings appearing in 'format', the date value corresponding to the first format string is used. Other strings are taken as delimiters. If isdate("1234-yyyy", "yyyy-yyyy"), true is returned.

Returned value: boolean type. If any input is NULL, NULL is returned.

1.5.7.3.7 LASTDAY

Function declaration:

```
datetime lastday(datetime date)
```

Purpose: It is used to return the last day of the current month to which the date belongs. The value is accurate to day. The hour, minute, and second part is expressed as 00:00:00.

Description:

date: datetime type. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other input types, an error is returned.

Returned value: datetime type. If the input is NULL, NULL is returned.

1.5.7.3.8 TO_DATE

Function declaration:

```
datetime to_date(string date, string format)
```

Purpose: It is used to convert the 'date' string into a date value.

Description:

- **date:** string type. It indicates the date value of the string type to be converted. If the input is of the bigint, decimal, double, or datetime type, it is implicitly converted into a value of the string type before this computation. For all other types of inputs or NULL, an error is returned.
- **format:** a constant of the string type in the date format. For all other types of inputs and non-constant values, an error is returned. It does not support the extended date format. Other characters are ignored as invalid characters in parsing. The format parameter must contain yyyy. Otherwise, an error is returned. If there are redundant format strings in the format, the corresponding date value of the first format string is used, and the rest are processed as separators. For example, `to_date('1234-2234', 'yyyy-yyyy')` returns '1234-01-01 00:00:00'.

Returned value: datetime type. The format is `yyyy-mm-dd hh:mi:ss`. If any input is NULL, NULL is returned.

Example:

```
to_date('Alibaba2017-12*03', 'Alibabayyyy-mm*dd') = 2017-12-03 00:00:00
to_date('20170718', 'yyyymmdd') = 2017-07-18 00:00:00
to_date('201707182030', 'yyyymmddhhmi')=2017-07-18 20:30:00
to_date('2017718', 'yyyymmdd')
-- Invalid format. NULL is returned.
to_date('Alibaba2017-12*3', 'Alibabayyyy-mm*dd')
-- Invalid format. NULL is returned.
to_date('2017-24-01', 'yyyy')
```

```
-- Invalid format. NULL is returned.
```

1.5.7.3.9 TO_CHAR

Function declaration:

```
string to_char(datetime date, string format)
```

Purpose: It is used to convert a value of the date type into a string based on the specified format.

Description:

- **date:** date value of the datetime type to be converted. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other types of inputs, an error is returned.
- **format:** a constant of the string type. If it is not a constant or is of a different type, an error is returned. In format, the date format part is replaced with the corresponding data and other characters are output directly.

Returned value: string type. If any input is NULL, NULL is returned.

Example:

```
to_char('2017-12-03 00:00:00', 'Alibabayyyy-mm*dd') = 'Alibaba2017-12*03'  
to_char('2017-07-18 00:00:00', 'yyyymmdd') = '20170718'  
to_char('Alibaba 2017-12*3', 'Alibaba yyyy-mm*dd')  
-- Null is returned.  
to_char('2017-24-01', 'yyyy')  
-- Null is returned.  
to_char('2017718', 'yyyymmdd')  
-- Null is returned.
```



Note:

For more information about conversion from other types into the string type, see [String functions — TO_CHAR](#).

1.5.7.3.10 UNIX_TIMESTAMP

Function declaration:

```
bigint unix_timestamp(datetime date)
```

Purpose: It is used to convert a date into a datetime value of the integer type in the Unix format.

Description:

date: datetime type. It indicates the date. If the input is a string, it is implicitly converted into a value of the datetime type before this computation. For all other input types, an error is returned.

Returned value: bigint type. It indicates the date value in Unix format. If date is NULL, NULL is returned.

1.5.7.3.11 FROM_UNIXTIME

Function declaration:

```
datetime from_unixtime(bigint unixtime)
```

Purpose: It is used to convert a Unix date value from the BIGINT type to the DATETIME type.

Description:

unixtime: BIGINT type. It is a date value in the Unix format. If the input is of the STRING, DECIMAL, or DOUBLE type, it is implicitly converted into a value of the BIGINT type before computation.

Returned value: DATETIME type. If unixtime is NULL, NULL is returned.



Note:

In the HIVE-compatible mode (where `set odps.sql.hive.compatible=true`; has been executed), if the input is of the STRING type, a date value of the STRING type is returned.

Example:

```
from_unixtime(123456789) = 1973-11-30 05:33:09;
```

1.5.7.3.12 WEEKDAY

Function declaration:

```
bigint weekday (datetime date)
```

Purpose: It is used to return the day of week for the specified date.

Description:

date: datetime type. If the input is of the string type, it is implicitly converted to a value of the datetime type before this computation. For all other input types, an error is returned.

Returned value: bigint type. If the input is NULL, NULL is returned. Monday is the first day of a week and the returned value is 0. Days are numbered in ascending order starting from 0. If the day is Sunday, the returned value is 6.

1.5.7.3.13 WEEKOFYEAR

Function declaration:

```
bigint weekofyear(datetime date)
```

Purpose: It is used to return the calendar week of the year that the specified date falls in. The system uses Monday as the first day of the week.



Note:

If a week extends into the next year, the week belongs to the year containing four days or more. If more days fall in the first year, the week is considered as the last week of the first year. If more days fall in the second year, the week is considered as the first week of the second year.

Description:

date: the date of the datetime type. If the input is of the string type, it is implicitly converted to a value of the datetime type before this computation. For all other input types, an error is returned.

Returned value: bigint type. If the input is NULL, NULL is returned.

Example:

```
select weekofyear(to_date("20171229", "yyyymmdd")) from dual;
Returned value:
+-----+
| _c0    |
+-----+
| 1       |
+-----+
-- 20171229 is in year 2017, but the most days of the week are in year
   2018. Therefore, the returned value is 1, which indicates the first
   week of year 2018.
select weekofyear(to_date("20171231", "yyyymmdd")) from dual;
-- 1 is returned.
select weekofyear(to_date("20181229", "yyyymmdd")) from dual;
```

```
-- The returned value is 53.
```

1.5.7.3.14 Additional date functions

MaxCompute 2.0 provides additional date functions. You must add the following SET statement before SQL statements contained in the date functions:

```
set odps.sql.type.system.odps2=true;
```



Note:

You must submit and execute the SET statement and the SQL statements of the new functions simultaneously.

Example:

```
set odps.sql.type.system.odps2=true;
select year('2017-01-01 12:30:00') = 2017 from dual;
```

The date functions described in subsequent topics are new in MaxCompute 2.0.

1.5.7.3.15 YEAR

Function declaration:

```
INT year(string date)
```

Purpose: It is used to return the year of the specified date.

Description:

date: the date of the string type. The date format must include yyyy-mm-dd and have no redundant strings. Otherwise, NULL is returned.

Returned value: int type.

Example:

```
year('2017-01-01 12:30:00') = 2017
year('2017-01-01') = 2017
year('17-01-01') = 17
year(2017-01-01) = null
year('2017/03/09') = null
```

```
year(null) = null
```

1.5.7.3.16 QUARTER

Command syntax:

```
int quarter(datetime/timestamp/string date)
```

Purpose: It is used to return the quarter of the input date, ranging from 1 to 4.

Description:

date: datetime, timestamp, or string type. The date format must include yyyy-mm-dd and have no redundant strings. Otherwise, NULL is returned.

Returned value: int type. If the input is NULL, NULL is returned.

Example:

```
quarter('2017-11-12 10:00:00') = 4  
quarter('2017-11-12') = 4
```

1.5.7.3.17 MONTH

Function declaration:

```
INT month(string date)
```

Purpose: It is used to return the month of the input date.

Description:

date: This value must be a date of the string type. For all other input types, an error is returned.

Returned value: int type.

Example:

```
month('2017-09-01') = 9  
month('20170901') = null
```

1.5.7.3.18 DAY

Function declaration:

```
INT day(string date)
```

Purpose: It is used to return the day of the input date.

Description:

date: This value must be a string type date. For all other input types, an error is returned.

Returned value: int type.

Example:

```
day('2017-09-01') = 1  
day('20170901') = null
```

1.5.7.3.19 DAYOFMONTH

Function declaration:

```
INT dayofmonth(date)
```

Purpose: It is used to return the day of the month for the input date.

Description:

date: This value must be a string type date. For all other input types, an error is returned.

Returned value: int type.

Example:

```
dayofmonth('2017-09-01') = 1  
dayofmonth('20170901') = null
```

1.5.7.3.20 HOUR

Function declaration:

```
INT hour(string date)
```

Purpose: It is used to return the hour of the input date.

Description:

date: This value must be a string type date. For all other input types, an error is returned.

Returned value: int type.

Example:

```
hour('2017-09-01 12:00:00') = 12  
hour('12:00:00') = 12
```

```
hour('20170901120000') = null
```

1.5.7.3.21 MINUTE

Function declaration:

```
INT minute(string date)
```

Purpose: It is used to return the minute of the input date.

Description:

date: This value must be a string type date. For all other input types, an error is returned.

Returned value: int type.

Example:

```
minute('2017-09-01 12:30:00') = 30  
minute('12:30:00') = 30  
minute('20170901120000') = null
```

1.5.7.3.22 SECOND

Function declaration:

```
INT second(string date)
```

Purpose: It is used to return the second of the input date.

Description:

date: This value must be a string type date. For all other input types, an error is returned.

Returned value: int type.

Example:

```
second('2017-09-01 12:30:45') = 45  
second('12:30:45') = 45
```



```
second('20170901123045') = null
```

1.5.7.3.23 FROM_UTC_TIMESTAMP

Function declaration:

```
timestamp from_utc_timestamp({any primitive type}*, string timezone)
```

Purpose: It is used to convert a UTC timestamp to a timestamp for a specified timezone.

Description:

- **{any primitive type}*:** the timestamp. The type can be **TIMESTAMP**, **DATETIME**, **TINYINT**, **SMALLINT**, **INT**, or **BIGIN**.
- **timezone:** Specifies the destination timezone, such as **PST**.

Returned value: **DATETIME** type.

Example:

```
select from_utc_timestamp(1501557840,'PST') = '1970-01-18 09:05:57.84'  
select from_utc_timestamp('1970-01-30 16:00:00','PST') = '1970-01-30  
08:00:00.0'  
select from_utc_timestamp('1970-01-30','PST') = '1970-01-29 16:00:00.0'  
,
```

1.5.7.3.24 CURRENT_TIMESTAMP

Function declaration:

```
timestamp current_timestamp()
```

Purpose: The current timestamp is returned as a Timestamp-type value. The value is not fixed.

Returned value: timestamp type.

Example:

```
select current_timestamp() from dual;
```

```
-- '2017-08-03 11:50:30.661' is returned.
```

1.5.7.3.25 ADD_MONTHS

Function declaration:

```
string add_months(string startdate, int nummonths)
```

Purpose: It is used to return the date, which is 'nummonths' months later than 'startdate'.

Description:

- **startdate:** This value must be a string type date. The date format must contain yyyy-mm-dd. Otherwise, NULL is returned.
- **num_months:** int type.

Returned value: This value must be a string type date. The format is yyyy-mm-dd.

Example:

```
Add_months ('2017-02-14', 3) = '2017-05-14'  
add_months('17-2-14',3) = '0017-05-14'  
add_months('2017-02-14 21:30:00',3) = '2017-05-14'  
add_months('20170214',3) = null
```

1.5.7.3.26 LAST_DAY

Function declaration:

```
string last_day(string date)
```

Purpose: It is used to return the last date of the month.

Description:

date: string type. The format is yyyy-MM-dd HH:mi:ss or yyyy-mm-dd.

Returned value: This value must be a datetime type date. The format is yyyy-mm-dd.

Example:

```
last_day('2017-03-04') = '2017-03-31'  
last_day('2017-07-04 11:40:00') = '2017-07-31'
```

```
last_day('20170304') = null
```

1.5.7.3.27 NEXT_DAY

Function declaration:

```
string next_day(string startdate, string week)
```

Purpose: It is used to return the next date that is later than startdate and matches the week value. That is, the date of the day specified of the next week.

Description:

- **startdate:** string type. The format is yyyy-MM-dd HH:mi:ss or yyyy-mm-dd.
- **week:** string type. The name of a day, or the first 2 or 3 letters of the day, for example, Mo, TUE, or FRIDAY.

Returned value: This value must be a string type date. The format is yyyy-mm-dd.

Example:

```
next_day('2017-08-01','TU') = '2017-08-08'  
next_day('2017-08-01 23:34:00','TU') = '2017-08-08'  
Next_day ('20170801 ', 'tu') = NULL
```

1.5.7.3.28 MONTHS_BETWEEN

Function declaration:

```
double months_between(datetime/timestamp/string date1, datetime/  
timestamp/string date2)
```

Purpose: It is used to return the number of months between date1 and date2.

Description:

- **date1:** datetime, timestamp, or string type. The format is yyyy-MM-dd HH:mi:ss or yyyy-mm-dd.
- **date2:** datetime, timestamp, or string type. The format is yyyy-MM-dd HH:mi:ss or yyyy-mm-dd.

Returned value: double type.

- If 'date1' is later than 'date2', the returned value is positive. If 'date2' is later than 'date1', the returned value is negative.

- When date1 and date2 correspond to the last days of two months, the returned value is an integer representing the number of months. Otherwise, the formula is (date1 - date2)/31.

Example:

```
months_between('1997-02-28 10:30:00', '1996-10-30') = 3.9495967741935485
months_between('1996-10-30', '1997-02-28 10:30:00' ) = -3.9495967741935485
months_between('1996-09-30', '1996-12-31') = -3.0
```

1.5.7.4 Window functions

1.5.7.4.1 Overview

In MaxCompute SQL statements, you can use the window function to analyze and process data flexibly. The window function can only appear in SELECT clauses. It does not support nested Window or aggregation functions. The window function cannot be used with the same-level aggregation functions at the same time.

A MaxCompute SQL statement supports up to five window functions.

Syntax:

```
window_func() over (partition by col1, [col2...]
[order by col1 [asc|desc][, col2[asc|desc]...]] windowing_clause)
```

Description:

- **PARTITION BY** specifies partition columns. The rows on which the partition column values are the same are considered to be in the same window. A window can contain up to 100 million rows of data (we recommend that the number of rows does not exceed 5 million). Otherwise, an error is returned.
- Use **ORDER BY** to specify the rule for sorting data in a window.

- You can use **ROWS** in **windowing_clause** to specify the partitioning method.

There are two methods:

- **rows between x preceding|following and y preceding|following**
indicates a window range from the xth row preceding or following the current row to the yth row preceding or following the current row.
- **rows x preceding|following** indicates a window range from the xth row preceding or following the current row to the current row.



Note:

- **x and y must be integer constants greater than or equal to 0. Their values range from 0 to 10,000. 0 indicates the current row.**
- **You must specify ORDER BY before using ROWS to specify a window range.**
- **Not all window functions open windows using the method specified by ROWS. The method is only supported by the following functions: AVG, COUNT, MAX, MIN, STDDEV, and SUM.**

1.5.7.4.2 COUNT

Command syntax:

```
bigint count([distinct] expr) over(partition by col1[, col2...]
[order by col1 [asc|desc][, col2[asc|desc]...]] [windowing_clause])
```

Purpose: It is used to return the number of values on the **expr** column.

Description:

- **expr:** any type. When it is NULL, this row is not involved in computation. If the **distinct** keyword is specified, this parameter indicates that only distinct values are counted.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The count value of **expr** in the current window is returned if **ORDER BY** is not set. The returned results are sorted in the specified order if **ORDER BY** is specified, and the value is the count value from the start row to the current row in the current window.

Returned value: bigint.



Note:

If the distinct keyword is specified, ORDER BY cannot be used.

Example:

The user_id column of the bigint type exists in the test_src table.

```
select user_id,count(user_id) over (partition by user_id) as count
from test_src;
```

user_id	count
1	3
1	3
1	3
2	1
3	1

```
-- If ORDER BY is not specified, the number of values on the user_id
column from the current partition is returned.
```

```
select user_id,count(user_id) over (partition by user_id order by
user_id) as count from test_src;
```

user_id	count
1	1
1	2
1	3
2	1
3	1

```
-- If ORDER BY is specified, the count value from the start row to the
current row from the current partition is returned.
```

1.5.7.4.3 AVG

Function declaration:

```
avg([distinct] expr) over(partition by col1[, col2...]
[order by col1 [asc|desc] [, col2[asc|desc]...]] [windowing_clause])
```

Purpose: It is used to calculate the average value.

Description:

- **distinct:** If the distinct keyword is specified, this parameter indicates that the average value of distinct values is calculated.
- **expr:** double type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before computation. If the input is of another type, an error is returned. If the input is NULL, this row is not used in computation. The input cannot be of the boolean type.
- **partition by col1[, col2]...:** specifies the partitions used in the computation.

- **order by col1 [asc|desc], col2[asc|desc]:** The count value of expr in the current window is returned if ORDER BY is not set. The returned results are sorted in the specified order if ORDER BY is specified, and the value is the count value from the start row to the current row in the current window.

Returned value: double type.



Note:

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.4 MAX

Function declaration:

```
max([distinct] expr) over(partition by col1[, col2...]
[order by col1 [asc|desc][, col2[asc|desc]...]] [windowing_clause])
```

Purpose: It is used to return the maximum value.

Description:

- **expr:** any types except the boolean type. If the value is NULL, the corresponding row is not involved in the operation. If the distinct keyword is specified, this parameter indicates that the maximum value of the distinct values is taken (whether this parameter is set or not does not affect the result).
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The maximum value in the current window is returned if ORDER BY is not set. If ORDER BY is set, the returned results are sorted in the specified order, and the values are the maximum values from the start row to the current row in the current window.

Returned value: The type is the same as that of expr.



Note:

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.5 MIN

Function declaration:

```
min([distinct] expr) over(partition by col1[, col2...]
[order by col1 [asc|desc][, col2[asc|desc]...]] [windowing_clause])
```

Purpose: It is used to return the minimum value.

Description:

- **expr:** any types except the boolean type. If the value is NULL, the corresponding row is not involved in the operation. If the distinct keyword is specified, this parameter indicates that the minimum value of distinct values is taken (whether this parameter is set or not does not affect the result).
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The minimum value in the current window is returned if ORDER BY is not set. If ORDER BY is set, the returned results are sorted in the specified order, and the returned value is the minimum value in the current window from the start row to the current row.

Returned value: The type is the same as that of expr.

**Note:**

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.6 MEDIAN

Function declaration:

```
double median(double number) over(partition by col1[, col2...])
decimal median(decimal number) over(partition by col1[,col2...])
```

Purpose: It is used to calculate the median.

Description:

- **number:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned. If the input is NULL, NULL is returned.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.

Returned value: double type.

1.5.7.4.7 STDDEV

Function declaration:

```
double stddev([distinct] expr) over(partition by col1[, col2...] [order
by col1 [asc|desc][, col2[asc|desc]...]] [windowing_clause])
decimal stddev([distinct] expr) over(partition by col1[,col2...] [order
by col1 [asc|desc][, col2[asc|desc]...]] [windowi ng_clause])
```

Purpose: It is used to calculate the population standard deviation.

Description:

- **expr:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned. If the input value is NULL, then NULL is returned. If the distinct keyword is specified, this parameter indicates that the population standard deviation of distinct values is calculated.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The population standard deviation of the current window is returned if ORDER BY is not set. If ORDER BY is set, the returned results are sorted in the specified order, and the values are the population standard deviation of the start row to the current row in the current window.

Returned value: When the input is of the decimal type, a value of the decimal type is returned. Otherwise, a value of the double type is returned.



Note:

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.8 STDDEV_SAMP

Function declaration:

```
double stddev_samp([distinct] expr) over(partition by col1[, col2...] [
order by col1 [asc|desc][, col2[asc|desc]...] [windowing_clause])
decimal stddev_samp([distinct] expr) over(partition by col1[,col2...] [
order by col1 [asc|desc][, col2[asc|desc]...] [windowing_clause])
```

Purpose: It is used to calculate the sample standard deviation.

Description:

- **expr:** double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned. If the input is NULL, NULL is returned. If the distinct keyword is specified, this parameter indicates that the sample standard deviation of distinct values is calculated.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The sample standard deviation of the current window is returned if ORDER BY is not set. If ORDER BY is set, the

returned results are sorted in the specified order, and the values are the sample standard deviation of the start row to the current row in the current window.

Returned value: When the input is of the decimal type, a value of the decimal type is returned. Otherwise, a value of the double type is returned.



Note:

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.9 SUM

Function declaration:

```
sum([distinct] expr) over(partition by col1[, col2...]  
[order by col1 [asc|desc][, col2[asc|desc]...]] [windowing_clause])
```

Purpose: It is used calculate the sum.

Description:

- **expr:** double, decimal, or bigint type. If the input is of the string type, it is implicitly converted into a value of the double type before computation. If the input is of another type, an error is returned. If the value is NULL, this row is not calculated. If the distinct keyword is specified, this parameter indicates that the sum of distinct values is calculated.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** The sum of the expr value in the current window is returned if ORDER BY is not set. If ORDER BY is set, the returned results are sorted in the order specified. The returned results are the cumulative sum of start row to the current row in the current window.

Returned value: When the input is of the bigint type, a value of the bigint type is returned. When the input is of the double or string type, a value of the double type is returned.



Note:

If the distinct keyword is specified, ORDER BY cannot be set.

1.5.7.4.10 DENSE_RANK

Function declaration:

```
bigint dense_rank() over(partition by col1[, col2...]
```

```
order by col1 [asc|desc][, col2[asc|desc]...])
```

Purpose: It is used to calculate the consecutive ranking of values. The data in the same row of col2 has the same rank.

Description:

- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** This parameter specifies the value for deciding the ranking.

Returned value: bigint type.

Example:

The emp table contains the following data:

```
| empno | ename | job | mgr | hiredate | sal | comm | deptno |
7369, SMITH, CLERK, 7902, 1980-12-17 00:00:00, 800, , 20
7499, ALLEN, SALESMAN, 7698, 1981-02-20 00:00:00, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 1981-02-22 00:00:00, 1250, 500, 30
7566, JONES, MANAGER, 7839, 1981-04-02 00:00:00, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 1981-09-28 00:00:00, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 1981-05-01 00:00:00, 2850, , 30
7782, CLARK, MANAGER, 7839, 1981-06-09 00:00:00, 2450, , 10
7788, SCOTT, ANALYST, 7566, 1987-04-19 00:00:00, 3000, , 20
7839, KING, PRESIDENT, , 1981-11-17 00:00:00, 5000, , 10
7844, TURNER, SALESMAN, 7698, 1981-09-08 00:00:00, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 1987-05-23 00:00:00, 1100, , 20
7900, JAMES, CLERK, 7698, 1981-12-03 00:00:00, 950, , 30
7902, FORD, ANALYST, 7566, 1981-12-03 00:00:00, 3000, , 20
7934, MILLER, CLERK, 7782, 1982-01-23 00:00:00, 1300, , 10
7948, JACCKA, CLERK, 7782, 1981-04-12 00:00:00, 5000, , 10
7956, WELAN, CLERK, 7649, 1982-07-20 00:00:00, 2450, , 10
7956, TEBAGE, CLERK, 7748, 1982-12-30 00:00:00, 1300, , 10
```

To obtain their serial number, the employees must be group by their departments and sorted by SAL in descending order.

```
SELECT deptno
       , ename
       , sal
       , DENSE_RANK() OVER (PARTITION BY deptno ORDER BY sal DESC) AS
  nums
-- DEPTNO (department) is the partition used in the computation, and
-- SAL (salary) is used as basis for sorting returned results.
FROM emp;
-- Returned result:
+-----+-----+-----+-----+
| deptno | ename | sal   | nums |
+-----+-----+-----+-----+
| 10     | JACCKA | 5000.0 | 1    |
| 10     | King  | 5000.0 | 1    |
| 10     | CLARK | 2450.0 | 2    |
| 10     | WELAN | 2450.0 | 2    |
| 10     | TEBAGE | 1300.0 | 3    |
| 10     | Miller | 1300.0 | 3    |
```

20		SCOTT	3000.0	1	
20	Ford	3000.0	1		
20	JONES	2975.0	2		
20	ADAMS	1100.0	3		
20	SMITH	800.0	4		
30		BLAKE	2850.0	1	
30		ALLEN	1600.0	2	
30		TURNER	1500.0	3	
30		MARTIN	1250.0	4	
30	WARD	1250.0	4		
30	JAMES	950.0	5		

1.5.7.4.11 RANK

Command syntax:

```
bigint rank() over(partition by col1[, col2...] order by col1 [asc|desc]
[, col2[asc|desc]...])
```

Purpose: It is used to return a ranking value. The ranking of the same row data with col2 drops.

Description:

- **partition by col2[, col2..]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** specifies the rule for deciding the ranking.

Returned value: bigint type.

Example:

Table emp contains the following data:

```
| empno | ename | job | mgr | hiredate | sal | comm | deptno |
7369, SMITH, CLERK, 7902, 1980-12-17 00:00:00, 800, , 20
7499, ALLEN, SALESMAN, 7698, 1981-02-20 00:00:00, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 1981-02-22 00:00:00, 1250, 500, 30
7566, JONES, MANAGER, 7839, 1981-04-02 00:00:00, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 1981-09-28 00:00:00, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 1981-05-01 00:00:00, 2850, , 30
7782, CLARK, MANAGER, 7839, 1981-06-09 00:00:00, 2450, , 10
7788, SCOTT, ANALYST, 7566, 1987-04-19 00:00:00, 3000, , 20
7839, KING, PRESIDENT, , 1981-11-17 00:00:00, 5000, , 10
7844, TURNER, SALESMAN, 7698, 1981-09-08 00:00:00, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 1987-05-23 00:00:00, 1100, , 20
7900, JAMES, CLERK, 7698, 1981-12-03 00:00:00, 950, , 30
7902, FORD, ANALYST, 7566, 1981-12-03 00:00:00, 3000, , 20
7934, MILLER, CLERK, 7782, 1982-01-23 00:00:00, 1300, , 10
7948, JACCKA, CLERK, 7782, 1981-04-12 00:00:00, 5000, , 10
7956, WELAN, CLERK, 7649, 1982-07-20 00:00:00, 2450, , 10
```

```
7956,TEBAGE,CLERK,7748,1982-12-30 00:00:00,1300,,10
```

Now group the employees by department. Sort the employees in each group in descending order based on the salary. Each employee obtains a number that represents their position in the group.

```
SELECT deptno
       , ename
       , sal
       , RANK() OVER (PARTITION BY deptno ORDER BY sal DESC) AS nums
-- DEPTNO (department) is the partitioning column. The sal column is
-- sorted to generate the ranking value for each employee.
FROM emp;
```

-- Returned result:

deptno	ename	sal	nums
10	JACCKA	5000.0	1
10	KING	5000.0	1
10	CLARK	2450.0	3
10 WELAN	2450.0	3	
10	TEBAGE	1300.0	5
10 MILLER	1300.0	5	
20	SCOTT	3000.0	1
20	FORD	3000.0	1
20	JONES	2975.0	3
20	ADAMS	1100.0	4
20	SMITH	800.0	5
30	BLAKE	2850.0	1
30	ALLEN	1600.0	2
30	TURNER	1500.0	3
30	MARTIN	1250.0	4
30	WARD	1250.0	4
30	JAMES	950.0	6

1.5.7.4.12 LAG

Function declaration:

```
lag(expr, bigint offset, default) over(partition by col1[, col2...] [
order by col1 [asc|desc][, col2[asc|desc]...]])
```

Purpose: It is used to retrieve the value in the row with a negative offset from the current row. For example, if the current row is *rn*, the value retrieved is from the row *rn - offset*.

Description:

- **expr:** any type.
- **offset:** a constant of the bigint type. If the input is of the string or double type, it is implicitly converted into a value of the bigint type before computation, and the offset is greater than 0.

- **default:** a constant. It specifies the default value when the offset is out of the valid range. The default value is NULL.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** indicates the sorting order of the returned results.

Returned value: The type is the same as that of expr.

1.5.7.4.13 LEAD

Function declaration:

```
lead(expr, bigint offset, default) over(partition by col1[, col2...][  
order by col1 [asc|desc][, col2[asc|desc]...])
```

Purpose: It is used to retrieve the value in the row with a positive offset from the current row. For example, if the current row is *rn*, the value retrieved is from the row *rn + offset*.

Description:

- **expr:** any type.
- **offset:** a constant of the bigint type. If the input is of the string or double type, it is implicitly converted into a value of the bigint type before computation, and the offset is greater than 0.
- **default:** a constant. It specifies the default value when the offset is out of the valid range.
- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], > col2[asc|desc]:** indicates the sorting order of the returned results.

Returned value: The type is the same as that of expr.

Example:

```
select c_double_a,c_string_b,c_int_a,lead(c_int_a,1) over(partition by  
c_double_a order by c_string_b) from dual;  
select c_string_a,c_time_b,c_double_a,lead(c_double_a,1) over(  
partition by c_string_a order by c_time_b) from dual;
```

```
select c_string_in_fact_num,c_string_a,c_int_a,lead(c_int_a) over(
partition by c_string_in_fact_num order by c_string_a) from dual;
```

1.5.7.4.14 PERCENT_RANK

Function declaration:

```
percent_rank() over(partition by col1[, col2...] order by col1 [asc|desc]
[, col2[asc|desc]...])
```

Purpose: It is used to return the relative ranking of a row in a group of data.

Description:

- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], col2[asc|desc]:** specifies the value for the ranking.

Returned value: double type. **Value range:** 0 to 1. The relative ranking is calculated using the following formula: $(\text{rank}-1)/(\text{number of rows}-1)$.



Note:

The number of rows in a window cannot exceed 10,000,000.

1.5.7.4.15 ROW_NUMBER

Function declaration:

```
row_number() over(partition by col1[, col2...] order by col1 [asc|desc]
[, col2[asc|desc]...])
```

Purpose: It is used to calculate the row number, which starts from 1.

Description:

- **partition by col1[, col2...]:** specifies the partitions used in the computation.
- **order by col1 [asc|desc], > col2[asc|desc]:** indicates the sorting value of the returned result.

Returned value: bigint type.

Example:

If table emp contains the following data:

```
| Empno | ename | job | Mgr | hiredate | Sal | REM | deptno |
7369, Smith, clerk, maid-12-17 00:00:00, 800, 20
7499, Allen, salesman, maid-02-20 00:00:00, 1600,300, 30
7521, Ward, salesman, maid-02-22 00:00:00, 1250,500, 30
7566, Jones, Manager, fig-04-02 00:00:00, 2975, 20
7654 Martin, salesman, fig-09-28 00:00:00, fig, 30
```

```

7698, Blake, Manager, fig-05-01 00:00:00, 2850, 30
7782, Clark, Manager, fig-06-09 00:00:00, 2450, 10
7788, Scott, analyst, fig-04-19 00:00:00, 3000, 20
00:00:00, King, President, 1991-11-17 5000, 7839, 10
7844, Turner, salesman, fig-09-08 00:00:00, 1500,0, 30
7876, Adams, clerk, maid-05-23 00:00:00, 1100, 20
7900 James, clerk, maid-12-03 00:00:00, 950, 30
7902 Ford, analyst, fig-12-03 00:00:00, 3000, 20
7934 Miller, clerk, fig-01-23 00:00:00, 1300, 10
7948, jaccka, clerk, fig-04-12 00:00:00, 5000, 10
7956, welan, clerk, fig-07-20 00:00:00, 2450, 10
7956,TEBAGE,CLERK,7748,1982-12-30 00:00:00,1300,,10

```

Now, all employees need to be grouped by department, and each group must be sorted in descending order according to SAL to obtain the serial number in own group.

```

SELECT deptno
       , ename
       , sal
       , ROW_NUMBER() OVER (PARTITION BY deptno ORDER BY sal DESC) AS
  nums
-- DEPTNO (department) is the partition used in the computation, and
SAL (salary) is used as basis for sorting results.
FROM emp;
-- Returned result:

```

deptno	ename	sal	nums
10	JACCKA	5000.0	1
10	KING	5000.0	2
10	CLARK	2450.0	3
10	WELAN	2450.0	4
10	TEBAGE	1300.0	5
10	MILLER	1300.0	6
20	SCOTT	3000.0	1
20	FORD	3000.0	2
20	JONES	2975.0	3
20	ADAMS	1100.0	4
20	SMITH	800.0	5
30	BLAKE	2850.0	1
30	ALLEN	1600.0	2
30	TURNER	1500.0	3
30	MARTIN	1250.0	4
30	WARD	1250.0	5
30	JAMES	950.0	6

1.5.7.4.16 CLUSTER_SAMPLE

Command syntax:

```
boolean cluster_sample(bigint x[, bigint y]) over(partition by col1[, col2..])
```

Purpose: It is used to conduct cluster sampling.

Description:

- **x: bigint type. $x \geq 1$.** If the parameter y is specified, x indicates that a window is divided into x parts. Otherwise, x indicates that x rows of records are extracted from a window (that is, the returned value is true if there are x rows). If x is NULL, NULL is returned.
- **y: a constant of the bigint type. $y \geq 1$, $y \leq x$.** This parameter extracts y records from x parts into which a window is divided (that is, the returned value is true if y records exist). If y is NULL, NULL is returned.
- **partition by col1[, col2]:** specifies the partitions used in the computation.

Returned value: boolean type.

Example:

The test_tbl table has two columns: key and value. The key column stores the group name of each value. The group names are groupa and groupb. The value column stores the values. The table structure is like this:

key	value
groupa	-1.34764165478145
groupa	0.740212609046718
groupa	0.167537127858695
groupa	0.630314566185241
GroupA	0.0112401388646925
groupa	0.199165745875297
groupa	-0.320543343353587
groupa	-0.273930924365012
groupa	0.386177958942063
groupa	-1.09209976687047
groupb	-1.10847690938643
groupb	-0.725703978381499
groupb	1.05064697475759
groupb	0.135751224393789
groupb	2.13313102040396
groupb	-1.11828960785008
groupb	-0.849235511508911
groupb	1.27913806620453
groupb	-0.330817716670401
groupb	-0.300156896191195
groupb	2.4704244205196
groupb	-1.28051882084434

Run the following SQL statement to take a sample of 10% of the values in each group:

```
select key, value from (select key, value, cluster_sample(10, 1) over(
partition by key) as flag from tbl) sub where flag = true;
-- Returned result:
```

key	value
-----	-------

```
| groupa | -0.273930924365012 |
| groupb | -1.11828960785008 |
+-----+
```

1.5.7.4.17 NTILE

Function declaration:

```
BIGINT ntile(BIGINT n) over(partition by col1[, col2...] [order by col1
[asc|desc] [, col2[asc|desc]...]] [windowing_clause]))
```

Purpose: It is used to split grouped data into *n* slices and return the current slice number. If the slice is uneven, the distribution of the first slice is increased.

Description:

n: BIGINT type.

Returned value: BIGINT type.

Example:

Table emp has the following data:

```
| empno | ename | job | mgr | hiredate | sal | comm | deptno |
7369, SMITH, CLERK, 7902, 1980-12-17 00:00:00, 800, , 20
7499, ALLEN, SALESMAN, 7698, 1981-02-20 00:00:00, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 1981-02-22 00:00:00, 1250, 500, 30
7566, JONES, MANAGER, 7839, 1981-04-02 00:00:00, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 1981-09-28 00:00:00, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 1981-05-01 00:00:00, 2850, , 30
7782, CLARK, MANAGER, 7839, 1981-06-09 00:00:00, 2450, , 10
7788, SCOTT, ANALYST, 7566, 1987-04-19 00:00:00, 3000, , 20
7839, KING, PRESIDENT, , 1981-11-17 00:00:00, 5000, , 10
7844, TURNER, SALESMAN, 7698, 1981-09-08 00:00:00, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 1987-05-23 00:00:00, 1100, , 20
7900, JAMES, CLERK, 7698, 1981-12-03 00:00:00, 950, , 30
7902, FORD, ANALYST, 7566, 1981-12-03 00:00:00, 3000, , 20
7934, MILLER, CLERK, 7782, 1982-01-23 00:00:00, 1300, , 10
7948, JACCKA, CLERK, 7782, 1981-04-12 00:00:00, 5000, , 10
7956, WELAN, CLERK, 7649, 1982-07-20 00:00:00, 2450, , 10
7956, TEBAGE, CLERK, 7748, 1982-12-30 00:00:00, 1300, , 10
```

Group all employees by department, sort each group in descending order by salary, and then obtain sequence numbers of employees in each group.

```
-- Execute the following statement:
select deptno, ename, sal, NTILE(3) OVER(PARTITION BY deptno ORDER BY
sal desc) AS nt3 from emp;
-- Returned result:
```

```
+-----+-----+-----+-----+
| deptno | ename | sal | nt3 |
+-----+-----+-----+-----+
| 10 | JACCKA | 5000.0 | 1 |
| 10 | KING | 5000.0 | 1 |
| 10 | WELAN | 2450.0 | 2 |
| 10 | CLARK | 2450.0 | 2 |
```

10	TEBAGE	1300.0	3
10	MILLER	1300.0	3
20	SCOTT	3000.0	1
20	FORD	3000.0	1
20	JONES	2975.0	2
20	ADAMS	1100.0	2
20	SMITH	800.0	3
30	BLAKE	2850.0	1
30	ALLEN	1600.0	1
30	TURNER	1500.0	2
30	MARTIN	1250.0	2
30	WARD	1250.0	3
30	JAMES	950.0	3

1.5.7.4.18 NTH_VALUE

Function declaration:

```
nth_value(expr, bigint n [, boolean skipNulls]) over(partition by col1
[, col2...] order by col1 [asc|desc][, col2[asc|desc]...])
```

Purpose: It is used to return the nth value in partitions used in the computation.

Description:

- **expr:** required. Any type.
- **n:** returns the nth value. It starts from 1 and is of the BIGINT type.
- **skipNulls:** specifies whether to ignore the rows whose values are NULL. This parameter is of the BOOLEAN type. The default value is false.

Returned value: the nth value in partitions used in the computation.



Note:

If skipNulls is set to true, the nth non-NULL value is returned. If the nth non-NULL value does not exist, NULL is returned.

Example:

```
select a, nth_value(a + 1, 1) over (partition by a order by a) from
values (3), (1), (2) as t(a);
-- If n is 1, NTH_VALUE is equivalent to FIRST_VALUE.
-- Returned results:
-- 1      2
-- 2      3
```

-- 3 4

1.5.7.4.19 CUME_DIST

Function declaration:

```
cume_dist() over(partition by col1[, col2...] order by col1 [asc|desc]
[, col2[asc|desc]...])
```

Purpose: It is used to return the cumulative distribution. The cumulative distribution is the ratio between the number of rows whose values are less than or equal to the current value of the group and the total number of rows in the group.

Description: None.



Note:

The order by column specifies values to be compared.

Returned value: the ratio of the number of rows whose values are equal to or less than the current value in the group to the total number of rows in the group.

Example:

Table emp has the following data:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	,20	
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	,300,30	
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	,500,30	
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	,20	
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	,1400,30	
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	,30	
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	,10	
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	,20	
7839	KING	PRESIDENT	,	1981-11-17 00:00:00	5000	,10	
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	,0,30	
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	,20	
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	,30	
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	,20	
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	,10	
7948	JACCKA	CLERK	7782	1981-04-12 00:00:00	5000	,10	
7956	WELAN	CLERK	7649	1982-07-20 00:00:00	2450	,10	
7956	TEBAGE	CLERK	7748	1982-12-30 00:00:00	1300	,10	

Group all employees by department, and then obtain the cumulative distribution of salary for each group.

```
SELECT deptno
      , ename
      , sal
      , concat(round(cume_dist() OVER(PARTITION BY deptno ORDER BY sal desc
)*100,2),'%') as cume_dist
```

```
FROM emp;
```

Returned result is as follows.

Table 1-20: Returned result

deptno	ename	sal	cume_dist
10	JACCKA	5000.0	33.33%
10	KING	5000.0	33.33%
10	CLARK	2450.0	66.67%
10	WELAN	2450.0	66.67%
10	TEBAGE	1300.0	100.0%
10	MILLER	1300.0	100.0%
20	SCOTT	3000.0	40.0%
20	FORD	3000.0	40.0%
20	JONES	2975.0	60.0%
20	ADAMS	1100.0	80.0%
20	SMITH	800.0	100.0%
30	BLAKE	2850.0	16.67%
30	ALLEN	1600.0	33.33%
30	TURNER	1500.0	50.0%
30	MARTIN	1250.0	83.33%
30	WARD	1250.0	83.33%
30	JAMES	950.0	100.0%

1.5.7.4.20 FIRST_VALUE

Function declaration:

```
first_value(expr) over(partition by col1[, col2...] order by col1 [asc|  
desc][, col2[asc|desc]...])
```

Purpose: It is used to sort partitions and return the first value in the range from the beginning to the current row.

Description:

expr: required. Any type.

Returned value: the first expr value in partitions used in the computation.

Example:

Table emp has the following data:

```
| empno | ename | job | mgr | hiredate | sal | comm | deptno |
7369, SMITH, CLERK, 7902, 1980-12-17 00:00:00, 800, , 20
7499, ALLEN, SALESMAN, 7698, 1981-02-20 00:00:00, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 1981-02-22 00:00:00, 1250, 500, 30
7566, JONES, MANAGER, 7839, 1981-04-02 00:00:00, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 1981-09-28 00:00:00, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 1981-05-01 00:00:00, 2850, , 30
7782, CLARK, MANAGER, 7839, 1981-06-09 00:00:00, 2450, , 10
7788, SCOTT, ANALYST, 7566, 1987-04-19 00:00:00, 3000, , 20
7839, KING, PRESIDENT, , 1981-11-17 00:00:00, 5000, , 10
7844, TURNER, SALESMAN, 7698, 1981-09-08 00:00:00, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 1987-05-23 00:00:00, 1100, , 20
7900, JAMES, CLERK, 7698, 1981-12-03 00:00:00, 950, , 30
7902, FORD, ANALYST, 7566, 1981-12-03 00:00:00, 3000, , 20
7934, MILLER, CLERK, 7782, 1982-01-23 00:00:00, 1300, , 10
7948, JACCKA, CLERK, 7782, 1981-04-12 00:00:00, 5000, , 10
7956, WELAN, CLERK, 7649, 1982-07-20 00:00:00, 2450, , 10
7956, TEBAGE, CLERK, 7748, 1982-12-30 00:00:00, 1300, , 10
```

Group all employees by department, sort each group in descending order by salary, and then obtain the name of the first employee in each group.

```
SELECT deptno
      , ename
      , sal
      , FIRST_VALUE(ename) OVER(PARTITION BY deptno ORDER BY sal
desc) AS first1-- Obtain the name of the first employee in each group
after descending sorting by salary.
FROM emp;
```

Returned result is as follows.

Table 1-21: Returned result

deptno	ename	sal	first1
10	JACCKA	5000.0	JACCKA
10	KING	5000.0	JACCKA
10	CLARK	2450.0	JACCKA
10	WELAN	2450.0	JACCKA
10	TEBAGE	1300.0	JACCKA
10	MILLER	1300.0	JACCKA
20	SCOTT	3000.0	SCOTT
20	FORD	3000.0	SCOTT

deptno	ename	sal	first1
20	JONES	2975.0	SCOTT
20	ADAMS	1100.0	SCOTT
20	SMITH	800.0	SCOTT
30	BLAKE	2850.0	BLAKE
30	ALLEN	1600.0	BLAKE
30	TURNER	1500.0	BLAKE
30	MARTIN	1250.0	BLAKE
30	WARD	1250.0	BLAKE
30	JAMES	950.0	BLAKE

1.5.7.4.21 LAST_VALUE

Function declaration:

```
last_value(expr) over(partition by col1[, col2...] order by col1 [asc|desc][, col2[asc|desc]...])
```

Purpose: It is used to sort partitions and return the last value in the range from the beginning to the current row.

Description:

expr: required. Any type.

Returned value: the last expr value in partitions used in the computation.

Example:

Table emp has the following data:

```
| empno | ename | job | mgr | hiredate | sal | comm | deptno |
7369, SMITH, CLERK, 7902, 1980-12-17 00:00:00, 800, , 20
7499, ALLEN, SALESMAN, 7698, 1981-02-20 00:00:00, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 1981-02-22 00:00:00, 1250, 500, 30
7566, JONES, MANAGER, 7839, 1981-04-02 00:00:00, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 1981-09-28 00:00:00, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 1981-05-01 00:00:00, 2850, , 30
7782, CLARK, MANAGER, 7839, 1981-06-09 00:00:00, 2450, , 10
7788, SCOTT, ANALYST, 7566, 1987-04-19 00:00:00, 3000, , 20
7839, KING, PRESIDENT, , 1981-11-17 00:00:00, 5000, , 10
7844, TURNER, SALESMAN, 7698, 1981-09-08 00:00:00, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 1987-05-23 00:00:00, 1100, , 20
7900, JAMES, CLERK, 7698, 1981-12-03 00:00:00, 950, , 30
7902, FORD, ANALYST, 7566, 1981-12-03 00:00:00, 3000, , 20
7934, MILLER, CLERK, 7782, 1982-01-23 00:00:00, 1300, , 10
7948, JACCKA, CLERK, 7782, 1981-04-12 00:00:00, 5000, , 10
7956, WELAN, CLERK, 7649, 1982-07-20 00:00:00, 2450, , 10
```

```
7956,TEBAGE,CLERK,7748,1982-12-30 00:00:00,1300,,10
```

Group all employees by department, and then obtain the name of the last employee in each group.

```
SELECT deptno
      , ename
      , sal
      , LAST_VALUE(ename) OVER(PARTITION BY deptno  ) AS last1
FROM emp;
```

The returned result is as follows.

Table 1-22: Returned result

deptno	ename	sal	last1
10	TEBAGE	1300.0	WELAN
10	CLARK	2450.0	WELAN
10	KING	5000.0	WELAN
10	MILLER	1300.0	WELAN
10	JACCKA	5000.0	WELAN
10	WELAN	2450.0	WELAN
20	FORD	3000.0	JONES
20	SCOTT	3000.0	JONES
20	SMITH	800.0	JONES
20	ADAMS	1100.0	JONES
20	JONES	2975.0	JONES
30	TURNER	1500.0	BLAKE
30	JAMES	950.0	BLAKE
30	ALLEN	1600.0	BLAKE
30	WARD	1250.0	BLAKE
30	MARTIN	1250.0	BLAKE
30	BLAKE	2850.0	BLAKE

1.5.7.5 Aggregate functions

1.5.7.5.1 Overview

An aggregate function aggregates multiple input records into an output record. The input is mapped many-to-one to the output. An aggregate function can be used with the GROUP BY clause at the same time.

1.5.7.5.2 COUNT

Command syntax:

```
bigint count([distinct|all] value)
```

Purpose: It is used to return the number of records.

Description:

- **distinct|all:** indicates whether duplicate records are cleared in counting. The default value is all, indicating that records are counted. If it is set to distinct, only records with distinct values are counted.
- **value:** any type. When it is NULL, this row is not involved in computation. value can be *. When it is set to count(*), the number of all rows is returned.

Returned value: bigint type.

Example:

In the tbla table, the col1 column is of the bigint type.

```
+-----+
| COL1 |
+-----+
| 1     |
+-----+
| 2     |
+-----+
| NULL  |
+-----+
select count(*) from tbla;
-- 3 is returned.
select count(col1) from tbla;
-- The value is 2.
```

Aggregate functions can be used with the GROUP BY statement. For example, table test_src contains two columns: key (string type), and value (double type).

The data in the test_src table:

```
+-----+-----+
| key | value |
```

```

+-----+-----+
| a     | 2.0    |
+-----+-----+
| a     | 4.0    |
+-----+-----+
| b     | 1.0    |
+-----+-----+
| b     | 3.0    |
+-----+-----+

```

```

select key, count(value) as count from test_src group by key;
-- Run the preceding SQL statement. The output is:

```

```

+-----+-----+
| key   | count  |
+-----+-----+
| a     | 2      |
+-----+-----+
| b     | 2      |
+-----+-----+

```

Aggregate functions perform aggregation on values of the same key. The usage of the following aggregate functions is the same as that of this function and is not described in detail in this document.

1.5.7.5.3 AVG

Function declaration:

```

double avg(double value)
decimal avg(decimal value)

```

Purpose: It is used to calculate the average value.

Description:

value: double type or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned. If the value is NULL, this row is not used for calculation. The input cannot be of the boolean type.

Returned value: If the input is of the decimal type, a value of the decimal type is returned. For all other valid input types, a value of the double type is returned.

Example:

In the tbla table, the value column is of the bigint type.

```

+-----+
| value |
+-----+
| 1     |
| 2     |
| NULL  |
+-----+

```

```

select avg(value) as avg from tbla;

```

```
+-----+
|  avg  |
+-----+
|  1.5  |
+-----+
-- The avg result of this column is as follows: (1 + 2) / 2 = 1.5.
```

1.5.7.5.4 MAX

Function declaration:

```
max(value)
```

Purpose: It is used to return the maximum value.

Description:

value: can be any data type. If the column value is NULL, the corresponding row is not involved in the operation. Values of the boolean type are excluded from the computation.

Returned value: The type is the same as that of value.

Example:

In the tbla table, the col1 column is of the bigint type.

```
+-----+
|  col1 |
+-----+
|   1   |
+-----+
|   2   |
+-----+
| NULL  |
+-----+
select max(value) from tbla;
-- 2 is returned.
```

1.5.7.5.5 MIN

Function declaration:

```
MIN(value)
```

Purpose: It is used to return the minimum value.

Description:

value: a column of any data type. If a value in the column is NULL, the corresponding row is not involved in the operation. Boolean types are not allowed in this operation.

Returned value: The type is the same as that of value.

Example:

In the tbla table, the value column is of the bigint type.

```
+-----+  
| value |  
+-----+  
| 1     |  
+-----+  
| 2     |  
+-----+  
| NULL  |  
+-----+  
select min(value) from tbla;  
-- 1 is returned.
```

1.5.7.5.6 MEDIAN

Function declaration:

```
double median(double number)  
decimal median(decimal number)
```

Purpose: It is used to calculate the median.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other input types, an error is returned. If the input is NULL, a failure is returned.

Returned value: double or decimal type.

1.5.7.5.7 STDDEV

Function declaration:

```
double stddev(double number)  
decimal stddev(decimal number)
```

Purpose: It is used to calculate the population standard deviation.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned. If the input value is NULL, a failure is returned.

Returned value: double or decimal type.

1.5.7.5.8 STDDEV_SAMP

Function declaration:

```
double stddev_samp(double number)
decimal stddev_samp(decimal number)
```

Purpose: It is used to calculate the sample standard deviation.

Description:

number: double or decimal type. If the input is of the string or bigint type, it is implicitly converted into a value of the double type before this computation. For all other types of inputs, an error is returned. If the input is NULL, a failure is returned.

Returned value: double or decimal type.

1.5.7.5.9 SUM

Function declaration:

```
sum(value)
```

Purpose: It is used to calculate the sum.

Description:

value: double, decimal, or bigint type. If the input is of the string type, it is implicitly converted into a value of the double type before computation. If a value in the column is NULL, this row is not used for calculation. Values of the boolean type are excluded from calculation.

Returned value: When the input is of the bigint type, a value of the bigint type is returned. When the input is of the double or string type, a value of the double type is returned.

Example:

In the tbla table, the value column is of the bigint type.

```
+-----+
| value|
+-----+
|  1   |
+-----+
|  2   |
+-----+
```

```
| NULL |
+-----+
select sum(value) from tbla;
-- 3 is returned.
```

1.5.7.5.10 WM_CONCAT

Function declaration:

```
string wm_concat(string separator, string str)
```

Purpose: It is used to use the specified separator as the delimiter to link values in a string.

Description:

- **separator:** the delimiter, which is a constant of the string type. If it is of another type or is not a constant, an error is returned.
- **str:** string type. If the input is of the bigint, double, or datetime type, it is implicitly converted to a value of the string type before this computation. For all other input types, an error is returned.

Returned value: string type.



Note:

If test_src in the select wm_concat(',', name) from > test_src; statement is an empty set, NULL is returned.

1.5.7.5.11 PERCENTILE

Function declaration:

```
DOUBLE percentile(BIGINT col, p)
array<double> percentile(BIGINT col, array(p1 [, p2]...))
```

Purpose: It is used to return the pth percentile of the specified column. p must be between 0 and 1.



Notice:

You can only calculate true percentiles for integer values.

Description:

- **col:** BIGINT type.
- **p:** must be between 0 and 1.

Example:

Column c1 in table test has the following data:

c1
8
9
10
11

Calculate the pth percentile of column c1 in table test.

```
-- Execute the following statement:
select percentile(c1,0),percentile(c1,0.3),percentile(c1,0.5),
percentile(c1,1) from test;
-- Returned result:
+-----+-----+-----+-----+
| _c0      | _c1      | _c2      | _c3      |
+-----+-----+-----+-----+
| 8.0      | 8.9      | 9.5      | 11.0     |
+-----+-----+-----+-----+
-- Execute the following statement:
select percentile(c1,array(0,0.3,0.5,1))from test;
-- Returned result:
+-----+
| _c0 |
+-----+
| [8, 8.9, 9.5, 11] |
+-----+
```

1.5.7.5.12 Additional aggregate functions

MaxCompute 2.0 provides additional aggregate functions. You must add the following SET statement before SQL statements contained in the aggregate functions:

```
set odps.sql.type.system.odps2=true;
```

**Note:**

You must submit and execute the SET statement and the SQL statements of the new functions simultaneously.

The aggregate functions described in subsequent topics are new in MaxCompute 2.0.

1.5.7.5.13 COLLECT_LIST

Command syntax:

```
ARRAY collect_list(col)
```

Purpose: It is used to convert the values on the col column into an array.

Description:

col: a table column of any data type.

Returned value: array type.

1.5.7.5.14 COLLECT_SET

Command syntax:

```
ARRAY collect_set(col)
```

Purpose: It is used to convert the values on the col column with duplicates removed into an array.

Description:

col: a table column of any data type.

Returned value: array type.

1.5.7.5.15 VARIANCE/VAR_POP

Function declaration:

```
DOUBLE variance(col)  
DOUBLE var_pop(col)
```

Purpose: It is used to calculate the variance of the specified numeric column.

Description:

col: numeric type column. NULL is returned for other types.

Returned value: DOUBLE type.

Example:

Column c1 in table test has the following data:

```
+-----+  
| c1    |  
+-----+  
| 8     |  
| 9     |  
+-----+
```


10
11

Calculate the variance of column c1 in table test.

```
-- Execute the following statement:
select variance(c1) from test;
-- or
select var_pop(c1) from test;
-- Returned result:
+-----+
| _c0    |
+-----+
| 1.25   |
+-----+
```

1.5.7.5.16 VAR_SAMP

Function declaration:

```
DOUBLE var_samp(col)
```

Purpose: It is used to calculate the sample variance of the specified numeric column.

Description:

col: numeric type column. NULL is returned for other types.

Returned value: DOUBLE type.

Example:

Column c1 in table test has the following data:

c1
8
9
10
11

Calculate the variance of column c1 in table test.

```
-- Execute the following statement:
select var_samp(c1) from test;
-- Returned result:
+-----+
| _c0    |
+-----+
| 1.6666666666666667 |
+-----+
```

```
+-----+
```

1.5.7.5.17 COVAR_POP

Function declaration:

```
DOUBLE covar_pop(col1, col2)
```

Purpose: It is used to calculate the population covariance of two specified numeric columns.

Description:

col1 and col2: numeric type columns. NULL is returned for other types.

Example:

Columns c1 and c2 in table test have the following data:

c1	c2
3	2
14	5
50	14
26	75

Calculate the population covariance of columns c1 and c2.

```
-- Execute the following statement:
select covar_pop(c1,c2) from test;
-- Returned result:
+-----+
| _c0    |
+-----+
| 123.49999999999997 |
+-----+
```

1.5.7.5.18 COVAR_SAMP

Function declaration:

```
DOUBLE covar_samp(col1, col2)
```

Purpose: It is used to calculate the sample covariance of two specified numeric columns.

Description:

col1 and col2: numeric type columns. NULL is returned for other types.

Example:

Columns c1 and c2 in table test have the following data:

c1	c2
3	2
14	5
50	14
26	75

Calculate the sample covariance of columns c1 and c2.

```
-- Execute the following statement:
select covar_samp(c1,c2) from test;
-- Returned result:
+-----+
| _c0    |
+-----+
| 164.66666666666663 |
+-----+
```

1.5.7.6 Other functions

1.5.7.6.1 ARRAY

Function declaration:

```
array(value1,value2, ...)
```

Purpose: It is used to create an array by using input values.

Description:

value: any type. All the values must be of the same type.

Returned value: ARRAY type.

Example:

```
select array(123,456,789) from dual;
-- Returned result:
[123, 456, 789]
```

1.5.7.6.2 ARRAY_CONTAINS

Function declaration:

```
array_contains(ARRAY<T> a, value v)
```

Purpose: It is used to check whether array a contains value v.

Description:

- **a**: array type.
- **v**: The given value **v** must be of the same type as the data in the array.

Returned value: boolean type.

Example:

```
select array_contains(array('a','b'), 'a') from dual;  
-- True is returned.  
select array_contains(array(456,789),123) from dual;  
-- False is returned.
```

1.5.7.6.3 CAST

Command syntax:

```
cast(expr as <type>)
```

Purpose: It is used to convert an expression of one data type to another. For example, `cast('1' as bigint)` converts 1 of the string type to the integer type. If the conversion fails, an error is returned.



Note:

- `cast(double as bigint)` converts a value of the double type into a value of the bigint type.
- `cast(string as bigint)` converts a value of the string type into a value of the bigint type. If the string is composed of numerals expressed in integer form, it is directly converted into a value of the bigint type. If the string is comprised of numerals expressed in the 'float' or 'exponent' form, it is converted to 'double' type first and then to 'bigint' type.
- For `cast(string as datetime)` or `cast(datetime as > string)`, the datetime format is `yyyy-mm-dd hh:mi:ss` by default.

1.5.7.6.4 COALESCE

Command syntax:

```
coalesce(expr1, expr2, ...)
```

Purpose: It is used to return the first non-NULL value in the list. If all values in the list are NULL, NULL is returned.

Description:

expr: a value to be tested. All these values must be of the same type or be NULL. Otherwise, an error is returned.

Returned value: The type is the same as that of the input.



Note:

At least one parameter is provided. Otherwise, an error is returned.

1.5.7.6.5 DECODE

Function declaration:

```
decode(expression, search, result[, search, result]...[, default])
```

Purpose: It is used to implement the if-then-else conditional branching feature.

Description:

- **expression:** expression to be compared.
- **search:** search string to be compared with the expression.
- **result:** the value returned when the value of search matches the expression.
- **default:** optional. If no search string matches the expression, the default value is returned. If it is not specified, NULL is returned.

Returned value: The matched search is returned. If there are no matches, the default value is returned. If default is not specified, NULL is returned.



Note:

- At least three parameters are specified.
- All results must share the same type or be NULL. Inconsistent data types will cause an error. All values of search and expression must be of the same type. Otherwise, an error is returned.
- If the search option in decode has repeated records and matches the expression, the first search value is returned.

Example:

```
select decode(customer_id,  
1, 'Taobao',  
2, 'Alipay',  
3, 'Aliyun',  
NULL, 'N/A',
```

```
'Others') as result from sale_detail;
```

The preceding DECODE function implements the feature in the following if-then-else statement:

```
if customer_id = 1 then result := 'Taobao';  
elsif customer_id = 2 then result := 'Alipay';  
elsif customer_id = 3 then result := 'Aliyun';  
...  
else  
result := 'Others';  
end if;
```

**Notice:**

The MaxCompute SQL statement returns NULL when calculating NULL = NULL. However, in the DECODE function, values of NULL and NULL are equal. In the preceding example, when the value of customer_id is NULL, the DECODE function returns N/A.

1.5.7.6.6 EXPLODE

Function declaration:

```
explode (var)
```

Purpose: It is used to convert one row of data into multiple rows of UDTF. If var is of the array type, the array stored in the column is converted into multiple rows. If var is of the map type, each key-value pair of the map stored in the column is converted into a row with two columns, with one column for the key and the other for the value.

Description:

var: array < T > type or map < K,V > type.

Returned value: transposed rows.

**Note:**

Limits on the use of UDTFs:

- Only one UDTF is allowed in a SELECT statement, and other columns are not allowed.
- One select can only have one UDTF and no other columns can appear.

Example:

```
explode(array(null, 'a', 'b', 'c')) col
```

1.5.7.6.7 GET_IDCARD_AGE

Function declaration:

```
get_idcard_age(idcardno)
```

Purpose: It is used to return the current age based on the ID card number. The current age is the current year minus the birth year on the ID card.

Description:

idcardno: string type, ID number of 15-digit or 18-digit. During the calculation, the validity of the ID card is verified based on the province code and the last check code. If the verification fails, NULL is returned.

Returned value: bigint type. If the input is NULL, NULL is returned. If the difference of the current year minus the birth year is greater than 100, then NULL is returned.

1.5.7.6.8 GET_IDCARD_BIRTHDAY

Function declaration:

```
get_idcard_birthday(idcardno)
```

Purpose: It is used to return the date of birth based on the ID card number.

Description:

idcardno: string type, a 15-digit or 18-digit ID card number. During computation, the validity of the ID card is verified based on the province code and the last check code. If the verification fails, NULL is returned.

Returned value: datetime type. If the input is NULL, NULL is returned.

1.5.7.6.9 GET_IDCARD_SEX

Function declaration:

```
get_idcard_sex(idcardno)
```

Purpose: It is used to return the gender based on the ID card number. The returned value is M (male) or F (female).

Description:

idcardno: string type, a 15-digit or 18-digit ID card number. During computation, the validity of the ID card is verified based on the province code and the last check code. If the verification fails, NULL is returned.

Returned value: string type. If the input is NULL, NULL is returned.

1.5.7.6.10 GREATEST

Function declaration:

```
greatest(var1, var2, ...)
```

Purpose: It is used to return the maximum value among the input values.

Description:

var: bigint, double, datetime, or string type. If all values are NULL, NULL is returned.

Returned value:

- The greatest value in input parameter. If the implicit conversion is not needed, return type is the same as input parameter type.
- NULL is interpreted as the minimum value.
- If the input parameters are of different types, values of the double, bigint, and string types are converted into values of the double type for comparison, and values of the string and datetime types are converted into values of the datetime type for comparison. Implicit conversion of other types is not allowed.

1.5.7.6.11 INDEX

Function declaration:

```
index(var1[var2])
```

Purpose: It is used to return the specified element in a given array, or return the value of the specified key in a given map.

Description:

- **var1:** array < T > type or map < K,V > type.
- **var2:** If var1 is of the array < T > type, var2 must be the bigint type must be larger or equal to 0. If var1 is of the map < K,V > type, var2 is of the K type.

Returned value:

- If var1 is of the array < T > type, a value of the T type is returned. If var2 is out of range of array < T > elements, NULL is returned.
- If var1 is of the map < K,V > type, a value of the V type is returned. If no key is var2 in map < K,V >, NULL is returned.

Example:

If var1 is an array, run the following SQL statement:

```
select array('a','b','c')[2] from dual;
-- Returned result:
+-----+
| _c0 |
+-----+
| c   |
+-----+
```

If var1 is of the map type, run the following SQL statement:

```
select str_to_map("test1=1,test2=2")["test1"] from dual;
-- Returned result:
+-----+
| _c0 |
+-----+
| 1   |
+-----+
```



Notice:

- To use the SQL statement, remove the index and run var1[var2] directly. Otherwise, a syntax error is returned.
- If Var1 is NULL, NULL is returned.

1.5.7.6.12 MAX_PT

Function declaration:

```
max_pt(table_full_name)
```

Purpose: For partitioned tables, it is used to return the maximum values in the first-level partitions that have data files and sort the values in alphabetic order.

Description:

table_full_name: string type. It specifies a table name (project name required, for example, prj.src). You must have the read permission on the table.

Returned value: maximum value in the primary partition.

Example:

Partitioned table tbl has the following partitions with data files: pt='20170901' and pt='20170902'. In the following statement, the returned value of max_pt is '20170902'. The MaxCompute SQL statement reads data from the '20120902' partition.

```
select * from tbl where pt=max_pt('myproject.tbl');
```

**Note:**

If a new partition is added by using alter table, but there is no data file in this partition, then this partition is not returned.

1.5.7.6.13 ORDINAL

Function declaration:

```
ordinal(bigint nth, var1, var2, ...)
```

Purpose: It is used to sort the input variables in ascending order, and return the specified nth value.

Description:

- **nth:** bigint type. It specifies the position at which the value is to be returned. If it is NULL, NULL is returned.
- **var:** bigint, double, datetime, or string type.

Returned value:

- The value in nth bit. If the implicit conversion is not needed, return type is the same as input parameter type.
- If type conversion is performed, values of the double, bigint, and string types are converted into values of the double type. Values of the string and datetime types are converted into values of the datetime type. Implicit conversion of other types is not allowed.
- NULL is the least value.

Example:

```
ordinal(3, 1, 3, 2, 5, 2, 4, 6) = 2
```

1.5.7.6.14 LEAST

Function declaration:

```
least(var1, var2, ...)
```

Purpose: It is used to returns the minimum value among the input values.

Description:

var: bigint, double, datetime, or string type. If all values are NULL, NULL is returned.

Returned value:

- The least value in input parameter; If the implicit conversion is not needed, return type is the same as input parameter type.
- If type conversion is performed, values of the double, bigint, and string types are converted into values of the double type. Values of the string and datetime types are converted into values of the datetime type. Implicit conversion of other types is not allowed.
- NULL is interpreted as the minimum value.

1.5.7.6.15 SIZE

Function declaration:

```
size(map<K, V>)  
size(array<T>)
```

Purpose: size(map) is used to return the number of key-value pairs in the given map, and size(array) is used to return the number of elements in the given array.

Description:

- map: map type.
- array: array type.

Returned value: int type.

Example:

```
select size(map('a',123,'b',456)) from dual;  
-- 2 is returned.
```

```
select size(map('a',123,'b',456,'c',789)) from dual;  
-- 3 is returned.  
select size(array('a','b')) from dual;  
-- 2 is returned.  
select size(array(123,456,789)) from dual;  
-- 3 is returned.
```

1.5.7.6.16 SPLIT

Function declaration:

```
split(str, pat)
```

Purpose: It is used to split a string using the specified separator.

Description:

- **str:** string type. The string to be separated.
- **pat:** string type. It indicates the separator and supports regular expressions.

Returned value: array <string>. The returned array contains elements extracted from the string based on the specified separator.

Example:

```
select split("a,b,c",",") from dual;  
-- Returned result:  
+-----+  
| _c0 |  
+-----+  
| [a, b, c] |  
+-----+
```

1.5.7.6.17 STR_TO_MAP

Function declaration:

```
str_to_map(text [, delimiter1 [, delimiter2]])
```

Purpose: It is used to divide 'text' into K-V pairs with 'delimiter1', and to separate each K-V pair with 'delimiter2'.

Description:

ext: string type. It indicates the string to be separated.

delimiter1: string type. It is the delimiter. If it is not specified, the default value ',' is used.

delimiter2: string type. It is the delimiter. If it is not specified, the default value '=' is used.

Returned value: map < string, string >. The elements are the K-V results of the separation of 'text' by the strings 'delimiter1' and 'delimiter2'.

Example:

```
select str_to_map("test1=1,test2=2") from dual;
-- Returned result:
+-----+
| a      |
+-----+
| {Test1: 1, Test2: 2} |
```

1.5.7.6.18 UNIQUE_ID

Function declaration:

```
STRING UNIQUE_ID()
```

Purpose: It is used to return a random but unique ID, for example, 29347a88-1e57-41ae-bb68-a9edbdd94212_1. This function runs more efficiently than UUID.

1.5.7.6.19 UUID

Function declaration:

```
string uuid()
```

Purpose: It returns a random ID, for example, 29347a88-1e57-41ae-bb68-a9edbdd94212.

1.5.7.6.20 SAMPLE

Function declaration:

```
boolean sample(x, y, column_name)
```

Purpose: It is used to sample all values read from the specified column based on the given settings, and filters out the rows that do not meet the sampling condition.

Description:

- **x, y: bigint type.** It indicates that data is hashed to x portions and the yth portion is taken. y can be omitted. If y is omitted, the first portion is taken and column_name must also be omitted. x and y are constants of the integer type and are greater than 0. If they are of another type or if they are less than or equal to 0

, an error is returned. If y is greater than x, an error is returned. If either x or y is NULL, NULL is returned.

- **column_name**: target column of sampling. **column_name** can be omitted. If **column_name** is omitted, random sampling is performed based on values of x and y. It can be of any type, and the column value can be NULL. No implicit conversion is performed. If **column_name** is the constant NULL, an error is reported.

Returned value: boolean type.



Note:

To avoid data skew resulting from the NULL value, a uniform hash of x is made for a value of NULL in **column_name**. If **column_name** is not added, the output is not necessarily uniform since the data size is smaller. So **column_name** is suggested to be added to get better output.

Example:

Table **tbla** contains a column named **cola**.

```
select * from tbla where sample (4, 1 , cola) = true;
-- The values are hashed to four portions based on cola, and the first
   portion is used.
select * from tbla where sample (4, 2) = true;
-- The values in each row are randomly hashed to four portions, and
   the second portion is used.
```

1.5.7.6.21 CASE WHEN expression

MaxCompute provides the following two kinds of CASE WHEN syntax formats:

```
case value
when (_condition1) then result1
when (_condition2) then result2
...
else resultn
end

case
when (_condition1) then result1
when (_condition2) then result2
when (_condition3) then result3
...
else resultn
```

```
end
```

CASE WHEN flexibly returns different values based on the calculation result of the expression. Alibaba Cloud StreamCompute supports two types of CASE WHEN expressions:

```
select
case
when shop_name is null then 'default_region'
when shop_name like 'hang%' then 'zj_region'
end as region
From sale_detail;
```



Note:

- If there are values of only the bigint and double type in the results, the results are converted into values of the double type.
- If there is a value of the string type in the results, the results are all converted into values of the string type. If the result of a type cannot be converted (for example, boolean type), an error is returned.
- Conversion between other types is not allowed.

1.5.7.6.22 IF

Function declaration:

```
if(testCondition, valueTrue, valueFalseOrNull)
```

Purpose: It is used to determine whether 'testCondition' is true. If it is true, valueTrue is returned. If it is not true, valueFalseOrNull is returned.

Description:

testCondition: boolean type. The expression to be determined true or not.

valueTrue: the value returned when expression 'testCondition' is true.

valueFalseOrNull: the value returned when expression 'testCondition' is false. It can be set to NULL.

Returned value: The type is the same as that of valueTrue or valueFalseOrNull.

Example:

```
select if(1=2,100,200) from dual;
-- Returned result:
+-----+
| _c0    |
+-----+
```

```
| 200 |  
+-----+
```

1.5.7.6.23 Additional functions

MaxCompute 2.0 provides additional functions.

The functions described in the following topics are new in this version.

1.5.7.6.24 MAP

Function declaration:

```
map(K key1, V value1, K key2, V value2, ...)
```

Purpose: It is used to create a map with the given K-V pairs.

Description:

key/value: The types of all keys are the same and must be of one of the basic types.

The types of all values are the same and can be of any type.

Returned value: map type.

Example:

```
select map('a',123,'b',456) from dual;  
-- Returned result:  
{a:123, b:456}
```

1.5.7.6.25 MAP_KEYS

Function declaration:

```
map_keys(map<K, V> )
```

Purpose: It is used to return all keys in the map parameter as an array.

Description:

map: data of the map type.

Returned value: array type. If the input is NULL, NULL is returned.

Example:

```
select map_keys(map('a',123,'b',456)) from dual;  
-- Returned result:
```



```
[a, b]
```

1.5.7.6.26 MAP_VALUES

Function declaration:

```
map_values(map<K, V>)
```

Purpose: It is used to return all values in the map parameter as an array.

Description:

map: map type.

Returned value: array type. If the input is NULL, NULL is returned.

Example:

```
select map_keys(map('a',123,'b',456)) from dual;  
-- Returned result:  
[123, 456]
```

1.5.7.6.27 SORT_ARRAY

Function declaration:

```
sort_array(ARRAY<T>)
```

Purpose: It is used to sort a given array.

Description:

ARRAY: array type. The data in the array is of any type.

Returned value: array type.

Example:

```
select sort_array(array('a','c','f','b')),sort_array(array(4,5,7,2,5,8  
)),sort_array(array('You','Me','He')) from dual;  
-- Returned result:  
[a, b, c, f] [2, 4, 5, 5, 7, 8] [him, you, me]
```

1.5.7.6.28 POSEXplode

Command syntax:

```
posexplode(ARRAY<T>)
```

Purpose: It is used to explode the given array. Each value is given a row and each row has two columns corresponding to the subscript (starting from 0) and the array element.

Description:

ARRAY: array type. Data in the array can be of any type.

Returned value: table generation function.

Example:

```
select posexplode(array('a','c','f','b')) from dual;
-- Returned result:
```

pos	val
0	a
1	c
2	f
3	b

1.5.7.6.29 STRUCT

Function declaration:

```
struct(value1,value2, ...)
```

Purpose: It is used to create a struct using a given value list.

Description:

value: any type.

Returned value: struct type. The field names of the created struct are col1, col2, and so on.

Example:

```
select struct('a',123,'ture',56.90) from dual;
-- Returned result:
{col1:a, col2:123, col3:true, col4:56.9}
```

1.5.7.6.30 NAMED_STRUCT

Function declaration:

```
named_struct(string name1, T1 value1, string name2, T2 value2, ...)
```

Purpose: It is used to create a struct using a given name-value list.

Description:

- **value:** any type.
- **name:** field name of the string type.

Returned value: struct type. The field names of the created struct are name1, name2, and so on.

Example:

```
select named_struct('user_id',10001,'user_name','bob','married','F','weight',63.50) from dual;
-- Returned result:
{user_id:10001, user_name:bob, married:F, weight:63.5}
```

1.5.7.6.31 INLINE

Function declaration:

```
inline(ARRAY<STRUCT<f1:T1, f2:T2, ... >>)
```

Purpose: It is used to expand a struct, with each element corresponding to a row, and each struct element in each row corresponding to a column.

Description:

STRUCT: The values in the array can be of any type.

Returned value: table generation function.

Example:

```
select inline(array(named_struct('user_id',10001,'user_name','bob','married','F','weight',63.50))) from dual;
-- Returned result:
```

user_id	user_name	married	weight
10001	bob	F	63.5

1.5.7.6.32 BETWEEN AND expression

Command syntax:

```
A [NOT] BETWEEN B AND C
```

If A, B, or C is NULL, then the value is NULL. If A is greater than or equal to B, and less than or equal to C, the value is true. Otherwise, the value is false.

Example:

The emp table contains the following data:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	500	30

```

7566,JONES,MANAGER,7839,1981-04-02 00:00:00,2975,,20
7654,MARTIN,SALESMAN,7698,1981-09-28 00:00:00,1250,1400,30
7698,BLAKE,MANAGER,7839,1981-05-01 00:00:00,2850,,30
7782,CLARK,MANAGER,7839,1981-06-09 00:00:00,2450,,10
7788,SCOTT,ANALYST,7566,1987-04-19 00:00:00,3000,,20
7839,KING,PRESIDENT,,1981-11-17 00:00:00,5000,,10
7844,TURNER,SALESMAN,7698,1981-09-08 00:00:00,1500,0,30
7876,ADAMS,CLERK,7788,1987-05-23 00:00:00,1100,,20
7900,JAMES,CLERK,7698,1981-12-03 00:00:00,950,,30
7902,FORD,ANALYST,7566,1981-12-03 00:00:00,3000,,20
7934,MILLER,CLERK,7782,1982-01-23 00:00:00,1300,,10
7948,JACCKA,CLERK,7782,1981-04-12 00:00:00,5000,,10
7956,WELAN,CLERK,7649,1982-07-20 00:00:00,2450,,10
7956,TEBAGE,CLERK,7748,1982-12-30 00:00:00,1300,,10

```

Run the following command to query data where sal is greater than or equal to 1,000 and less than or equal to 1,500:

```

select * from emp where sal BETWEEN 1000 and 1500;
-- Returned result:
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename | job | mgr | hiredate | sal | comm |
| deptno |      |     |     |          |     |      |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+
| 7521 | WARD | SALESMAN | 7698 | 1981-02-22 00:00:00 | 1250.0 | |
| 500.0 |      | 30 |     |          |     |      |
| 7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 00:00:00 | 1250.0 |
| 1400.0 |      | 30 |     |          |     |      |
| 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 00:00:00 | 1500.0 |
| 0.0 |      | 30 |     |          |     |      |
| 7876 | ADAMS | CLERK | 7788 | 1987-05-23 00:00:00 | 1100.0 |
NULL | 20 |     |     |          |     |      |
| 7934 | MILLER | CLERK | 7782 | 1982-01-23 00:00:00 | 1300.0 |
NULL | 10 |     |     |          |     |      |
| 7956 | TEBAGE | CLERK | 7748 | 1982-12-30 00:00:00 | 1300.0 |
NULL | 10 |     |     |          |     |      |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

```

1.5.7.6.33 NVL

Function declaration:

```
nvl(T value, T default_value)
```

Purpose: It is used to return default_value if value is NULL and return value otherwise.

Example:

Table t_data has three columns of c1 string, c2 bigint, and c3 datetime, as well as the following data:

```

+-----+-----+-----+
| c1 | c2 | c3 |
+-----+-----+-----+

```

```

+-----+-----+-----+
| NULL | 20      | 2017-11-13 05:00:00 |
| ddd  | 25      | NULL                |
| bbb  | NULL    | 2017-11-12 08:00:00 |
| aaa  | 23      | 2017-11-11 00:00:00 |
+-----+-----+-----+

```

Use the NVL function to output the NULL values in c1 to 00000, the NULL values in c2 to 0, and the NULL values in c3 to "-".

```

-- Execute the following statement:
SELECT nvl(c1,'00000'),nvl(c2,0) nvl(c3,'-') from nvl_test;
-- Returned result:
+-----+-----+-----+
| _c0 | _c1      | _c2 |
+-----+-----+-----+
| bbb | 0        | 2017-11-12 08:00:00 |
| ddd | 25       | -    |
| 00000 | 20      | 2017-11-13 05:00:00 |
| aaa | 23       | 2017-11-11 00:00:00 |
+-----+-----+-----+

```

1.5.8 UDFs

1.5.8.1 Overview

UDF is short for user defined function. MaxCompute provides a variety of built-in functions. You can also create UDFs based on specific computing requirements. You can use UDFs as using common built-in functions. This topic briefs how to use SQL UDFs. For more information about SQL UDFs, see the official documentation on UDFs.

The following table lists the extended UDFs in MaxCompute.

Table 1-23: UDF category

UDF category	Description
UDF	User defined scalar functions are commonly referred to as UDFs. There is a one-to-one mapping between the input and output. Each time a UDF reads a row of data, it writes an output value.
UDTF	User defined table valued functions are commonly referred to as UDTFs. Each time a UDTF is called, it outputs multiple rows of data. UDTFs are the only category that returns multiple fields. A UDF only returns one value each time.

UDF category	Description
UDAF	User defined aggregation functions are commonly referred to as UDAFs. A UDAF aggregates multiple input records into one output record. There is a many-to-one mapping between input and output. A UDAF can be used together with the GROUP BY clause (SQL) at the same time. For more information about the syntax, see aggregation functions.

**Note:**

In general, UDFs refer to all user defined functions: UDFs, UDAFs, and UDTFs. In a narrow sense, UDFs only refer to user defined scalar functions. This term is used interchangeably in this document. You will have to determine the exact meaning based on the context.

1.5.8.2 Types of parameters and returned values

UDFs support the following MaxCompute SQL data types:

- Basic data types: BIGINT, DOUBLE, BOOLEAN, DATETIME, DECIMAL, STRING, TINYINT, SMALLINT, INT, FLOAT, VARCHAR, BINARY, and TIMESTAMP.
- Complex data types: ARRAY, MAP, and STRUCT.

**Note:**

In UDFs, you can define the writable attribute of parameters.

The usage of some basic data types (such as TINYINT, SMALLINT, INT, FLOAT, VARCHAR, BINARY, and TIMESTAMP) in Java UDFs is as follows:

- UDAFs and UDTFs use the @Resolve annotation to obtain signatures. Example: `@Resolve("smallint->varchar(10)")`.
- UDFs reflect and analyze the evaluate() method to obtain signatures. In this case, there is a one-to-one mapping between MaxCompute built-in types and Java types.

To use complex data types (ARRAY, MAP, and STRUCT) in Java UDFs, take the following steps:

- UDTFs use the `@Resolve` annotation to specify signatures. Example: `@Resolve("array<string>,struct<a1:bigint,b1:string>,string->map<string,bigint>,struct<b1:bigint>")`.
- UDFs use the signature of the `evaluate()` method to map the input and output types. For more information, see the mappings between MaxCompute types and Java types. In the preceding example, ARRAY corresponds to `java.util.List`, MAP corresponds to `java.util.Map`, and STRUCT corresponds to `com.aliyun.odps.data.Struct`.
- UDAFs and UDTFs use the `@Resolve` annotation to obtain signatures. Example: `@Resolve("smallint->varchar(10)")`.

**Notice:**

- You can use `type,*` to add any number of parameters. Example: `@resolve("string,*->array<string>")`. Note that you must add a subtype after array.
- The field name and field type of `com.aliyun.odps.data.Struct` cannot be reflected. Therefore, the `@Resolve` annotation is required. If you want to use struct in a UDF, you must add the `@Resolve` annotation to the UDF class. This annotation only affects the overloads of parameters or returned values that contain `com.aliyun.odps.data.Struct`.
- A class supports only one `@Resolve` annotation. A UDF that contains struct can only reload parameters or returned values once.

The following table lists the mapping between MaxCompute and Java data types.

Table 1-24: Data type mapping

MaxCompute type	Java type
TINYINT	<code>java.lang.Byte</code>
SMALLINT	<code>java.lang.Short</code>
INT	<code>java.lang.Integer</code>
BIGINT	<code>java.lang.Long</code>
FLOAT	<code>java.lang.Float</code>
DOUBLE	<code>java.lang.Double</code>
DECIMAL	<code>java.math.BigDecimal</code>

MaxCompute type	Java type
BOOLEAN	java.lang.Boolean
STRING	java.lang.String
VARCHAR	com.aliyun.odps.data.Varchar
BINARY	com.aliyun.odps.data.Binary
DATETIME	java.util.Date
TIMESTAMP	java.sql.Timestamp
ARRAY	java.util.List
MAP	java.util.Map
STRUCT	com.aliyun.odps.data.Struct

**Note:**

- Java data types and the data types of returned values are objects, and must start with a capitalized letter.
- The NULL value in SQL is represented by a NULL reference in Java. The Java primitive type is not allowed because it cannot represent a NULL value in SQL.
- The ARRAY type in MaxCompute corresponds to a list, not an array, in Java.

The following table compares the API features of two languages.

Table 1-25: API feature comparison

Supported language	UDF	UDAF	UDTF	DATETIME type	Read resource file	Read resource table
Python	Yes	Yes	Yes	Yes	Yes	Yes
Java	Yes	Yes	Yes	Yes	Yes	Yes

1.5.8.3 UDFs

A UDF must inherit the `com.aliyun.odps.udf.UDF` class and implement the `EVALUATE` method. The `EVALUATE` method must be a non-static public method. The types of parameters and returned values of the `EVALUATE` method are used as the UDF signatures in SQL. This means that users can implement multiple `EVALUATE`

methods in a UDF. When a UDF is called, the framework matches the correct EVALUATE method based on the parameter type called by the UDF.

Example:

```
package org.alidata.odps.udf.examples;
import com.aliyun.odps.udf.UDF;
public final class Lower extends UDF { public String evaluate(String s
) { if (s == null) { return null; } return s.toLowerCase();
}
}
```



Note:

You can implement `void setup(ExecutionContext ctx)` **and** `void close()` **to implement UDF initialization and termination code, respectively.**

UDFs are used in the same way as built-in functions in MaxCompute SQL. For more information, see [Built-in functions](#).

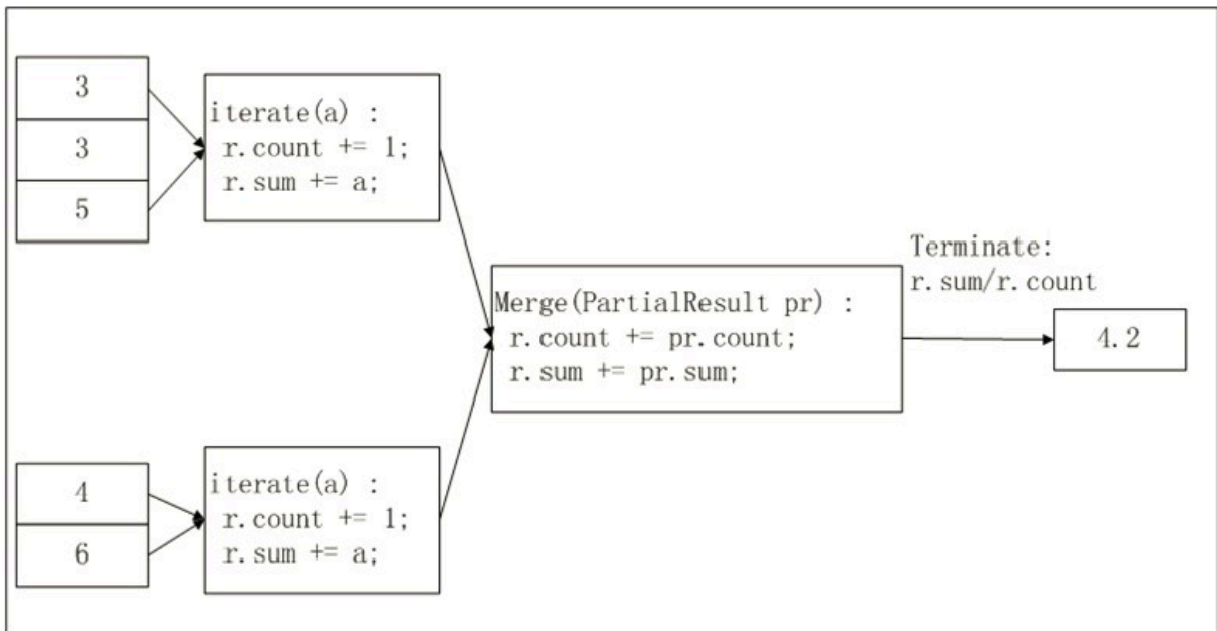
1.5.8.4 UDAFs

To implement a Java UDAF, you must inherit the `com.aliyun.odps.udf.UDAF` class and implement the following APIs:

```
public abstract class Aggregator implements ContextFunction {
    @Override
    public void setup(ExecutionContext ctx) throws UDFException {
    }
    @Override
    public void close() throws UDFException {
    }
    /**
     * Create an aggregate buffer
     * @return Writable - Aggregate buffer
     */
    abstract public Writable newBuffer();
    /**
     * @param buffer - Aggregate buffer
     * @param args - Parameter specified when SQL calls UDAFs
     * @throws UDFException
     */
    abstract public void iterate(Writable buffer, Writable[] args) throws
    UDFException;
    /**
     * generate final result
     * @param buffer
     * @return final result of Object UDAF
     * @throws UDFException
     */
    abstract public Writable terminate(Writable buffer) throws UDFExcepti
    on;
    abstract public void merge(Writable buffer, Writable partial) throws
    UDFException;
}
```

}

The most important APIs are `iterate`, `merge`, and `terminate`. The primary logic of UDAFs relies on the implementation of these three APIs. In addition, you must implement a custom writable buffer. As an example, the following figure briefly illustrates the implementation logic and computational flow of the `avg` (average value) MaxCompute UDAF function.



In the preceding figure, the input data is sliced by a certain size (for description of slicing, see [MapReduce](#)). The size of each slice is suitable for a worker to complete in an appropriate period of time. You need to manually configure the size of the slices.

The UDAF calculation process is divided into two phases:

- **Phase 1:** Each Worker counts the number of data rows and the sum of the data in each slice. The user can regard the counted number and sum as an intermediate result.
- **Phase 2:** The Worker summarizes the information gained from the previous phase within each slice. In the final output, $r.sum / r.count$ is the average of all input data.

The following example shows how to calculate an average by using a UDAF:

```

import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import com.aliyun.odps.io.DoubleWritable;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.udf.Aggregator;
import com.aliyun.odps.udf.UDFException;
  
```

```
import com.aliyun.odps.udf.annotation.Resolve;
@Resolve({"double->double"})
public class AggrAvg extends Aggregator {
    private static class AvgBuffer implements Writable { private double
sum = 0;
private long count = 0;
@Override
public void write(DataOutput out) throws IOException { out.writeDouble
(sum);
out.writeLong(count);
}
@Override
public void readFields(DataInput in) throws IOException { sum = in.
readDouble();
count = in.readLong();
}
}
private DoubleWritable ret = new DoubleWritable();
@Override
public Writable newBuffer() { return new AvgBuffer();
}
@Override
public void iterate(Writable buffer, Writable[] args) throws
UDFException { DoubleWritable arg = (DoubleWritable) args[0];
AvgBuffer buf = (AvgBuffer) buffer; if (arg != null) {
buf.count += 1; buf.sum += arg.get();
}
}
@Override
public Writable terminate(Writable buffer) throws UDFException {
AvgBuffer buf = (AvgBuffer) buffer;
if (buf.count == 0) { ret.set(0);
} else {
ret.set(buf.sum / buf.count);
}
return ret;
}
@Override
public void merge(Writable buffer, Writable partial) throws UDFExcepti
on { AvgBuffer buf = (AvgBuffer) buffer;
AvgBuffer p = (AvgBuffer) partial; buf.sum += p.sum;
buf.count += p.count;
}
}
```

**Notice:**

- The SQL syntax used by UDAFs is the same as that used by common built-in aggregate functions. For more information, see [Aggregate functions](#).
- The way to run UDTFs is the same as that to run UDFs. For more information, see [Run UDFs](#).

1.5.8.5 UDTFs

1.5.8.5.1 Overview

Java UDTFs must inherit the `com.aliyun.odps.udf.UDTF` class. This class requires the implementation of four APIs. The following table lists the definitions of these APIs.

Table 1-26: API definitions

API	Description
<code>public void setup(ExecutionContext ctx) throws UDFException</code>	The initialization method to call the user-defined initialization behavior before a UDTF processes the input data. SETUP is called once first in each worker.
<code>public void process(Object[] args) throws UDFException</code>	This method is called by the framework . Each SQL record calls PROCESS once . The parameters of PROCESS are the specified UDTF input parameters in the SQL statement. The input parameters are passed in as Object[], and the results are output by the FORWARD function . You need to call FORWARD in the PROCESS function to determine the output data.
<code>public void close() throws UDFException</code>	The termination method of UDTF. This method is called by the framework for only once after the last record is processed.
<code>public void forward(Object ...o) throws UDFException</code>	You can call the FORWARD method to output data. Each time FORWARD is called, it outputs one record. The record corresponds to the column specified by the UDTF AS clause in the SQL statement .

UDTF example:

```
package org.alidata.odps.udtf.examples;
import com.aliyun.odps.udf.UDTF;
import com.aliyun.odps.udf.UDTFCollector;
import com.aliyun.odps.udf.annotation.Resolve;
import com.aliyun.odps.udf.UDFException;
```

```
// TODO define input and output types, e.g., "string,string->string,
bigint".
@Resolve({"string,bigint->string,bigint"}) public class MyUDTF extends
UDTF {
@Override public void process(Object[] args) throws UDFException {
String a = (String) args[0];
Long b = (Long) args[1];
for (String t: a.split("\\s+")) { forward(t, b);
}
}
}
```

The preceding example shows how to create a UDTF in MaxCompute. If this UDTF is named `user_udtf`, you can run the following SQL statement to call this UDTF:

```
select user_udtf(col0, col1) as (c0, c1) from my_table;
```

The values in `my_table` `col0` and `col1` are as follows:

col0	col1
A B	1
C D	2

The result of the `SELECT` statement is as follows:

c0	c1
A	1
B	1
C	2
D	2

1.5.8.5.2 UDTF description

Common uses of UDTFs in SQL:

```
select user_udtf(col0, col1) as (c0, c1) from my_table;
select user_udtf(col0, col1) as (c0, c1) from (select * from my_table
distribute by col1 sort by col1) t;
```



Notice:

The following limits apply to the use of UDTF.

- **No other expressions are allowed in a SELECT clause.**

```
select col0, user_udtf(col0, col1) as (c0, c1) from mytable;
```

- **UDTFs cannot be nested.**

```
select user_udtf(mp_udtf(col0,col1)) as (c0,c1)from mytable;
```

UDTF examples

The user can use a UDTF to read MaxCompute resources. The following are examples of reading MaxCompute resources by using UDTFs:

1. **Write UDTF program. The JAR package (udtfexample1.jar) is exported after compilation.**

```
package com.aliyun.odps.examples.udf;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Iterator;
import com.aliyun.odps.udf.ExecutionContext;
import com.aliyun.odps.udf.UDFException;
import com.aliyun.odps.udf.UDTF;
import com.aliyun.odps.udf.annotation.Resolve;
/**
 * project: example_project
 * table: wc_in2
 * partitions: p2=1,p1=2
 * columns: colc,colb
 */
@Resolve({ "string,string->string,bigint,string" }) public class
UDTFResource extends UDTF { ExecutionContext ctx;
long fileResourceLineCount;
long tableResource1RecordCount;
long tableResource2RecordCount;
@Override
public void setup(ExecutionContext ctx) throws UDFException { this.
ctx = ctx;
try {
InputStream in = ctx.readResourceFileAsStream("file_resource.txt");
BufferedReader br = new BufferedReader(new InputStreamReader(in));
String line;
fileResourceLineCount = 0;
while ((line = br.readLine()) != null) { fileResourceLineCount++;
}
br.close();
Iterator<Object[]> iterator = ctx.readResourceTable("table_resource1
").iterator();
tableResource1RecordCount = 0;
while (iterator.hasNext()) { tableResource1RecordCount++; iterator.
next();
}
iterator = ctx.readResourceTable("table_resource2").iterator();
tableResource2RecordCount = 0;
while (iterator.hasNext()) { tableResource2RecordCount++;
iterator.next();
}
}
```

```

    } catch (IOException e) { throw new UDFException(e);
    }
    }
    @Override
    public void process(Object[] args) throws UDFException { String a =
        (String) args[0];
        long b = args[1] == null ? 0 : ((String) args[1]).length();
        forward(a, b, "fileResourceLineCount=" + fileResourceLineCount +
            "|tableResource1RecordCount=" + tableResource1RecordCount + "|"
            + tableResource2RecordCount=" + tableResource2RecordCount);
    }
    }
}

```

2. Add resources to MaxCompute.

```

Add file file_resource.txt;
Add jar udtfexample1.jar;
Add table table_resource1 as table_resource1;
Add table table_resource2 as table_resource2;

```

3. Create UDTF function (mp_udtf) in MaxCompute.

```

create function mp_udtf as com.aliyun.odps.examples.udf.UDTFResource
using 'udtfexample1.jar, file_resource.txt, table_resource1,
table_resource2';

```

4. Create resource tables 'table_resource1' and 'table_resource2' in MaxCompute, and insert the corresponding data.

5. Run this UDTF.

```

select mp_udtf("10","20") as (a, b, fileResourceLineCount) from
table_resource1;
-- Command output:
+-----+-----+-----+
| a | b | fileResourceLineCount |
+-----+-----+-----+
| 10 | 2 | fileResourceLineCount=3|tableResource1RecordCount=0|
tableResource2RecordCount=0 |
| 10 | 2 | fileResourceLineCount=3|tableResource1Record
Count=0|tableResource2RecordCount=0 |
+-----+-----+-----+

```



Note:

You can also use the same method to obtain resources. For more information, see [MapReduce examples](#).

UDTF examples — Complex data types

The code in the following example defines a UDF with three overloads. The first overload uses array as the parameter; the second uses map as the parameter; and the third uses struct as the parameter. The third overload uses a struct type as the

parameter or returned value, the UDF class must be supplemented with a `@Resolve` annotation to specify the specific type of struct.

```
@Resolve("struct<a:bigint>,string->string")
public class UdfArray extends UDF {
    public String evaluate(List<String> vals, Long len) {
        return vals.get(len.intValue());
    }
    public String evaluate(Map<String,String> map, String key) {
        return map.get(key);
    }
    public String evaluate(Struct struct, String key) {
        return struct.getFieldValue("a") + key;
    }
}
```

You can import a complex data type in the UDF:

```
create function my_index as 'UdfArray' using 'myjar.jar';
select id, my_index(array('red', 'yellow', 'green'), colorOrdinal) as
color_name from colors;
```

1.5.8.6 Python UDFs

1.5.8.6.1 Restricted environment

MaxCompute UDF uses Python V2.7. It executes user codes in a sandbox. The following operations are restricted in the sandbox:

- Read and write local files.
- Start subprocesses.
- Start threads.
- Conduct socket communication.
- Call other systems.

Due to these restrictions, user-uploaded code must all be implemented by Python, as C extension modules are disabled.

In addition, not all modules in the Python standard library are available for use. Modules that involve the preceding features are disabled. Description of available modules in the standard library:

1. All modules implemented purely by Python are available.

2. The following C extension modules are available for use.

- **array**
- **audioop**
- **binascii**
- **_bisect**
- **cmath**
- **_codecs_cn**
- **_codecs_hk**
- **_codecs_iso2022**
- **_codecs_jp**
- **_codecs_kr**
- **_codecs_tw**
- **_collections**
- **cStringIO**
- **datetime**
- **_functools**
- **future_builtins**
- **_hashlib**
- **_heapq**
- **itertools**
- **_json**
- **_locale**
- **_lsprof**
- **math**
- **_md5**
- **_multibytecodec**
- **operator**
- **_random**
- **_sha256**
- **_sha512**
- **_sha**
- **_struct**
- **strop**

- time
- unicodedata
- _weakref
- cPickle

3. Some modules have limited functionality. For example, the sandbox limits the size that user codes can write to the standard output and standard error output. `sys.stdout` and `sys.stderr` can write up to 20 KB. Any remaining characters are ignored.

1.5.8.6.2 Third-party libraries

Common third-party libraries are installed in the operating environment to supplement the standard library. The supported third-party libraries include NumPy.



Warning:

The use of third-party libraries is also subject to restrictions. For example, local or remote I/O operations are prohibited. Therefore, the related APIs in the third-party libraries are disabled.

1.5.8.6.3 Types of parameters and returned values

You can run the following command to specify the types of parameters and returned values:

```
@odps.udf.annotate(signature)
```

Python UDFs support the following MaxCompute SQL data types: `bigint`, `string`, `double`, `boolean`, and `datetime`. Before you run a SQL statement, you must specify the parameter types and returned value types of all functions. Python is a dynamically-typed language. You need to add decorators to the UDF class to specify the function signature.

The function signature is specified by a string. The syntax is as follows:

```
arg_type_list '->' type_list  
arg_type_list: type_list | '*' | ''  
type_list: [type_list ','] type  
type: 'bigint' | 'string' | 'double' | 'boolean' | 'datetime'
```



Note:

- The part to the left of the arrow indicates the type of parameter. The part to the right of the arrow indicates the type of returned value.
- The returned value of a UDTF can contain multiple columns. The returned value of a UDF or UDAF can contain only one column.
- * represents a variable argument. If a variable argument is specified, the UDF, UDTF, or UDAF can match any type of parameter.

Examples of valid signature:

```
'bigint,double->string'  
-- The parameter is of the bigint or double type, and the returned  
value is of the string type.  
'bigint,boolean->string,datetime'  
-- The UDTF parameter is of the bigint or boolean type, and the  
returned value is of the string or datetime type.  
'*->string'  
-- Specify a variable argument: The input parameter can be of any type  
, and the returned value is of the string type.  
'->double'  
-- The parameter is NULL and the returned value is of the double type.
```

If an invalid signature is found during query parsing, an error is returned and the execution is banned. During execution, the UDF parameter with the type specified by the function signature is transferred to the user. The user returned value must be of the type specified by the function signature. Otherwise, an error is returned. The following table shows the mappings between MaxCompute SQL types and Python types.

Table 1-27: Mapping

MaxCompute SQL type	Python type
Bigint	int
String	str
Double	float
Boolean	bool
Datetime	int



Note:

- A value of the datetime type is passed to user code as the int type. The value is the number of milliseconds that have elapsed since the epoch time. You can use

the datetime module in the Python standard library to process the datetime type

- NULL corresponds to none in Python.

In addition, the parameter of `odps.udf.int(value[, silent=True])` is modified. Parameter `silent` is added. If `silent` is true and the value cannot be converted to the `int` type, none is returned instead of an error.

1.5.8.6.4 UDFs

Implementing a Python UDF is as easy as defining a new-style class and implementing the `evaluate` method.

Example:

```
from odps.udf import annotate
@annotate("bigint,bigint->bigint")
class myplus (object ):
    def evaluate (self, arg0, arg1 ):
        If none in (arg0, arg1 ):
            return none
        return arg0 + arg1
```



Notice:

A Python UDF must have its signature specified through `annotate`.

1.5.8.6.5 UDAFs

Description:

- `class odps.udf.BaseUDAF`: inherit this class to implement a Python UDAF.
- `BaseUDAF.new_buffer()`: implement this method and return the median 'buffer' of the aggregate function. Buffer must be mutable object (such as list and dict). The size of the buffer should not increase with the amount of data. The buffer size should not exceed 2 MB after marshal.
- `BaseUDAF.iterate(buffer[, args, ...])`: This method aggregates args into the median buffer.
- `BaseUDAF.merge(buffer, pBuffer)`: This method aggregates two median buffers; that is, aggregate pBuffer into buffer.
- `BaseUDAF.terminate(buffer)`: This method converts the median 'buffer' into the MaxCompute SQL basic types.

The following example shows how to calculate an average by using a UDAF:

```
#coding:utf-8
from odps.udf import annotate
from odps.udf import BaseUDAF
@annotate('double->double')

class Average(BaseUDAF):
    def new_buffer(self):
        return [0, 0]
    def iterate(self, buffer, number):
        If number is not None:
            buffer[0] += number
            buffer[1] += 1
    def merge(self, buffer, pBuffer):
        buffer [0] + = pBuffer [0]
        buffer [1] + = pBuffer [1]
    def terminate (self, buffer ):
        If buffer [1] = 0:
            return 0.0
        return buffer[0] / buffer[1]
```

1.5.8.6.6 UDTFs

The parameters are described as follows.

Table 1-28: Parameters

Parameter	Description
class odps.udf.BaseUDTF	Base class for a Python UDTF. Users inherit this class and implement methods such as PROCESS and CLOSE.
BaseUDTF.init()	Initialization method. To implement this method for an inherited class, you must call the initialization method super(BaseUDTF, self).init() for the base class at the beginning . The INIT method will only be called once during the entire UDTF life cycle; that is, before the first record is processed. When the UDTF needs to save internal states, all states can be initialized in this method .
BaseUDTF.process([args, ...])	The method is called by the MaxCompute SQL framework. The process method is called for each record passed in from SQL. The parameters passed into the process method are the parameters passed into the UDTF in SQL statements.

Parameter	Description
BaseUDTF.forward([args, ...])	The UDTF output method, which is called by user code. Each time FORWARD is called, one record is output. The parameters of FORWARD are the UDTF output parameters specified in SQL statements.
BaseUDTF.close()	The UDTF termination method. This method is called by the MaxCompute SQL framework. This method is called only once, after the last record is processed.

Example:

```
#coding:utf-8
# explode. py
from odps.udf import annotate

from odps.udf import BaseUDTF
@annotate('string -> string')
class Explode(BaseUDTF):
    -- Output string as multiple comma-separated records.
    def process(self, arg):
        props = arg.split(',')
        for p in props:
            self.forward(p)
```

**Notice:**

A Python UDTF can also specify the parameter type or returned value type without adding 'annotate'. In this case, the function can match any input parameter in SQL. The type of returned value cannot be deduced, but all output parameters will be considered to be of the string type. Therefore, when FORWARD is called, all output values must be converted into values of the string type.

1.5.8.6.7 Reference resources

You can reference file and table resources in Python UDF through the `odps.distcache` module.

Syntax for referencing file resources:

```
odps.distcache.get_cache_file(resource_name)
```

**Note:**

- **Description:** returns the content of the specified resource. `resource_name` is a string that corresponds to the name of an existing resource in the current project. If the resource name is invalid or does not exist, an error is returned.
- **Returned value:** returns file-like object. After this object is used, the caller must call the `CLOSE` method to release the resource file that is opened.

Example:

```
from odps.udf import annotate
from odps.distcache import get_cache_file
@annotate('bigint->string')
class DistCacheExample(object):
    def __init__(self):
        cache_file = get_cache_file('test_distcache.txt')
        kv = {}
        for line in cache_file:
            line = line.strip()
            if not line:
                continue
            k, v = line.split()
            kv[int(k)] = v
        cache_file.close()
        self.kv = kv
    def evaluate(self, arg):
        return self.kv.get(arg)
```

Command syntax:

```
odps.distcache.get_cache_table(resource_name)
```

**Note:**

- **Description:** returns the content of the specified resource table. `resource_name` is a string that corresponds to the name of an existing resource table in the current project. If the resource table name is invalid or does not exist, an error is returned.
- **Returned value:** returns a value of the generator type. The caller traverses the table to obtain the content. Each time the caller traverses the table, a record is obtained in the form of a tuple.

Example:

```
from odps.udf import annotate
from odps.distcache import get_cache_table
@annotate('->string')
class DistCacheTableExample(object):
    def __init__(self):
        self.records = list(get_cache_table('udf_test'))
        self.counter = 0
        self.ln = len(self.records)
```

```
def evaluate(self):  
    if self.counter > self.ln - 1:  
        return None  
    ret = self.records[self.counter]  
    self.counter += 1  
    return str(ret)
```

1.5.9 UDTs

1.5.9.1 Overview

User-defined types (UDTs) are introduced in MaxCompute 2.0 for the latest version of the SQL engine. UDTs allow you to reference classes or objects of third-party languages in SQL statements to obtain data or call methods.

UDTs are typically applied in the following scenarios:

- **Scenario 1:** MaxCompute does not have built-in functions to complete tasks that can be easily performed using other languages. For example, there are some tasks that can be performed by calling a single built-in Java class. Performing these tasks with user defined functions (UDFs) is complex.
- **Scenario 2:** You need to call a third-party library in SQL statements to implement the corresponding feature. You want to use a feature provided by a third-party library directly in a SQL statement, instead of wrapping the feature inside a UDF.
- **Scenario 3:** SELECT TRANSFORM allows you to include objects and classes in SQL statements to make these SQL statements easier to read and maintain. For some languages, such as Java, the source code can be only executed after it is compiled. You want to reference objects and classes of these languages in SQL statements.



Notice:

- UDTs only support Java.
- All operators use the semantics of MaxCompute SQL.
- UDTs cannot be used as shuffle keys in the JOIN, GROUP BY, DISTRIBUTE BY, SORT BY, ORDER BY, and CLUSTER BY clauses.
- DDL statements do not support UDTs. You cannot create tables that contain UDT objects. The final output cannot be UDT types.

1.5.9.2 Feature summary

UDTs allow you to reference classes or objects of third-party languages in SQL statements to obtain data or call methods.

The UDTs supported in MaxCompute are very different from those in other SQL engines.

UDTs supported by other SQL engines are similar to the struct composite type in MaxCompute. UDTs supported by MaxCompute are similar to the CREATE TYPE statement. A UDT contains both fields and methods. Additionally, MaxCompute does not require that you use Data Definition Language (DDL) statements to define type mappings. MaxCompute allows you to reference types directly in SQL statements.

Example:

```
set odps.sql.type.system.odps2=true;
SELECT Integer.MAX_VALUE;
-- A similar output is displayed:
+-----+
| max_value |
+-----+
| 2147483647 |
+-----+
```

The expression in the preceding SELECT statement is similar to a Java expression and executed in the same manner as it would in Java. The expression specifies a UDT in MaxCompute.

You can use UDFs to implement all features provided by UDTs, but with some complexity. If you use a UDF to implement the same feature, you need to follow these steps:

1. Define a UDF class.

```
package com.aliyun.odps.test;
public class IntegerMaxValue extends com.aliyun.odps.udf.UDF {
    public Integer evaluate() {
        return Integer.MAX_VALUE;
    }
}
```

2. Compile the UDF as a JAR package. Upload the JAR package and create a function.

```
add jar odps-test.jar;
```

```
create function integer_max_value as 'com.aliyun.odps.test.  
IntegerMaxValue' using 'odps-test.jar';
```

3. Call the function in a SQL statement.

```
select integer_max_value();
```

A UDT simplifies this procedure. By using UDTs, you can use features provided by other languages in SQL statements.

1.5.9.3 Feature details

The preceding example shows how to use UDTs to access Java static fields. UDTs can be used to implement a number of functions. The following example shows the UDT execution procedure and its features.

```
-- Sample data  
@table1 := select * from values ('1000000000000000000000') as t(x);  
@table2 := select * from values (100L) as t(y);  
-- Logic of the code  
@a := select new java.math.BigInteger(x) x from @table1;  
-- Create a new object  
@b := select java.math.BigInteger.valueOf(y) y from @table2;      --  
Call a static method.  
select /*mapjoin(b)*/ x.add(y).toString() from @a a join @b b;  --  
Call an instance method
```

Command output:

```
1000000000000000000000100
```

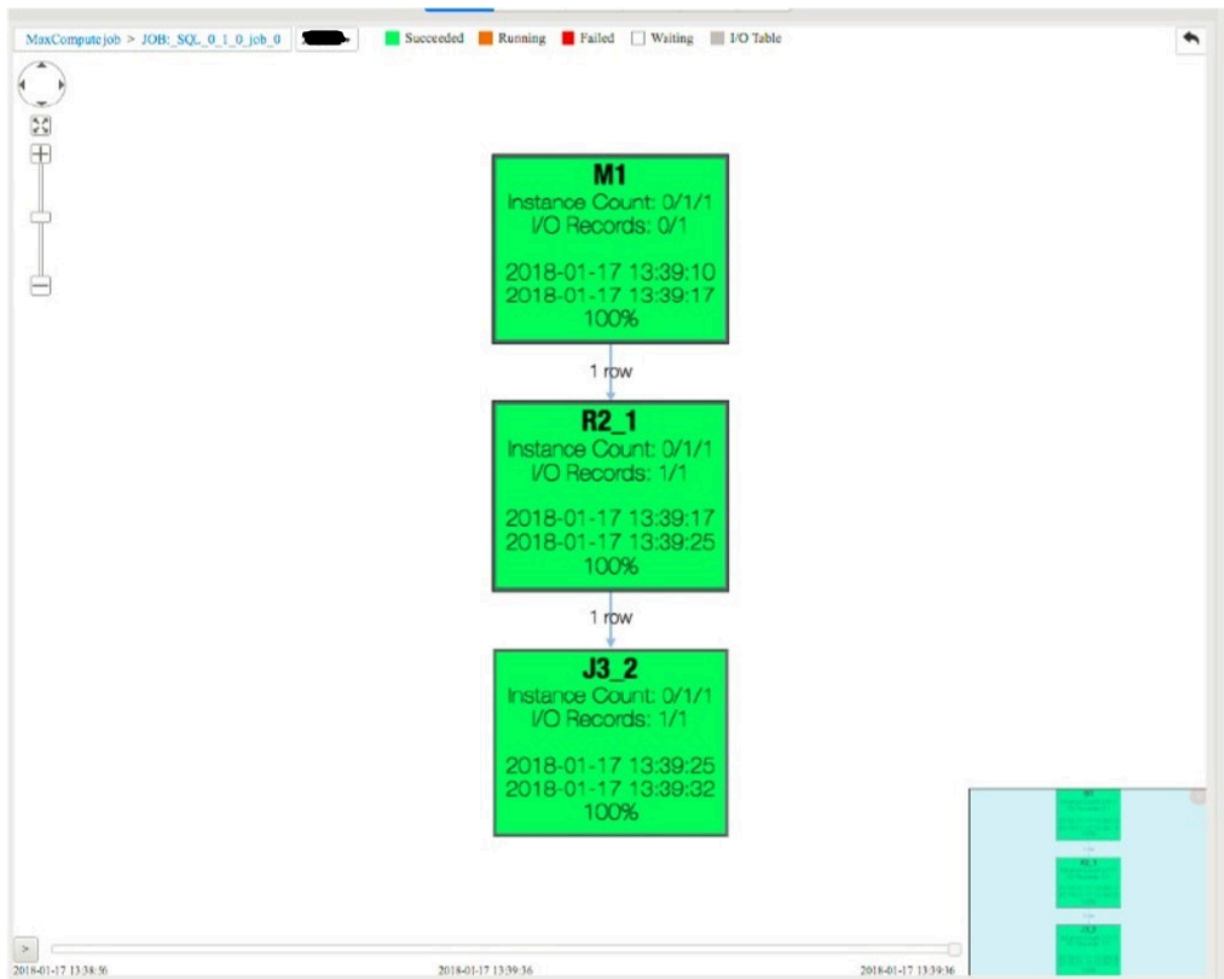


Note:

This example also shows how to use subqueries using UDT columns. This task is difficult to accomplish with UDFs. Variable a is java.math.BigInteger type, but not a built-in type. You can pass the UDT data to another operator and then call its method. You can also use the UDT data in data shuffling.

UDT execution procedure

Figure 1-5: Example



This figure shows that a UDT has three stages: M1, R2, and J3. Only the *new java.math.BigInteger(x)* method is called in the M1 stage. The *java.math.BigInteger.valueOf(y)* and *x.add(y).toString()* methods are called separately at the J3 stage.

When a JOIN clause is used, data must be reshuffled similar to as it would in MapReduce. Data is processed in multiple stages. Typically, data processing at different stages are performed in different processes or different physical machines. The UDT encapsulates these stages and acts as a JVM.

Detailed features

- UDTs only support Java.

- UDTs also allow you to upload JAR packages and directly reference their contents. UDTs have provided flags.
 - **set odps.sql.session.resources:** specifies one or more resources that you need to reference. Separate multiple resources with commas (.). Example: `set odps.sql.session.resources=foo.sh,bar.txt;` Example:

```
set odps.sql.type.system.odps2=true;
set odps.sql.session.resources=odps-test.jar; -- To reference the
JAR package, you must first upload the package to the correspond
ing project and make sure that it is a JAR type resource.
select new com.aliyun.odps.test.IntegerMaxValue().evaluate();
```

**Notice:**

This flag is the same as the resource setting flag in SELECT TRANSFORM. Therefore, this flag controls two features.

- **odps.sql.session.java.imports:** specifies one or more default Java packages. Separate multiple Java packages with commas (.). It is similar to the IMPORT statement in Java. You can specify a class path, such as `java.math.BigInteger`, or use `*.static import` is not supported. Example:

```
set odps.sql.type.system.odps2=true;
set odps.sql.session.resources=odps-test.jar;
set odps.sql.session.java.imports=com.aliyun.odps.test. -- Specify
the default Java package.
select new IntegerMaxValue().evaluate();
```

- UDTs allow you to:
 - Instantiate objects using the new operator.
 - Instantiate arrays using the new operator, including ArrayList initialization. Example: `new Integer[] { 1, 2, 3 }.`
 - Call methods, including static methods. You can create objects in the factory pattern.
 - Access fields, including static fields.

**Notice:**

- Identifiers in UDTs include package names, class names, method names, and field names. All identifiers are case-sensitive.
- UDTs support SQL syntax type conversion, such as `cast (1 as java.lang.Object)`. UDTs do not support Java syntax type conversion, such as `(Object)1`.

- Anonymous classes and lambda expressions are not supported.
- Functions that do not return values cannot be called in UDTs.

**Note:**

This is because UDTs are typically used in expressions and functions that do not return values cannot be called in expressions.

- All Java SDK classes can be referenced by UDTs. The JDK runtime environment is JDK 1.8. Later versions may not be supported.
- All operators use the semantics of MaxCompute SQL. The result of `String.valueOf(1) + String.valueOf(2)` is 3. The two strings are implicitly converted to double type values and summed. If you use Java string concatenation to merge the strings, the result will be 12.

In addition to the string concatenation methods in MaxCompute and Java, you may also have confusion about the `=` operator. The `=` operator in SQL statements is used as a comparison operator. You must call the `equals` method in Java to compare whether two objects are equivalent. The `=` operator cannot be used to verify the equivalence of two objects.

- Java data types are mapped to built-in data types. The mapping table can be applied to UDTs.
 - Built-in type data can directly call the method of the Java type to which the built-in type is mapped. Example: `'123'.length()` , `1L.hashCode()`.
 - UDTs can be used in built-in functions and UDFs. For example, in `chr(Long.valueOf('100'))`, `Long.valueOf` returns a `java.lang.Long` type value. Built-in function `chr` supports the built-in type of `BIGINT`.
 - Java primitive type data is automatically converted to boxing type data and the preceding two rules can be applied in this situation.

**Notice:**

For certain built-in new data types, you must add the `set odps.sql.type.system.odps2=true` statement to declare these types. Otherwise, an error occurs.

- UDTs completely support Java generics. For example, the compiler can determine that the value returned by the `java.util.Arrays.asList(new java`

`.math.BigInteger('1'))` method is `java.util.List<java.math.BigInteger>` based on the parameter type.

**Notice:**

You must set the type parameter in a construct function or use `java.lang.Object`. This is the same as in Java. You must set the type parameter in a construct function or use `java.lang.Object`. This is the same as in Java. For example, the result of `new java.util.ArrayList(java.util.Arrays.asList('1', '2'))` is `java.util.ArrayList<Object>`. The result of `new java.util.ArrayList<String>(java.util.Arrays.asList('1', '2'))` is `java.util.ArrayList<String>`.

- UDTs do not have a clear definition of object equality. This is caused by data reshuffling. The JOIN example shows that objects may be transmitted between different processes or physical machines. During the transmission, an object may be referenced as two different objects. For example, an object may be shuffled to two machines and then reshuffled.

Therefore, when you use UDTs, you must use the `equals` method to compare two objects instead of using the `=` operator.

**Note:**

Objects in the same row or column are guaranteed to be correlated in some way. However, there may not be a correlation between objects in different rows and columns.

- UDTs cannot be used as shuffle keys in the JOIN, GROUP BY, DISTRIBUTE BY, SORT BY, ORDER BY, and CLUSTER BY clauses.

UDTs can be used in any stages in expressions, but cannot be output as final results. For example, you cannot call the `group by new java.math.BigInteger('123')` method, but can call the `group by new java.math.BigInteger('123').hashCode()` method. This is because the value returned by `hashCode` is an `int.class` type, which can be used as a built-in type of INT.

- The following type conversion rules are extended in UDTs:
 - UDT objects can be converted to objects of its base classes by implicit conversion.
 - UDT objects can be forcibly converted to objects of its base classes or subclasses.
 - Data type conversion for two objects without inheritance follows the native conversion rules.

**Notice:**

The conversion may change the data. For example, `java.lang.Long` type data can be forcibly converted to `java.lang.Integer` type data. This process converts built-in BIGINT type data to INT type data. This process may cause data changes or even data precision changes.

- UDT objects cannot be saved or added to tables. DDL statements do not support UDTs. You cannot create tables that contain UDT objects unless the data is implicitly converted to one of the built-in types. In addition, the final output cannot be UDT types. However, you can call the `toString()` method to convert the UDT data to `java.lang.String` type data because the `toString()` method supports all Java types. You can use this method to check UDT data during debugging.

You can also add the `set odps.sql.udt.display.toString=true;` statement to enable MaxCompute to convert all UDT data to strings when the `java.util.Objects.toString(...)` method is called.

**Note:**

This flag is typically used for debugging because it can only be applied to the print statement. It cannot be applied to the INSERT statement.

BINARY is a built-in type and supports automatic serialization. You can then save the `byte[]` arrays. The saved `byte[]` arrays can be deserialized to BINARY type.

Some types may have their own serialization and deserialization methods, such as `protobuf`. To save UDTs, you must call serialization and deserialization methods to convert the data to BINARY data.

- You can use UDTs to achieve the feature provided by the SCALAR function. With built-in functions `COLLECT_LIST` and `EXPLODE`, you can use UDTs to achieve the features provided by aggregate and table functions.

1.5.9.4 More examples

1.5.9.4.1 Example of using Java arrays

Example:

```
set odps.sql.type.system.odps2=true;
set odps.sql.udt.display.tostring=true;
select
    new Integer[10],      -- Create an array that contains 10 elements.
    new Integer[] {c1, c2, c3}, -- Create an array that contains
three elements by initializing an ArrayList.
    new Integer[][] { new Integer[] {c1, c2}, new Integer[] {c3, c4
} }, -- Create a multidimensional array.
    new Integer[] {c1, c2, c3} [2], -- Access the elements in the
array using indexes.
    java.util.Arrays.asList(c1, c2, c3); -- This is another way to
create a built-in array. It creates a List<Integer>, which can be used
as an array<int>.
from values (1,2,3,4) as t(c1, c2, c3, c4);
```

1.5.9.4.2 Example of using JSON

The runtime of UDT carries a GSON dependency (version 2.2.4), which can be directly used in GSON.

Example:

```
set odps.sql.type.system.odps2=true;
set odps.sql.session.java.imports=java.util.*,java.com.google.gson.*;
-- To import multiple packages, separate the packages with commas
(,).
@a := select new Gson() gson; -- Create a GSON object.
select
    gson.toJson(new ArrayList<Integer>(Arrays.asList(1, 2, 3))), --
Convert an object to a JSON string.
    cast(gson.fromJson('["a","b","c"]', List.class) as List<String>)
--Deserialize the JSON string. GSON also forcibly converts the
deserialized result from List<Object> type to List<String> type.
from @a;
```

Compared with built-in function GET_JSON_OBJECT, this method is simple and improves efficiency by extracting content from the JSON string and deserializing the string to a supported data type.

In addition to GSON dependencies, MaxCompute runtime also carries other dependencies, including commons-logging (1.1.1), commons-lang (2.5), commons-io (2.4), and protobuf-java (2.4.1).

1.5.9.4.3 Example of using composite types

Built-in types of array and map are mapped to java.util.List and java.util.Map, respectively.

- Java objects in classes calling the `java.util.List` or `java.util.Map` API can be used in MaxCompute SQL composite type data processing.
- Array and map type data in MaxCompute can directly call the `java.util.List` or `java.util.Map` API.

Example:

```
set odps.sql.type.system.odps2=true;
set odps.sql.session.java.imports=java.util.*;
select
    size(new ArrayList<Integer>()),          -- Call built-in function
    size to obtain the size of the ArrayList.
    array(1,2,3).size(),                    -- Call the List method for
    built-in type array.
    sort_array(new ArrayList<Integer>()),    -- Sort the data in the
    ArrayList.
    al[1],                                  -- The Java List method
    does not support indexing. However, the array type supports indexing.
    Objects.toString(a),                    -- With this method, you can convert
    array type to string type data.
    array(1,2,3).subList(1, 2)              -- Get a sublist.
from (select new ArrayList<Integer>(array(1,2,3)) as al, array(1,2,3)
as a) t;
```

1.5.9.4.4 Example of aggregation

To achieve aggregation with UDTs, you must first use built-in function `COLLECT_SET` or `COLLECT_LIST` to convert the data to the List type and then call the UDT methods to aggregate the data.

The following example shows how to obtain the median from `BigInteger` data. You cannot directly call the built-in `MEDIAN` function because the data is `java.math.BigInteger` type.

```
set odps.sql.session.java.imports=java.math.*;
@test_data := select * from values (1),(2),(3),(5) as t(value);
@a := select collect_list(new BigInteger(value)) values from @
test_data; -- Aggregate the data to a list.
@b := select sort_array(values) as values, values.size() cnt from @a;
-- To obtain the median, first sort the data.
@c := select if(cnt % 2 == 1, new BigDecimal(values[cnt div 2]), new
    BigDecimal(values[cnt div 2 - 1].add(values[cnt div 2])).divide(new
    BigDecimal(2))) med from @b;
-- Final output.
select med.toString() from @c;
```

You cannot use the `COLLECT_LIST` function to implement partial aggregation because it aggregates all data. It is more efficient to use the built-in aggregator or UDAF object. We recommend that you use the built-in aggregator. Aggregating all data in a group increases the risk of data skew.

If the logic of the UDAF object is to aggregate all data in a similar manner to built-in function WM_CONCAT, using the COLLECT_LIST function is more efficient than using the UDAF object.

1.5.9.4.5 Example of using table-valued functions

Table-valued functions allow you to input and output multiple rows and columns. To input or output multiple rows and columns, follow these steps:

1. For more information about how to input multiple rows or columns, see the example of using aggregate functions.
2. To output multiple rows, you can use a UDT to define a Collection type (List or Map), and then call the EXPLODE function to split the collection into multiple rows.
3. A UDT can contain multiple fields. You can retrieve the data from the fields by calling different getter methods. The data is then output in multiple rows.

The following example shows how to split a JSON string and output the result as multiple columns:

```
@a := select ' [{"a":"1","b":"2"}, {"a":"1","b":"2"} ] ' str; -- Sample data
@b := select new com.google.gson.Gson().fromJson(str, java.util.List.class) l from @a; -- Deserialize the JSON string.
@c := select cast(e as java.util.Map<Object,Object>) m from @b lateral view explode(l) t as e; -- Call the EXPLODE function to split the string.
@d := select m.get('a') as a, m.get('b') as b from @c; -- Output the splitting result in multiple columns.
select a.toString() a, b.toString() b from @d; -- The final output.
Columns a and b in variable d are of the Object type.
```

1.5.9.5 Feature advantages

UDT has the following features:

- Easy to use. You do not need to define any functions.
- To improve the flexibility of SQL, all JDK supported features can be used directly.
- You can directly reference objects and classes of other languages in SQL statements.
- You can directly reference the libraries of other language and reuse code that you have written in other languages.
- You can create object-oriented features.

1.5.9.6 Performance advantages

UDTs and UDFs use similar execution procedures and provide similar performance. However, UDTs have higher performance in certain scenarios where the compute engine has been greatly improved.

- Deserialization is not required for objects in only one process. Deserialization is required only when the objects are transmitted among processes. This means that UDT do not incur any serialization or deserialization overhead when no data reshuffling is performed, such as calling the join or aggregator function.
- UDTs suffer no performance loss from reflection because the runtime of UDTs is based on Codegen, rather than based on reflection.
- Multiple UDTs can be wrapped into a single function call and executed together. In the following example, a single UDT is being called. UDTs focus on small-granularity data processing. This does not incur additional overhead for the API where multiple functions are called.

```
values[x].add(values[y]).divide(java.math.BigInteger.valueOf(2))
```

1.5.9.7 Security advantages

UDTs are restricted in the Java sandbox model similar to UDFs. To perform restricted operations, you must enable sandbox isolation or apply to join the sandbox whitelist.

1.5.10 UDJ

1.5.10.1 Overview

MaxCompute provides multiple JOIN methods natively, including INNER JOIN, RIGHT JOIN, OUTER JOIN, LEFT JOIN, FULL JOIN, SEMIJOIN, and ANTISEMIJOIN methods. You can use these native JOIN methods in most scenarios. However, these methods cannot handle multiple tables.

In most cases, you can build your code framework using UDFs. However, the current UDF, UDTF, and UDAF frameworks only can handle one table at a time. To perform user-defined operations for multiple tables, you have to use native JOIN methods, UDFs, UDTFs, and complex SQL statements. In certain cases when you handle multiple tables, you must use a custom MapReduce framework instead of SQL to complete the required task.

In any situation, these operations require technological expertise and may cause the following problems:

- Calling multiple JOIN methods in SQL statements can lead to computational black box that is complex and difficult to execute with minimal overheads.
- Using MapReduce even make optimal execution of code becomes impossible. Most of the MapReduce code is written in Java. The execution of the MapReduce code is less efficient than the execution of MaxCompute code generated by the LLVM code generator at an optimized native runtime.

With the addition of the MaxCompute 2.0 compute engine, the user defined join (UDJ) API has been added to the user defined function (UDF) framework. This API allows you to handle multiple tables and simplifies operations performed in the underlying MapReduce distributed system.

1.5.10.2 UDJ usage

1.5.10.2.1 Examples

The following example describes how to use UDJ in MaxCompute.

This example uses the payment table and the user_client_log table.

- The payment (user_id string,time datetime,pay_info string) table stores the payment information of a user. Each payment record includes the user ID, payment time, and the payment details.
- The user_client_log (user_id string,time datetime,content string) table stores user client records, including the user ID, operation time, and operation.

Requirements: For each record in the user_client_log table, locate the payment record that has the time closest to the operation time, and join and output the content of both records.

To complete this task by using standard join methods, you would need to join the two tables based on their common user_id fields, and then locate the payment record and operation that most closely match each other's time. The SQL statement may be written as follows:

```
SELECT
  p.user_id,
  p.time,
  merge(p.pay_info, u.content)
FROM
  payment p RIGHT OUTER JOIN user_client_log u
```

```
ON p.user_id = u.user_id and abs(p.time - u.time) = min(abs(p.time - u.time))
```

However, when you join two rows in the tables, you must calculate the minimum difference between the p.time and u.time under the same user_id, and the aggregate function cannot be called in the join condition. Because of this, this task cannot be completed by calling the standard JOIN method.

Can we use UDJ to solve this problem? Yes. The following topics describe how to use UDJ to satisfy the preceding requirements.

1.5.10.2.2 Use Java to write the UDJ code

Prerequisites

UDJ is a new feature, so a new SDK is required.

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>odps-sdk-udf</artifactId>
  <version>0.30.0</version>
  <scope>provided</scope>
</dependency>
```

The SDK contains a new abstract class UDJ. All UDJ features can be implemented through this class.

Sample code

The following sample code is used for reference only.

```
package com.aliyun.odps.udf.example.udj;
import com.aliyun.odps.Column;
import com.aliyun.odps.OdpsType;
import com.aliyun.odps.Yieldable;
import com.aliyun.odps.data.ArrayRecord;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.udf.DataAttributes;
import com.aliyun.odps.udf.ExecutionContext;
import com.aliyun.odps.udf.UDJ;
import com.aliyun.odps.udf.annotation.Resolve;
import java.util.ArrayList;
import java.util.Iterator;
/** For each record of right table, find the nearest record of left
 * table and
 * merge two records.
 */
@Resolve("->string,bigint,string")
public class PayUserLogMergeJoin extends UDJ {
    private Record outputRecord;
    /** Will be called prior to the data processing phase. User could
    implement
    * this method to do initialization work.
    */
    @Override
```

```
public void setup(ExecutionContext executionContext, DataAttributes
dataAttributes) {
    //
    outputRecord = new ArrayRecord(new Column[]{
        new Column("user_id", OdpsType.STRING),
        new Column("time", OdpsType.BIGINT),
        new Column("content", OdpsType.STRING)
    });
}
/** Override this method to implement join logic.
 * @param key Current join key
 * @param left Group of records of left table corresponding to the
current key
 * @param right Group of records of right table corresponding to the
current key
 * @param output Used to output the result of UDJ
 */
@Override
public void join(Record key, Iterator<Record> left, Iterator<Record
> right, Yieldable<Record> output) {
    outputRecord.setString(0, key.getString(0));
    if (! right.hasNext()) {
        // Empty right group, do nothing.
        return;
    } else if (! left.hasNext()) {
        // Empty left group. Output all records of right group without
merge.
        while (right.hasNext()) {
            Record logRecord = right.next();
            outputRecord.setBigint(1, logRecord.getDatetime(0).getTime());
            outputRecord.setString(2, logRecord.getString(1));
            output.yield(outputRecord);
        }
        return;
    }
    ArrayList<Record> pays = new ArrayList<>();
    // The left group of records will be iterated from the start to
the end
    // for each record of right group, but the iterator cannot be
reset.
    // So we save every records of left to an ArrayList.
    left.forEachRemaining(pay -> pays.add(pay.clone()));
    while (right.hasNext()) {
        Record log = right.next();
        long logTime = log.getDatetime(0).getTime();
        long minDelta = Long.MAX_VALUE;
        Record nearestPay = null;
        // Iterate through all records of left, and find the pay record
that has
        // the minimal difference in terms of time.
        for (Record pay: pays) {
            long delta = Math.abs(logTime - pay.getDatetime(0).getTime());
            if (delta < minDelta) {
                minDelta = delta;
                nearestPay = pay;
            }
        }
        // Merge the log record with nearest pay record and output to
the result.
        outputRecord.setBigint(1, log.getDatetime(0).getTime());
        outputRecord.setString(2, mergeLog(nearestPay.getString(1), log.
getString(1)));
        output.yield(outputRecord);
    }
}
```

```

    }
    String mergeLog(String payInfo, String logContent) {
        return logContent + ", pay " + payInfo;
    }
    @Override
    public void close() {
    }
}

```

**Notice:**

In this example, the NULL values in the entries are not processed. To simplify the data processing procedure, assume that no NULL values are contained in the tables.

Each time you call this JOIN method of UDJ, records that match the same key in the two tables are returned. Therefore, UDJ searches all records in the payment table to locate the record with the time closest to each record in the user_client_log table.

Assume that the user only has a few payment records. In this case, you can load the data in the payment table to the memory. Typically, there is sufficient memory to store the user payment data generated each day. What if this assumption is invalid? How can we resolve this issue? This issue will be discussed in Pre-sorting.

1.5.10.2.3 Create a UDJ function in MaxCompute

After you have written the UDJ code in Java, upload the code to MaxCompute SQL as a plug-in. You must have registered the code with MaxCompute first.

Assume that the code is compressed into JAR package `odps-udj-example.jar`. Use the Add JAR command to upload the JAR package to MaxCompute.

```
add jar odps-udj-example.jar;
```

Execute the CREATE FUNCTION statement to create UDJ function `pay_user_log_merge_join`, using JAR package `odps-udj-example.jar` and Java class `com.aliyun.odps.udf.example.udj.PayUserLogMergeJoin`.

```
create function pay_user_log_merge_join
as 'com.aliyun.odps.udf.example.udj.PayUserLogMergeJoin'
using 'odps-udj-example.jar';
```

1.5.10.2.4 Use UDJ in MaxCompute SQL

After you have registered UDJ in the database, UDJ can be used in MaxCompute SQL.

1. Create a sample source table.

```
create table payment (user_id string,time datetime,pay_info string);
create table user_client_log(user_id string,time datetime,content
string);
```

2. Create sample data.



Notice:

The data in this example is only used for reference. You may need to create different data in actual operations.

```
-- Create data in the payment table
INSERT OVERWRITE TABLE payment VALUES
('1335656', datetime '2018-02-13 19:54:00', 'PEqMSHyktn'),
('2656199', datetime '2018-02-13 12:21:00', 'pYvotuLDIT'),
('2656199', datetime '2018-02-13 20:50:00', 'PEqMSHyktn'),
('2656199', datetime '2018-02-13 22:30:00', 'gZhvdysOQb'),
('8881237', datetime '2018-02-13 08:30:00', 'pYvotuLDIT'),
('8881237', datetime '2018-02-13 10:32:00', 'KBuMzRpsko'),
('9890100', datetime '2018-02-13 16:01:00', 'gZhvdysOQb'),
('9890100', datetime '2018-02-13 16:26:00', 'MxONdLckwa')
;
-- Create data in the user_client_log table
INSERT OVERWRITE TABLE user_client_log VALUES
('1000235', datetime '2018-02-13 00:25:36', 'click FNOXAibRjkIaQPB'),
('1000235', datetime '2018-02-13 22:30:00', 'click GczrYaxvkiPultZ'),
('1335656', datetime '2018-02-13 18:30:00', 'click MxONdLckpAFUHRs'),
('1335656', datetime '2018-02-13 19:54:00', 'click mKRPGOcIFDyzTgM'),
('2656199', datetime '2018-02-13 08:30:00', 'click CZwafHsbJOPNitL'),
('2656199', datetime '2018-02-13 09:14:00', 'click nYHJqIpjevktToy'),
('2656199', datetime '2018-02-13 21:05:00', 'click gbAfPCwrGXvEjpI'),
('2656199', datetime '2018-02-13 21:08:00', 'click dhpZyWMuGjBOTJP'),
('2656199', datetime '2018-02-13 22:29:00', 'click bAsxnUdDhvfqaBr'),
('2656199', datetime '2018-02-13 22:30:00', 'click XIhZdLaOocQRmrY'),
('4356142', datetime '2018-02-13 18:30:00', 'click DYqShmGbIoWKier'),
('4356142', datetime '2018-02-13 19:54:00', 'click DYqShmGbIoWKier'),
('8881237', datetime '2018-02-13 00:30:00', 'click MpkvilgWSmhUuPn'),
('8881237', datetime '2018-02-13 06:14:00', 'click OkTYNUHMqZzLDyL'),
('8881237', datetime '2018-02-13 10:30:00', 'click OkTYNUHMqZzLDyL'),
('9890100', datetime '2018-02-13 16:01:00', 'click vOTQfBFjcgXisYU'),
('9890100', datetime '2018-02-13 16:20:00', 'click WxaLgOCcVEvhiFJ')
```


;

3. In MaxCompute SQL, use the UDJ function you have created:

```

SELECT r.user_id, from_unixtime(time/1000) as time, content FROM (
SELECT user_id, time as time, pay_info FROM payment
) p JOIN (
SELECT user_id, time as time, content FROM user_client_log
) u
ON p.user_id = u.user_id
USING pay_user_log_merge_join(p.time, p.pay_info, u.time, u.content)
r
AS (user_id, time, content)
;

```

**Note:**

The syntax of UDJ is similar to that of the standard JOIN statement. The only difference is that the USING clause is added to UDJ.

Description:

- `pay_user_log_merge_join` is the name of the UDJ function in SQL.
- `(p.time, p.pay_info, u.time, u.content)` are the columns used in these two tables.
- `r` is the alias of the result returned by the UDJ function. You can reference this alias in other SQL statements.
- `(user_id, time, content)` are the columns returned by the UDJ function.

4. Execute this SQL statement. A similar output is displayed:

user_id	time	content
1000235	2018-02-13 00:25:36	click FNOXAibRjkIaQPB
1000235	2018-02-13 22:30:00	click GczrYaxvkiPultZ
1335656	2018-02-13 18:30:00	click MxONdLckpAFUHSR, pay
1335656	2018-02-13 19:54:00	click mKRPgOciFDyzTgM, pay
2656199	2018-02-13 08:30:00	click CZwafHsbJOPNitL, pay
2656199	2018-02-13 09:14:00	click nYHJqIpjevKkToy, pay
2656199	2018-02-13 21:05:00	click gbAfPCwrGXvEjpI, pay
2656199	2018-02-13 21:08:00	click dhpZyWMuGjBOTJP, pay
2656199	2018-02-13 22:29:00	click bAsxnUdDhvfqaBr, pay
2656199	2018-02-13 22:30:00	click XIhZdLaOocQRmrY, pay
4356142	2018-02-13 18:30:00	click DYqShmGbIoWKier
4356142	2018-02-13 19:54:00	click DYqShmGbIoWKier
8881237	2018-02-13 00:30:00	click MpkvilgWSmhUuPn, pay

```
| 8881237 | 2018-02-13 06:14:00 | click OkTYNUHMqZzlDyL, pay
pYvotuLDIT |
| 8881237 | 2018-02-13 10:30:00 | click OkTYNUHMqZzlDyL, pay
KBuMzRpsko |
| 9890100 | 2018-02-13 16:01:00 | click vOTQfBFjcgXisYU, pay
gZhvdYSOQb |
| 9890100 | 2018-02-13 16:20:00 | click WxaLgOCcVEvhiFJ, pay
MxONdLckwa |
+-----+-----+-----+
```

As shown in the preceding code, the task that could not be performed by calling native JOIN methods has been completed by using UDJ.

1.5.10.2.5 Pre-sorting

An iterator is used to search all records in the payment table and locate payment records that match the query. To perform this task, you must load all payment records with the same user_id to an ArrayList. This method can be applied when the number of payment records is small. Due to RAM size limits, you must find another method to load the data if a large number of payment records have been generated.

This topic describes how to address this issue using the SORT BY clause. When the size of the payment data is too large to be stored in the memory, it would be easier to address this issue if all data in the table has already been sorted by time. You then only need to compare the first element in these two lists. UDJ code in Java:

```
@Override
public void join(Record key, Iterator<Record> left, Iterator<Record>
right, Yieldable<Record> output) {
    outputRecord.setString(0, key.getString(0));
    if (! right.hasNext()) {
        return;
    } else if (! left.hasNext()) {
        while (right.hasNext()) {
            Record logRecord = right.next();
            outputRecord.setBigint(1, logRecord.getDatetime(0).getTime());
            outputRecord.setString(2, logRecord.getString(1));
            output.yield(outputRecord);
        }
        return;
    }
    long prevDelta = Long.MAX_VALUE;
    Record logRecord = right.next();
    Record payRecord = left.next();
    Record lastPayRecord = payRecord.clone();
    while (true) {
        long delta = logRecord.getDatetime(0).getTime() - payRecord.
getDatetime(0).getTime();
        if (left.hasNext() && delta > 0) {
            // The delta of time between two records is decreasing, we can
            still
            // explore the left group to try to gain a smaller delta.
            lastPayRecord = payRecord.clone();
            prevDelta = delta;
            payRecord = left.next();
        }
    }
}
```

```

    } else {
        // Hit to the point of minimal delta. Check with the last pay
        record,
        // output the merge result and prepare to process the next
        record of
        // right group.
        Record nearestPay = Math.abs(delta) < prevDelta ? payRecord :
        lastPayRecord;
        outputRecord.setBigint(1, logRecord.getDatetime(0).getTime());
        String mergedString = mergeLog(nearestPay.getString(1),
        logRecord.getString(1));
        outputRecord.setString(2, mergedString);
        output.yield(outputRecord);
        if (right.hasNext()) {
            logRecord = right.next();
            prevDelta = Math.abs(
                logRecord.getDatetime(0).getTime() - lastPayRecord.
                getDatetime(0).getTime()
            );
        } else {
            break;
        }
    }
}
}
}

```

**Notice:**

After you have modified the UDJ code, you must update the corresponding JAR package.

When the created UDJ function is used in MaxCompute SQL, you must modify the command as follows:

```

SELECT r.user_id, from_unixtime(time/1000) as time, content FROM (
    SELECT user_id, time as time, pay_info FROM payment
) p JOIN (
    SELECT user_id, time as time, content FROM user_client_log
) u
ON p.user_id = u.user_id
USING pay_user_log_merge_join(p.time, p.pay_info, u.time, u.content)
r
AS (user_id, time, content)
SORT BY p.time, u.time
;

```

In the native SQL language, you must make a few modifications, add a SORT BY clause to the end of the UDJ clause, and then sort the data in both tables by time.

The execution result is the same as the result before the code is modified.

This method uses the SORT BY clause to pre-sort the data. To achieve the same result, only a maximum of three records need to be cached.

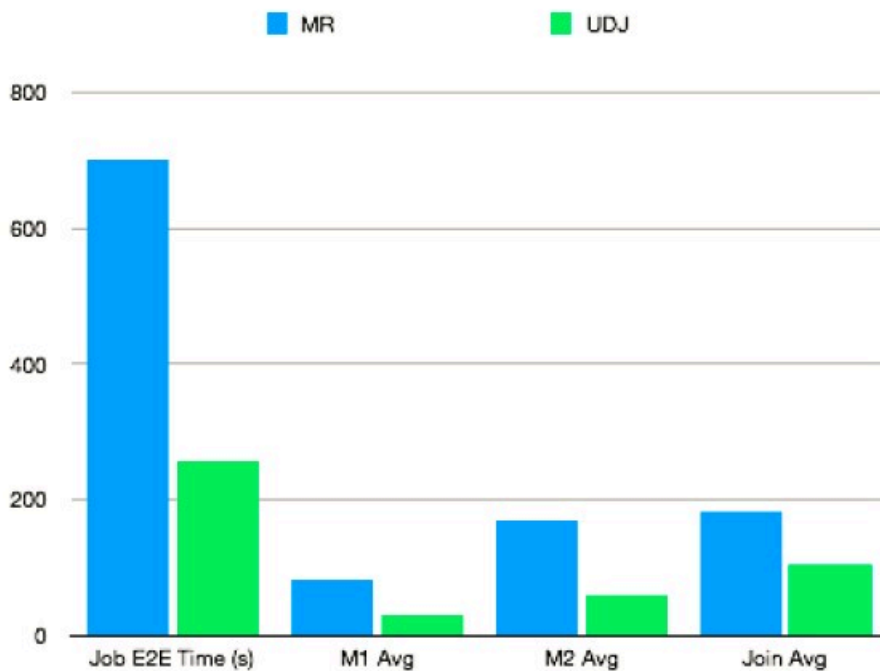
1.5.10.3 Performance advantages

Without UDJ, you must use MapReduce to handle complex cross-table computing tasks in a distributed system.

The following example uses an online MapReduce job to test the UDJ performance. This MapReduce job uses a complex algorithm to join two tables. This example uses UDJ to rewrite the SQL statements of the MapReduce job and checks the execution results.

Under the same programming concurrency, the comparison of performance is as follows.

Figure 1-6: Performance comparison



As shown in the figure, UDJ helps describe the complex logic of handling multiple tables, and greatly improves the query performance.



Note:

The code is only executed inside UDJ. The entire logic of the code is executed by the high-performance MaxCompute native runtime.

UDJ optimizes the MaxCompute runtime engine and the data exchange between interfaces. The join logic of UDJ is more efficient than that of the reduce stage.

1.5.11 MaxCompute SQL limits

The following table lists all MaxCompute SQL limits.

Table 1-29: Limits

Item	Maximum or limit	Category	Description
Table name length	128 bytes	Length	A table name or column name cannot contain special characters. It can contain only lowercase and uppercase letters, digits, and underscores (_). A name must start with a letter.
Comment length	1,024 bytes	Length	A comment can be up to 1,024 bytes in length.
Column definitions in a table	1,200	Quantity	A table can contain a maximum of 1,200 column definitions.
Partitions in a table	60,000	Quantity	A table can contain a maximum of 60,000 partitions.
Partition levels of a table	6	Quantity	A table can contain a maximum of six partition levels.
Statistical definitions of a table	100	Quantity	A table can contain a maximum of 100 statistical definitions.
Statistical definition length of a table	64,000	Length	The length of statistic definitions in a table cannot exceed 64,000.
Screen display	10,000 rows	Quantity	A SELECT statement can output a maximum of 10,000 rows.
INSERT targets	256	Quantity	A MULTIINS operation can insert a maximum of 256 data tables at a time.
UNION ALL	256 tables	Quantity	The UNION ALL operation can be performed on a maximum of 256 tables.
JOIN sources	16	Quantity	The JOIN operation can be performed on a maximum of 16 source tables.

Item	Maximum or limit	Category	Description
MAPJOIN	256 small tables	Quantity	A MAPJOIN operation can be performed on a maximum of 256 small tables.
MAPJOIN memory	512 MB	Quantity	The memory size for all small tables on which the MAPJOIN operation is performed cannot exceed 512 MB.
Window functions	5	Quantity	A SELECT statement can contain a maximum of five window functions.
PTINSUBQ	1,000 rows	Quantity	A PT IN SUBQUERY statement can output a maximum of 1,000 rows.
Length of a SQL statement	2 MB	Length	The maximum size of a SQL statement is 2 MB.
Conditions of a WHERE clause	256	Quantity	A WHERE clause can contain a maximum of 256 conditions.
Length of a column record	8 MB	Quantity	The maximum length of a table record is 8 MB.
IN parameters	1,024	Quantity	It specifies the maximum number of parameters in an IN clause, such as in(1,2,3, ..., 1024). A large number of parameters of an IN clause can slow down the compilation process. We recommend that you use no more than 1,024 parameters, but this is not the actual upper limit.
jobconf.json	1 MB	Length	The maximum size of the jobconf.json file is 1 MB. If a table contains a large number of partitions, the size of jobconf.json may exceed 1 MB.
View	Not writable	Operation	A view is not writable and does not support the INSERT operation.
Data type and position of a column	Unmodifiable	Operation	The data type and position of a column are not modifiable.
Java UDFs	Cannot be abstract or static	Operation	Java UDFs cannot be abstract or static.

Item	Maximum or limit	Category	Description
Query partitions	10,000	Quantity	A maximum of 10,000 partitions can be queried.

**Notice:**

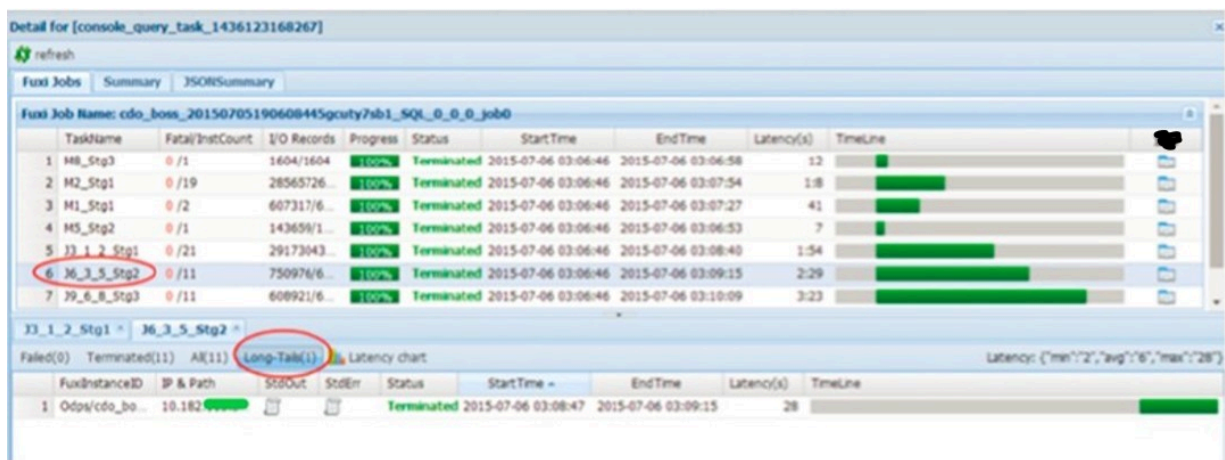
The preceding MaxCompute SQL limits cannot be modified manually.

1.5.12 Common MaxCompute SQL errors and solutions

1.5.12.1 Data skew

1.5.12.1.1 Overview

For a running job instance where the min, max, and avg values for the parameters time, input records, and output records are imbalanced (for example, max is much greater than avg), a data skew problem may have occurred. You can check the log view to locate the data skew problem, as shown in the following figure.



The Long Tails tab of each task shows the instance where the data skew occurred. The root cause of data skew is that the amounts of data processed by some instances are much higher than that processed by other instances, causing the running time of these instances to exceed the average time of other instances. As a result, the entire job slows down.

You can reduce the data skew of different SQL data types using different methods.

1.5.12.1.2 GROUP BY skew

Possible cause: The unbalanced distribution of GROUP BY keys causes data skew in the Reduce step.

Solution: Enable the group skew prevention parameter before running SQL statements:

```
set odps.sql.groupby.skewindata=true
```



Note:

If this parameter is set to true, the system adds random factors to the shuffle hash algorithm and adds a new task to prevent data skew.

1.5.12.1.3 DISTRIBUTE BY skew

Possible cause: Using constants for full-table sorting in DISTRIBUTE BY mode will result in data skew at the Reduce end.

Solution: Avoid the preceding operation.

1.5.12.1.4 JOIN skew

Possible cause: The unbalanced distribution of join on keys (such as a large number of repeated keys in multiple JOIN tables) causes surging Cartesian product data in some JOIN instances, which results in data skew.

Solution: The solutions to different scenarios are as follows:

- If there are small tables on both sides of 'join', perform 'map join' instead of 'join'.
- The skewed key can be dealt with by using individual logic. For example, a large amount of NULL data in keys on both sides of a table results in skew. In this case, you need to filter out the NULL data before performing the JOIN operation or replacing NULL values with random values by using the CASE WHEN clause, and then do JOIN operation.
- If you do not want to change SQL statements, set the following parameters to enable automatic optimization on MaxCompute:

```
set odps.sql.skewinfo=tab1:(col1, col2)[(v1, v2), (v3, v4), ...]  
set odps.sql.skewjoin=true;
```

1.5.12.1.5 MULTI-DISTINCT skew

Possible cause: Multiple DISTINCT keywords aggravate the GROUP BY skew problem.

Solution: You can use a two-layer GROUP BY to smooth the skew.

1.5.12.1.6 Data skew caused by misuse of dynamic partitioning

Possible cause: If dynamic partitioning is enabled, and there are K map instances and N target partitions, a number of small files ($K * N$) may be generated. A large amount of small files can greatly increase the management workload of the file system. Therefore, the following configuration takes effect by default:

```
set odps.sql.reshuffle.dynamicpt=true;
```

It introduces an additional level of ReduceTask to allow one or more reduce instances to write data to the same target partition. This prevents too many small files from being generated. However, dynamic partition shuffle may cause data skew.

Solution: If there are only a few target partitions, the system will not generate many small files. In this case, you can run the following command to disable the preceding function, or disable dynamic partitioning:

```
set odps.sql.reshuffle.dynamicpt=false;
```

1.5.12.2 Quota and resource usage

Computing resources in MaxCompute may be insufficient sometimes because of improper planning and use of cluster resources.

In general, tasks lacking computing resources have two characteristics, one of which is that the task gets stuck with the output remained at a certain stage. For example, in the following figure, the progress of the M1_Stg1 task has stayed at 0% (because R2_1_Stg1 depends on M1_Stg1, it stays at 0% until M1_Stg1 ends).

```

2016-01-29 13:52:09 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:14 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:19 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:24 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:29 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:34 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:39 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:44 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:49 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:54 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:52:59 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:04 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:09 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:15 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:20 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:25 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:30 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:35 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:40 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:45 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:50 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]
2016-01-29 13:53:55 M1_Stg1_job0:0/0/5[0%] R2_1_Stg1_job0:0/0/1[0%]

```

The other characteristic is that the task remains in "Ready" state in the Logview (as shown in the following figure) (a "Ready" task is awaiting allocation of resources; a "Waiting" task is waiting for completion of the dependent task). The "Ready" state indicates that the resources for running these stand-by task instances are insufficient. Once the instances obtain the necessary resources, they resume operating and change to "Running" state.

M1_Stg1					
Failed(0) Ready(5) All(5) Long-Tails(0) Latency chart					
	FuxiInstanceID	IP & Path	StdOut	StdErr	Status
1	Odps/odps_s...				Ready
2	Odps/odps_s...				Ready
3	Odps/odps_s...				Ready
4	Odps/odps_s...				Ready
5	Odps/odps_s...				Ready

Each task is split into subtasks based on the execution plan and shown in a DAG, and each subtask invokes multiple instances to execute the computation concurrently. In general, the resources required for invoking an instance are

a 1-core CPU and 2 GB of memory. A quota group is assigned to each project for reasonable resource allocation. The quota group determines the maximum amount of resources (CPU and memory) that can be used by all jobs in the project concurrently. Once the resource usage for simultaneously running tasks reaches the limit of the quota group, the tasks are stuck due to insufficient resources.

There are two methods to solve this problem:

- Run the tasks in idle periods.
- Increase the quota group for the project (handled by OAM personnel).

1.5.12.3 MaxCompute storage optimization tips

Partition tables reasonably

MaxCompute supports the concept of partitioning in a table. A partition refers to the specified partition space in the creation of a table; that is, a few fields in the specified table as the partition columns. In most cases, you can consider a partition as a directory in a file system. MaxCompute divides each value of the partition column into a partition (directory). Users can specify multi-level partitions (use multiple fields of the table as partition columns). Multi-level partitions are like multi-level directories. If you specify the name of the partition that you want to access when using the data, then only the corresponding partition are read, avoiding a full table scan. This improves the processing efficiency and reduces costs.

Example of a partitioning statement: `create table src (key string, value bigint) partitioned by (pt string);` In this example, `select * from src where pt='20160901';` specifies the partitioning format. MaxCompute takes only the data in the "20160901" partition as the input when generating a query plan.

Example of a non-partitioning statement: `select * from src where key = 'MaxCompute';` scans the entire table.

Partitioning is usually based on date or geographical region. You may also set partitions based on your business requirements. Example:

```
create table if not exists sale_detail(
  shop_name string,
  customer_id string,
  total_price double)
partitioned by (sale_date string, region string);
```

```
-- Create a two-level partitioned table, in which sale_date is level-1  
partition, and region level-2 partition.
```

Set table lifecycle reasonably

Storage space on MaxCompute is precious. You can set the life cycle of a table according to data usage. MaxCompute will delete expired data to save storage space.

Example: Run the `create table test3 (key boolean) partitioned by (pt string, ds string) lifecycle 100;` command to create a table with a lifecycle of 100. If the latest modification time of this table or partition was more than 100 days ago, the table or partition will be deleted.



Notice:

The lifecycle takes a partition as the smallest unit, so for a partitioned table, if some partitions reach the lifecycle threshold, they will be deleted directly. Partitions that have not reached the lifecycle threshold are not be affected.

Run the `alter table table_name set lifecycle days;` command to modify the lifecycle of an existing table.

Archive cold data

Some data need to be preserved either permanently or for a long period of time, but the frequency of access decreases over time. When the use frequency is very low, you can archive the data. The archive function saves data with RAID. Data is not simply stored as three copies. By using the Cauchy Reed-Solomon algorithm, data is stored as six copies of the original data plus three parity blocks. This improves the effective storage ratio from 1:3 to 1:1.5. In addition, MaxCompute uses the bzip2 algorithm to archive tables with a higher compression ratio than other algorithms. Combining the two algorithms reduces storage usage by more than 70%.

Archiving command format is as below:

```
ALTER TABLE table_name [PARTITION(partition_name='partition_value')]
ARCHIVE;
```

Example:

```
alter table my_log partition(ds='20140101') archive;
```

Merge small files

In the reduce calculation or real-time tunnel data collection, a large number of small files are generated. Too many small files may cause the following problems:

- **Many instances are occupied because a single instance can process only a small number of files. This results in a waste of resources, affecting the overall execution performance.**
- **The file system becomes larger, while the use ratio of disk space becomes smaller.**

Currently, there are two alternative ways to merge small files: ALTER merge mode and SQL merge mode:

- **The ALTER merge mode merges files through 'console' command. The command format is as follows:**

```
ALTER TABLE tablename [PARTITION] MERGE SMALLFILES;
```

- **Set control parameters after SQL execution is complete. Run `odps.task.merge.enabled=true`; to determine whether it is necessary to merge small files. If so, start FuxiJob to merge these files.**

1.5.12.4 UDF OOM error

Some jobs will report the OOM error during running. The error message is as follows:

```
FAILED: ODPS-0123144: Fuxi job failed - WorkerRestart errCode:9,errMsg
:SigKill(OOM), usually caused by OOM(out of memory)
```

This problem can be solved by setting the UDF runtime parameters:

```
odps.sql.mapper.memory=3072;
set odps.sql.udf.jvm.memory=2048;
```

```
set odps.sql.udf.python.memory=1536;
```

1.5.13 Common MaxCompute SQL parameter settings

1.5.13.1 MAP configurations

```
set odps.sql.mapper.cpu=100
```

Purpose: It is used to set the number of CPUs for each instance in a Map task.

Default value: 100. **Value range:** 50 to 800.

```
set odps.sql.mapper.memory=1024
```

Purpose: It is used to set the memory size for each instance in a Map task. **Default value:** 1024 MB. **Value range:** 256 MB to 12,288 MB.

```
set odps.sql.mapper.merge.limit.size=64
```

Purpose: It is used to set the maximum size of control files to be merged. **Default value:** 64 MB. You can set this variable to control the inputs of mappers. **Value range:** 0 to Integer.MAX_VALUE.

```
set odps.sql.mapper.split.size=256
```

Purpose: It is used to set the maximum data input volume for a map. **Default value:** 256 MB. You can set this variable to control the inputs of mappers. **Value range:** 1 to Integer.MAX_VALUE.

1.5.13.2 JOIN configurations

```
set odps.sql.joiner.instances=-1
```

Purpose: It is used to set the number of instances in a JOIN task. **Default value:** 1. **Value range:** 0 to 2,000.

```
set odps.sql.joiner.cpu=100
```

Purpose: It is used to set the number of CPUs for each instance in a JOIN task. **Default value:** 100. **Value range:** 50 to 800.

```
set odps.sql.joiner.memory=1024
```

Purpose: It is used to set the memory size for each instance in a JOIN task. **Default value:** 1,024 MB. **Value range:** 256 MB to 12,288 MB.

1.5.13.3 Reduce configurations

```
set odps.sql.reducer.instances=-1
```

Purpose: It is used to set the number of instances in a Reduce task. Default value: 1.
Value range: 0 to 2,000.

```
set odps.sql.reducer.cpu=100
```

Purpose: It is used to set the number of CPUs for each instance in a Reduce task.
Default value: 100. **Value range:** 50 to 800.

```
set odps.sql.reducer.memory=1024
```

Purpose: It is used to set the memory size for each instance in a Reduce task.
Default value: 1,024 MB. **Value range:** 256 to 12,288 MB.

1.5.13.4 UDF configurations

```
set odps.sql.udf.jvm.memory=1024
```

Purpose: It is used to set the maximum memory size for a UDF JVM heap. Default value: 1,024 MB. **Value range:** 256 to 12,288 MB.

```
set odps.sql.udf.timeout=600
```

Purpose: It is used to set the timeout value of a UDF. Default value: 600 seconds.
Value range: 0 to 3,600 seconds.

```
set odps.sql.udf.python.memory=256
```

Purpose: It is used to set the maximum memory size for UDF python. Default value: 256 MB. **Value range:** 64 to 3,072 MB.

```
set odps.sql.udf.optimize.reuse=true/false
```

Purpose: after start-up, each UDF function expression can only be calculated once, improving performance. The default is true.

```
set odps.sql.udf.strict.mode=false/true
```

Purpose: It is used to control functions regarding whether to return NULL or error if dirty data is encountered. If it is true, an error is returned. If it is false, NULL is returned.

1.5.13.5 MAPJOIN configurations

```
set odps.sql.mapjoin.memory.max=512
```

Purpose: It is used to set the maximum memory of a small table in MAPJOIN.

Default value 512 MB. Value range: 128 to 2,048 MB.

```
set odps.sql.reshuffle.dynamicpt=true/false
```

Purpose:

- Some scenarios of dynamic partitioning are time-consuming. Shutting them down can speed up SQL.
- If the dynamic partition value is very small, disabling dynamic partition can avoid data skew.

1.5.13.6 Configure data skew

```
set odps.sql.groupby.skewindata=true/false
```

Effect: enables the group by optimization.

```
set odps.sql.skewjoin=true/false
```

Effect: enables the join optimization. It takes effect only when `odps.sql.skewinfo` is configured.

```
set odps.sql.skewinfo
```

Purpose: It is used to set detailed information of join optimization. The command syntax is as follows:

```
set odps.sql.skewinfo=skewed_src:(skewed_key) [("skewed_value")]
```

Example:

The following command is used to set a single skewed data value in a single field:

```
set odps.sql.skewinfo=src_skewjoin1:(key) [("0")]  
-- Command output: explain select a.key c1, a.value c2, b.key c3, b.  
value c4 from src a join src_skewjoin1 b on a.key = b.key;
```

The following command is used to set multiple skewed data values in a single field:

```
set odps.sql.skewinfo=src_skewjoin1:(key) [("0") ("1")]
```



```
-- Command output: explain select a.key c1, a.value c2, b.key c3, b.  
value c4 from src a join src_skewjoin1 b on a.key = b.key;
```

1.5.14 MapReduce-to-SQL conversion for execution

1.5.14.1 Overview

MaxCompute provides a series of Java APIs for MapReduce to process data.

In MaxCompute 2.0, MapReduce programs are automatically converted to SQL for execution. After the conversion, you can use MaxCompute 2.0 compiler, optimizer, and execution engine to process the MapReduce programs. The new features of the SQL engine can also be used in MaxCompute 2.0. The features, performance, and stability of SQL engine are optimized.



Notice:

- No changes to the original APIs and job logics are required.
- Only MapReduce jobs of the OpenMr job type, which are written with MapReduce APIs, can be converted to SQL.

1.5.14.2 Local running settings

1. Download the latest [MaxCompute client](#) package to the local PC and make proper configurations.
2. Set the execution mode.

You can change the execution mode to better suit your business needs. The default execution mode is the lot mode. In the lot mode, jobs are executed by MapReduce. The MaxCompute 2.0 compiler, optimizer, and execution engine are not utilized.

You can enable the conversion flag by changing `odps.mr.run.mode`. Valid values: lot, sql, and hybrid.

- The first method is to enable the conversion flag at the project level. Because this method affects all jobs, it requires a project administrator to apply for it. Set the value of `odps.mr.run.mode` to hybrid or sql. In the hybrid mode, if

SQL execution fails, the job will be executed by MapReduce. In the SQL mode, when SQL execution fails, an error is returned.

- The second method is to enable the conversion flag at the session level and is only valid for the current job. The following two ways can be used:
 - Add the SET statement. Example: `set odps.mr.run.mode=hybrid`.
 - Configure the job parameter as follows:

```
JobConf job = new JobConf();  
job.set("odps.mr.run.mode", "hybrid")
```

The conversion flag will be enabled at the project level later by MaxCompute O&M personnel.

1.5.14.3 DataWorks running settings

Jobs running in DataWorks are updated by the O&M personnel of MaxCompute and DataWorks. There is no need to update the client manually.

1. Enable the conversion flag for a single job.

You can add the SET statement before a MapReduce job or configure the job parameter for it. This method takes effect at the session level and only influences the execution of the current job.

You can use either of the following methods to enable the conversion flag at the session level:

- Add the SET statement. Example: `set odps.mr.run.mode=hybrid`.
- Configure the job parameter as follows:

```
JobConf job = new JobConf();  
job.set("odps.mr.run.mode", "hybrid")
```

2. Enable the conversion flag at the project level. Set `odps.mr.run.mode` for a project. For more information, see [Local running settings](#).

1.5.14.4 View running details

You can use Logview and MaxCompute Studio to view MapReduce-to-SQL conversion results and running details of SQL jobs.

1. LogView XML.

Open Logview and click the LOT node in the center of the page. The SQL jobs that are converted from MapReduce jobs are included in the XML information of the node. Example:

```
create temporary function mr2sql_mapper_152955927079392291755 as '
com.aliyun.odps.mapred.bridge.LotMapperUDTF' using ;
create temporary function mr2sql_reducer_152955927079392291755 as '
com.aliyun.odps.mapred.bridge.LotReducerUDTF' using ;
@sub_query_mapper :=
SELECT k_id,v_gmt_create,v_gmt_modified,v_product_id,v_admin_seq,
v_sku_attr,v_sku_price,v_sku_stock,v_sku_code,v_sku_image,v_delivery
_time,v_sku_bulk_order,v_sku_bulk_discount,v_sku_image_version,
v_currency_code
FROM(
SELECT mr2sql_mapper_152955927079392291755(id,gmt_create,gmt_modifi
ed,product_id,admin_seq,sku_attr,sku_price,sku_stock,sku_code,
sku_image,delivery_time,sku_bulk_order,sku_bulk_discount,sku_image_
version,currency_code ) as (k_id,v_gmt_create,v_gmt_modified
,v_product_id,v_admin_seq,v_sku_attr,v_sku_price,v_sku_stock,
v_sku_code,v_sku_image,v_delivery_time,v_sku_bulk_order,v_sku_bulk
_discount,v_sku_image_version,v_currency_code)
FROM ae_antispam.product_sku_tt_inc
WHERE ds = "20180615" AND hh = "21"
UNION ALL
SELECT mr2sql_mapper_152955927079392291755(id,gmt_create,gmt_modifi
ed,product_id,admin_seq,sku_attr,sku_price,sku_stock,sku_code,
sku_image,delivery_time,sku_bulk_order,sku_bulk_discount,sku_image_
version,currency_code ) as (k_id,v_gmt_create,v_gmt_modified
,v_product_id,v_admin_seq,v_sku_attr,v_sku_price,v_sku_stock,
v_sku_code,v_sku_image,v_delivery_time,v_sku_bulk_order,v_sku_bulk
_discount,v_sku_image_version,v_currency_code)
FROM ae_antispam.product_sku
) open_mr_alias1
DISTRIBUTE BY k_id SORT BY k_id ASC;
@sub_query_reducer :=
SELECT mr2sql_reducer_152955927079392291755(k_id,v_gmt_create,
v_gmt_modified,v_product_id,v_admin_seq,v_sku_attr,v_sku_price,
v_sku_stock,v_sku_code,v_sku_image,v_delivery_time,v_sku_bulk_order
,v_sku_bulk_discount,v_sku_image_version,v_currency_code) as (id
,gmt_create,gmt_modified,product_id,admin_seq,sku_attr,sku_price,
sku_stock,sku_code,sku_image,delivery_time,sku_bulk_order,sku_bulk_d
iscount,sku_image_version,currency_code)
FROM @sub_query_mapper;
FROM @sub_query_reducer
INSERT OVERWRITE TABLE ae_antispam.product_sku
SELECT id,gmt_create,gmt_modified,product_id,admin_seq,sku_attr,
sku_price,sku_stock,sku_code,sku_image,delivery_time,sku_bulk_order,
sku_bulk_discount,sku_image_version,currency_code ;
```

2. LogView detail or summary.

You can see that the new execution engine is used to execute jobs.

```
Job run mode: fuxi job
```

Job run engine: execution engine

3. LogView detail or JSON summary.

The JSON summary information in MapReduce only contains the input and output information of Map and Reduce. However, the JSON summary information in SQL allows you to view details about each stage of SQL execution, such as all execution parameters, logical execution plans, physical execution plans, and execution details. Example:

```
"midlots" :
[
  "LogicalTableSink(table=[[odps_flighting.flt_20180621104445_s
tepl_ad_quality_tech_qp_algo_antifake_wordbag_filter_bag_cha
nge_result_lv2_20, auctionid,word,match_word(3) {0, 1, 2}]]
OdpsLogicalProject(auctionid=[0], word=[1], match_word=[2])
OdpsLogicalProject(auctionid=[0], word=[1], match_word=[2])
OdpsLogicalProject(auctionid=[0], word=[1], match_word=[2])
OdpsLogicalProject(auctionid=[2], word=[3], match_word=[4])
OdpsLogicalTableFunctionScan(invocation=[[MR2SQL_MAPPER_152955
294118813063732($0, $1)]()], rowType=[RecordType(VARCHAR(2147483647
) item_id, VARCHAR(2147483647) text, VARCHAR(2147483647) __tf_0_0,
VARCHAR(2147483647) __tf_0_1, VARCHAR(2147483647) __tf_0_2)])
OdpsLogicalTableScan(table=[[ad_quality_tech_qp_algo_antifake_wor
dbag_filter_bag_change_lv2_20, item_id,text(2) {0, 1}]]
]
```

1.5.15 Appendix

1.5.15.1 Escape character

String constants in MaxCompute SQL can be enclosed in single or double quotation marks, in double quotation marks enclosed in single quotation marks, or in single quotation marks enclosed in double quotation marks. Otherwise, they must be expressed with an escape character. Examples of correct expressions: "I'm a happy coder!" and 'I\'m a happy coder!'.

In MaxCompute SQL, the backslash (\) is an escape character, which expresses the special character in a string or interprets the character that follows as the character itself. When a string constant is read, if the backslash is followed by three valid octal digits in the range from 001 to 177, the system converts the ASCII values into the corresponding characters. The following table lists the mappings between escape sequences and represented characters.

Table 1-30: Escape sequences

Escape sequence	Represented character
<code>\b</code>	Backspace
<code>\t</code>	Tab
<code>\n</code>	Newline
<code>\r</code>	Carriage return
<code>\'</code>	Single quote
<code>\"</code>	Double quote
<code>\\</code>	Backslash
<code>\;</code>	Semicolon
<code>\Z</code>	Control-Z
<code>\0</code> or <code>\00</code>	Terminator

Example:

```
select length('a\tb') from dual;  
-- The result is 3, indicating that the string contains three  
characters, with "\t" regarded as one character. Any character  
following the escape sequence is interpreted as the character itself.
```

```
select 'a\ab',length('a\ab') from dual;  
-- The result is 'aab', with a length of 3. "\a" is interpreted as an  
ordinary "a".
```

1.5.15.2 LIKE matching

In LIKE matching, "%" indicates matching any number of characters; "_" indicates matching a single character. If the character "%" or "_" needs to be matched, escape conversion is required. "\\%" indicates matching "%", and "_" indicates matching "_".

**Note:**

For the character set of strings, MaxCompute SQL currently supports the UTF-8 character set. Data that is encoded in a different format may result in incorrect calculations.

1.5.15.3 Regular expressions

MaxCompute SQL adopts the PCRE library for regular expressions. Matching is performed character by character. The supported metacharacters are as follows:

- **^**: the beginning of a row
- **\$**: the end of a row
- **.**: any character
- *****: matches zero or multiple times.
- **+**: matches once or multiple times.
- **?**: matches a modifier. If this character follows any one of other delimiters (*****, **+**, **?**, **{n}**, **{n,}**, or **{n,m}**), the match is lazy. In the lazy mode, as few strings as possible are matched. In the default greedy mode, as many searched strings as possible are matched zero times or once.
- **A|B**: A or B
- **(abc)***: matches the abc sequence zero or multiple times.
- **{n}** or **{m,n}**: the number of matches
- **[ab]**: matches any character in the brackets.
- **[^ab]**: **^** represents NOT. This metacharacter matches any character that is neither a nor b.
- ****: the escape sequence
- **\n**: n represents digit 1 to 9. This metacharacter specifies backward reference.
- **\d**: digit
- **\D**: non-digit

- **[::]: POSIX character set**
 - **[[:alnum:]]:** letter or digit in the range of [a-zA-Z0-9]
 - **[[:alpha:]]:** letter in the range of [a-zA-Z]
 - **[[:ascii:]]:** ASCII character in the range of [\x00-\x7F]
 - **[[:blank:]]:** space and tab in the range of [\t]
 - **[[:cntrl:]]:** control character in the range of [\x00-\x1F\x7F]
 - **[[:digit:]]:** digit in the range of [0-9]
 - **[[:graph:]]:** any character except space in the range of [\x21-\x7E]
 - **[[:space:]]:** space in the range of [\t\r\n\v\f]
 - **[[:print:]]:** [[:graph:]] and [[:space:]] in the range of [\x20-\x7E]
 - **[[:lower:]]:** lowercase letter in the range of [a-z]
 - **[[:punct:]]:** punctuation in the range of [!\"#\$%&()*+,-./:;<=>? @\^_`{}~|]
 - **[[:upper:]]:** uppercase letter in the range of [A-Z]
 - **[[:xdigit:]]:** hexadecimal character in the range of [A-Fa-f0-9]

The system uses a backslash (\) as the escape character, so a backslash (\) in a regular expression indicates second escape. For example, the string to be matched by the regular expression is "a+b". The plus sign (+) is a special character in regex, and must be escaped to obtain the string "a+b". However, the system needs to escape the first backslash (escape character) before it can be read by regex. Hence, the expression to match "a+b" is "a\\+b".

The following example assumes that there is a table named test_dual:

```
select 'a+b' rlike 'a\\+b' from test_dual;
+-----+
_c1 |
+-----+
true |
+-----+
```

In extreme cases, to match the character "\", which is a special character in the regular engine, the expression must be "\\\". The system must perform an escape on the expression, so it is expressed as "\\\".

```
select 'a\\b', 'a\\b' rlike 'a\\\\b' from test_dual;
+-----+-----+
_c0 | _c1 |
+-----+-----+
a\b | true |
```

+-----+-----+

**Note:**

- If a MaxCompute SQL statement contains "a\b", 'a\b' is displayed in the output because MaxCompute escapes the expression.
- If a string contains a tab or tab character, the system reads '\t' and stores it as one character. Therefore, it is a common character in the regular expression mode.

```
select 'a\tb', 'a\tb' rlike 'a\tb' from test_dual;
+-----+-----+
_c0 | _c1 |
+-----+-----+
a b | true |
+-----+-----+
```

1.5.15.4 Reserved words

The following are all reserved words in MaxCompute SQL. Do not use these words to name tables, columns, or partitions. Otherwise, an error is returned. Reserved words are case-insensitive.

```
% & && ( ) * +. / ; < <= <> = > >= ? ADD AFTER ALL ALTER ANALYZE AND
ARCHIVE ARRAY AS ASC BEFORE BETWEEN BIGINT BINARY BLOB BOOLEAN BOTH
BUCKET BUCKETS BY CASCADE CASE CAST CFILE CHANGE CLUSTER CLUSTERED
CLUSTERSTATUS COLLECTION COLUMN COLUMNS COMMENT COMPUTE CONCATENAT
E CONTINUE CREATE CROSS CURRENT CURSOR DATA DATABASE DATABASES
DATE DATETIME DBPROPERTIES DEFERRED DELETE DELIMITED DESC DESCRIBE
DIRECTORY DISABLE DISTINCT DISTRIBUTE DOUBLE DROP ELSE ENABLE END
ESCAPED EXCLUSIVE EXISTS EXPLAIN EXPORT EXTENDED EXTERNAL FALSE FETCH
FIELDS FILEFORMAT FIRST FLOAT FOLLOWING FORMAT FORMATTED FROM FULL
FUNCTION FUNCTIONS GRANT GROUP HAVING HOLD_DDLTIME IDXPROPERTIES IF
IMPORT IN INDEX INDEXES INPATH INPUTDRIVER INPUTFORMAT INSERT INT
INTERSECT INTO IS ITEMS JOIN KEYS LATERAL LEFT LIFECYCLE LIKE LIMIT
LINES LOAD LOCAL LOCATION LOCK LOCKS LONG MAP MAPJOIN MATERIALIZ
ED MINUS MSCK NOT NO_DROP NULL OF OFFLINE ON OPTION OR ORDER OUT
OUTER OUTPUTDRIVER OUTPUTFORMAT OVER OVERWRITE PARTITION PARTITIONE
D PARTITIONPROPERTIES PARTITIONS PERCENT PLUS PRECEDING PRESERVE
PROCEDURE PURGE RANGE RCFILE READ READONLY READS REBUILD RECORDREAD
ER RECORDWRITER REDUCE REGEXP RENAME REPAIR REPLACE RESTRICT REVOKE
RIGHT RLIKE ROW ROWS SCHEMA SCHEMAS SELECT SEMI SEQUENCEFILE SERDE
SERDEPROPERTIES SET SHARED SHOW SHOW_DATABASE SMALLINT SORT SORTED SSL
STATISTICS STORED STREAMTABLE STRING STRUCT TABLE TABLESAMPLE
TBLPROPERTIES TEMPORARY TERMINATED TEXTFILE THEN TIMESTAMP TINYINT TO
TOUCH TRANSFORM TRIGGER TRUE UNARCHIVE UNBOUNDED UNDO UNION UNIONTYPE
```


UNIQUEJOIN UNLOCK UNSIGNED UPDATE USE USING UTC UTC_Timestamp VIEW WHEN
WHERE WHILE

1.6 MaxCompute Tunnel

1.6.1 Tunnel SDK overview

1.6.1.1 Overview

MaxCompute tools for data upload and download are compiled based on the tunnel SDK. This topic describes main tunnel SDK APIs. The SDK usage may vary based on the Java SDK version. For specific information, see Java SDK documents.

The following table lists the major APIs.

Table 1-31: Major APIs

API	Description
TableTunnel	The entry class of the MaxCompute Tunnel service.
TableTunnel.UploadSession	The session that uploads data to a MaxCompute table.
TableTunnel.DownloadSession	The session that downloads data from a MaxCompute table.



Notice:

The tunnel endpoint supports automatic routing based on the MaxCompute endpoint settings.

1.6.1.2 TableTunnel

This topic describes the TableTunnel API.

API definition:

```
public class TableTunnel {
    public DownloadSession createDownloadSession(String projectName,
        String tableName);
    public DownloadSession createDownloadSession(String projectName,
        String tableName, PartitionSpec partitionSpec);
    public UploadSession createUploadSession(String projectName, String
        tableName);
    public UploadSession createUploadSession(String projectName, String
        tableName, PartitionSpec partitionSpec);
    public DownloadSession getDownloadSession(String projectName, String
        tableName, PartitionSpec partitionSpec, String id);
    public DownloadSession getDownloadSession(String projectName, String
        tableName, String id);
}
```

```
public UploadSession getUploadSession(String projectName, String
tableName, PartitionSpec partitionSpec, String id);
public UploadSession getUploadSession(String projectName, String
tableName, String id); public void setEndpoint(String endpoint);
}
```

TableTunnel API description:

- **Lifecycle:** From the creation of the TableTunnel instance to the end of the program.
- **Purpose:** It provides a method to create Upload and Download objects.

1.6.1.3 UploadSession

This topic describes the UploadSession API.

API definition:

```
public class UploadSession {
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec) throws TunnelException;
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec, String uploadId) throws TunnelException;
public void commit(Long[] blocks); public Long[] getBlockList();
public String getId();
public TableSchema getSchema();
public UploadSession.Status getStatus(); public Record newRecord();
public RecordWriter openRecordWriter(long blockId);
public RecordWriter openRecordWriter(long blockId, boolean compress);
}
```

UploadSession API description.

Table 1-32: UploadSession API

Item	Description
Lifecycle	From the upload instance creation to the end of the uploading.
Purpose	<p>Creates an upload instance by calling a constructor method or by using the TableTunnel class.</p> <ul style="list-style-type: none">• Request mode: synchronous.• The server creates an upload session and generates a unique upload ID. You can get the upload ID by running getId on the console.

Item	Description
Upload data	<ul style="list-style-type: none">• Request mode: asynchronous.• Call <code>openRecordWriter</code> to generate a <code>RecordWriter</code> instance . The <code>blockId</code> parameter identifies the data to upload this time and the position of the data in the table. The value range is [0, 20000]. In case the uploading fails, the data is re-uploaded based on the block ID.
Check uploading	<ul style="list-style-type: none">• Request mode: synchronous.• Call <code>getStatus</code> to get the uploading status.• Call <code>getBlockList</code> to get a list of the block IDs of successful uploading instances, check the block ID list, and re-upload data for failed uploading instances.
Stop uploading	<ul style="list-style-type: none">• Request mode: synchronous.• Call <code>commit(Long[] blocks)</code>. The <code>blocks</code> parameter indicates the list of block IDs of successful uploading instances. The server will verify the block ID list.• The verification improves data correctness. If the provided block list is different from the block list on the server, an error is reported.
Status	<ul style="list-style-type: none">• UNKNOWN: Initial value set while server just creates a session.• NORMAL: An <code>UPLOAD</code> object is created successfully.• CLOSING: The server sets the upload session to <code>CLOSING</code> status before calling the <code>COMPLETE</code> method (to complete uploading).• CLOSED: The uploading is completed (data has been moved to the directory where the result table is).• EXPIRED: The upload session is timed out.• CRITICAL: An error occurs.

**Notice:**

- `blockId` in the same `UploadSession` API must be unique. That is, after a block ID is used to start `RecordWriter` in an upload session, data is written, and the session is closed and committed, this block ID cannot be used to start another `RecordWriter`.

- The maximum size of a block is 100 GB. We strongly recommend that 64 MB or more data is written into each block. Otherwise, the computing performance will seriously degrade.
- Each session has a 24-hour life cycle on the server.
- You are advised to have data prepared before calling `openRecordWriter`. A network action is triggered every time the Writer writes 8 KB data. If no network action is triggered in the last 120 seconds, the server closes the connection and the Writer becomes unavailable. You have to start a new Writer.

1.6.1.4 DownloadSession

This topic describes the `DownloadSession` class.

API definition:

```
public class DownloadSession {
    DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec) throws TunnelException
    DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec, String downloadId) throws TunnelExce
ption
    public String getId()
    public long getRecordCount() public TableSchema getSchema()
    public DownloadSession.Status getStatus()
    public RecordReader openRecordReader(long start, long count)
    public RecordReader openRecordReader(long start, long count, boolean
compress)
}
```

DownloadSession API description.

Table 1-33: DownloadSession API

Parameter	Description
Lifecycle	From the creation of the Download instance to the end of the download process.
Purpose	<p>Creates a Download instance by calling a constructor method or using <code>TableTunnel</code>.</p> <ul style="list-style-type: none">• Request mode: Synchronous.• The server creates a session for this Download and generates a unique download ID to mark the Download. The console can get data with a get ID. The operation has a high overhead. The server creates indexes for the data files. If many data files exist, the operation takes a long time. Then the server returns the total number of records, and starts concurrent downloads according to the number of records.

Parameter	Description
Download data	<ul style="list-style-type: none">• Request mode: Asynchronous.• Call openRecordReader to generate a RecordReader instance. The Start parameter marks the start position of record for this download. The value of Start is equivalent to or greater than 0. The Count parameter marks the number of records for this download. The value of Count is greater than 0.
View the download process	<ul style="list-style-type: none">• Request mode: Synchronous.• Call getStatus to get the download status.
Status	<ul style="list-style-type: none">• UNKNOWN: the initial value that is set when the server creates a session.• NORMAL: The download object is successfully created.• CLOSED: The download session is completed.• EXPIRED: The download session times out.

1.6.2 Tunnel SDK example

1.6.2.1 Simple upload example

This topic provides a simple upload example of Tunnel SDK.

Example:

```
import java.io.IOException;
import java.util.Date;
import com.aliyun.odps.Column;
import com.aliyun.odps.Odps;
import com.aliyun.odps.PartitionSpec;
import com.aliyun.odps.TableSchema;
import com.aliyun.odps.account.Account;
import com.aliyun.odps.account.AliyunAccount;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.RecordWriter;
import com.aliyun.odps.tunnel.TableTunnel;
import com.aliyun.odps.tunnel.TunnelException;
import com.aliyun.odps.tunnel.TableTunnel.UploadSession;
public class UploadSample {
    private static String accessId = "<your access id>";
    private static String accessKey = "<your access Key>";
    private static String tunnelUrl = "<your tunnel endpoint>";
    private static String odpsUrl = "<your odps endpoint>";
    private static String project = "<your project>";
    private static String table = "<your table name>";
    private static String partition = "<your partition spec>";
    public static void main(String args[]) {
        Account account = new AliyunAccount(accessId, accessKey);
        Odps odps = new Odps(account);
        odps.setEndpoint(odpsUrl);
        odps.setDefaultProject(project);
```

```
try {
    TableTunnel tunnel = new TableTunnel(odps);
    tunnel.setEndpoint(tunnelUrl);
    PartitionSpec partitionSpec = new PartitionSpec(partition);
    UploadSession uploadSession = tunnel.createUploadSession(project
,
        table, partitionSpec);
    System.out.println("Session Status is : "
        + uploadSession.getStatus().toString());
    TableSchema schema = uploadSession.getSchema();
    // After data is prepared, run the Writer command to start
writing data. The prepared data is written to a block.
    // Writing a small volume of data to each block can result in a
large number of small files.This greatly reduces computing performance
. We strongly recommend that you write at least 64 MB (and up to 100
GB) of data to each block.
    // You can estimate the total data volume based on the average
data volume and record count. For example, 64 MB < Average data volume
x Record count < 100 GB.
    RecordWriter recordWriter = uploadSession.openRecordWriter(0);
    Record record = uploadSession.newRecord();
    for (int i = 0; i < schema.getColumns().size(); i++) {
        Column column = schema.getColumn(i);
        switch (column.getType()) {
            case BIGINT:
                record.setBigint(i, 1L);
                break;
            case BOOLEAN:
                record.setBoolean(i, true);
                break;
            case DATETIME:
                record.setDatetime(i, new Date());
                break;
            case DOUBLE:
                record.setDouble(i, 0.0);
                break;
            case STRING:
                record.setString(i, "sample");
                break;
            default:
                throw new RuntimeException("Unknown column type: "
                    + column.getType());
        }
    }
    for (int i = 0; i < 10; i++) {
        // Write data to the server. A network transmission process is
triggered each time 8 KB of data is written.
        // If no data is transmitted for 120 seconds, the connection
times out. The Writer command becomes unavailable, and you must write
data again.
        recordWriter.write(record);
    }
    recordWriter.close();
    uploadSession.commit(new Long[]{0L});
    System.out.println("upload success!");
} catch (TunnelException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

}

1.6.2.2 Simple download example

This topic provides an example for the simple download function of Tunnel SDK.

Example:

```
import java.io.IOException; import java.util.Date;
import com.aliyun.odps.Column; import com.aliyun.odps.Odps;
import com.aliyun.odps.PartitionSpec; import com.aliyun.odps.
TableSchema; import com.aliyun.odps.account.Account;
import com.aliyun.odps.account.AliyunAccount; import com.aliyun.odps.
data.Record;
import com.aliyun.odps.data.RecordReader; import com.aliyun.odps.
tunnel.TableTunnel;
import com.aliyun.odps.tunnel.TableTunnel.DownloadSession; import com.
aliyun.odps.tunnel.TunnelException;
public class DownloadSample {
    private static String accessId = "<your access id>"; private static
    String accessKey = "<your access Key>";
    private static String tunnelUrl = "<your tunnel endpoint>";
    private static String odpsUrl = "<your odps endpoint>";
    private static String project = "<your project>"; private static
    String table = "<your table name>";
    private static String partition = "<your partition spec>";
    public static void main(String args[]) {
        Account account = new AliyunAccount(accessId, accessKey); Odps odps =
        new Odps(account); odps.setEndpoint(odpsUrl);
        odps.setDefaultProject(project);
        TableTunnel tunnel = new TableTunnel(odps); tunnel.setEndpoint(
        tunnelUrl);
        PartitionSpec partitionSpec = new PartitionSpec(partition); try {
        DownloadSession downloadSession = tunnel.createDownloadSession(project
        , table, partitionSpec);
        System.out.println("Session Status is : "
        + downloadSession.getStatus().toString());
        long count = downloadSession.getRecordCount(); System.out.println("
        RecordCount is: " + count);
        RecordReader recordReader = downloadSession.openRecordReader(0, count
        );
        Record record;
        while ((record = recordReader.read()) != null) { consumeRecord(record
        , downloadSession.getSchema());
        }
        recordReader.close();
        } catch (TunnelException e) { e.printStackTrace();
        } catch (IOException e1) { e1.printStackTrace();
        }
        }
        private static void consumeRecord(Record record, TableSchema schema)
        { for (int i = 0; i < schema.getColumns().size(); i++) {
        Column column = schema.getColumn(i); String colValue = null;
        switch (column.getType()) { case BIGINT: {
        Long v = record.getBigint(i);
        colValue = v == null ? null : v.toString(); break;
        }
        case BOOLEAN: {
        Boolean v = record.getBoolean(i); colValue = v == null ? null : v.
        toString(); break;
        }
        case DATETIME: {
```

```

Date v = record.getDatetime(i); colValue = v == null ? null : v.
toString(); break;
}
case DOUBLE: {
Double v = record.getDouble(i); colValue = v == null ? null : v.
toString(); break;
}
case STRING: {
String v = record.getString(i);
colValue = v == null ? null : v.toString(); break;
}
default:
throw new RuntimeException("Unknown column type: "
+ column.getType());
}
System.out.print(colValue == null ? "null" : colValue); if (i !=
schema.getColumns().size())
System.out.print("\t");
}
System.out.println();
}
}
}

```

1.6.2.3 Multithread upload example

This topic provides a multithread upload example of Tunnel SDK.

Example:

```

import java.io.IOException;
import java.util.ArrayList;
import java.util.Date;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import com.aliyun.odps.Column;
import com.aliyun.odps.Odps;
import com.aliyun.odps.PartitionSpec;
import com.aliyun.odps.TableSchema;
import com.aliyun.odps.account.Account;
import com.aliyun.odps.account.AliyunAccount;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.RecordWriter;
import com.aliyun.odps.tunnel.TableTunnel;
import com.aliyun.odps.tunnel.TunnelException;
import com.aliyun.odps.tunnel.TableTunnel.UploadSession;
class UploadThread implements Callable<Boolean> {
    private long id;
    private RecordWriter recordWriter;
    private Record record;
    private TableSchema tableSchema;
    public UploadThread(long id, RecordWriter recordWriter, Record
record, TableSchema tableSchema) {
        this.id = id;
        this.recordWriter = recordWriter;
        this.record = record;
        this.tableSchema = tableSchema;
    }
    @Override
    public Boolean call() {
        for (int i = 0; i < tableSchema.getColumns().size(); i++) {
            Column column = tableSchema.getColumn(i);

```



```

switch (column.getType()) {
    case BIGINT:
        record.setBigint(i, 1L);
        break;
    case BOOLEAN:
        record.setBoolean(i, true);
        break;
    case DATETIME:
        record.setDatetime(i, new Date());
        break;
    case DOUBLE:
        record.setDouble(i, 0.0);
        break;
    case STRING:
        record.setString(i, "sample");
        break;
    default:
        throw new RuntimeException("Unknown column type: "
            + column.getType());
}
}
try {
    for (int i = 0; i < 10; i++) {
        // Write data to the server. A network transmission process is
        // triggered each time 8 KB of data is written.
        // If no data is transmitted for 120 seconds, the connection
        // times out. The Writer command becomes unavailable and you must write
        // data again.
        recordWriter.write(record);
    }
    recordWriter.close();
} catch (IOException e) {
    e.printStackTrace();
    return false;
}
return true;
}
}
public class UploadThreadSample {
    private static String accessId = "<your access id>";
    private static String accessKey = "<your access Key>";
    private static String tunnelUrl = "<your tunnel endpoint>";
    private static String odpsUrl = "<your odps endpoint>";
    private static String project = "<your project>";
    private static String table = "<your table name>";
    private static String partition = "<your partition spec>";
    private static int threadNum = 10;
    public static void main(String args[]) {
        Account account = new AliyunAccount(accessId, accessKey);
        Odps odps = new Odps(account);
        odps.setEndpoint(odpsUrl);
        odps.setDefaultProject(project);
        try {
            TableTunnel tunnel = new TableTunnel(odps);
            tunnel.setEndpoint(tunnelUrl);
            PartitionSpec partitionSpec = new PartitionSpec(partition);
            UploadSession uploadSession = tunnel.createUploadSession(project
,
            table, partitionSpec);
            System.out.println("Session Status is : "
                + uploadSession.getStatus().toString());
            ExecutorService pool = Executors.newFixedThreadPool(threadNum);
            ArrayList<Callable<Boolean>> callers = new ArrayList<Callable<
Boolean>>();

```

```

        // After the data is prepared, open a writer to start multithrea
d data writing.
        // Writing a small volume of data to each block can result in a
large number of small files. This greatly affects computing performanc
e. We recommend that you write at least 64 MB (and up to 100 GB) of
data to each block.
        // You can estimate the total data volume based on the average
data volume and record count. For example, 64 MB < Average data volume
x Record count < 100 GB.
        for (int i = 0; i < threadNum; i++) {
            RecordWriter recordWriter = uploadSession.openRecordWriter(i);
            Record record = uploadSession.newRecord();
            callers.add(new UploadThread(i, recordWriter, record,
uploadSession.getSchema()));
        }
        pool.invokeAll(callers);
        pool.shutdown();
        Long[] blockList = new Long[threadNum];
        for (int i = 0; i < threadNum; i++)
            blockList[i] = Long.valueOf(i);
        uploadSession.commit(blockList);
        System.out.println("upload success!");
    } catch (TunnelException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
}

```

1.6.2.4 Multithread download example

This topic provides a multithread download example of Tunnel SDK.

Example:

```

import java.io.IOException;
import java.util.ArrayList; import java.util.Date; import java.util.
List;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutionException; import java.util.
concurrent.ExecutorService; import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import com.aliyun.odps.Column; import com.aliyun.odps.Odps;
import com.aliyun.odps.PartitionSpec; import com.aliyun.odps.
TableSchema; import com.aliyun.odps.account.Account;
import com.aliyun.odps.account.AliyunAccount; import com.aliyun.odps.
data.Record;
import com.aliyun.odps.data.RecordReader; import com.aliyun.odps.
tunnel.TableTunnel;
import com.aliyun.odps.tunnel.TableTunnel.DownloadSession; import com.
aliyun.odps.tunnel.TunnelException;
class DownloadThread implements Callable<Long> { private long id;
private RecordReader recordReader; private TableSchema tableSchema;
public DownloadThread(int id,
RecordReader recordReader, TableSchema tableSchema) { this.id = id;
this.recordReader = recordReader; this.tableSchema = tableSchema;
}
@Override
public Long call() {

```

```
Long recordNum = 0L; try {
    Record record;
    while ((record = recordReader.read()) != null) { recordNum++;
        System.out.print("Thread " + id + "\t"); consumeRecord(record,
            tableSchema);
    }
    recordReader.close();
} catch (IOException e) { e.printStackTrace();
}
return recordNum;
}

private static void consumeRecord(Record record, TableSchema schema)
{ for (int i = 0; i < schema.getColumns().size(); i++) {
    Column column = schema.getColumn(i); String colValue = null;
    switch (column.getType()) { case BIGINT: {
        Long v = record.getBigint(i);
        colValue = v == null ? null : v.toString(); break;
    }
    case BOOLEAN: {
        Boolean v = record.getBoolean(i); colValue = v == null ? null : v.
            toString(); break;
    }
    case DATETIME: {
        Date v = record.getDatetime(i); colValue = v == null ? null : v.
            toString(); break;
    }
    case DOUBLE: {
        Double v = record.getDouble(i); colValue = v == null ? null : v.
            toString(); break;
    }
    case STRING: {
        String v = record.getString(i);
        colValue = v == null ? null : v.toString(); break;
    }
    default:
        throw new RuntimeException("Unknown column type: "
            + column.getType());
    }
    System.out.print(colValue == null ? "null" : colValue); if (i !=
        schema.getColumns().size())
        System.out.print("\t");
    }
    System.out.println();
}
}

public class DownloadThreadSample {
    private static String accessId = "<your access id>"; private static
        String accessKey = "<your access Key>";
    private static String tunnelUrl = "<your tunnel endpoint>";
    private static String odpsUrl = "<your odps endpoint>";
    private static String project = "<your project>"; private static
        String table = "<your table name>";
    private static String partition = "<your partition spec>";
    private static int threadNum = 10; public static void main(String args
        []) {
        Account account = new AliyunAccount(accessId, accessKey);
        Odps odps = new Odps(account); odps.setEndpoint(odpsUrl); odps.
            setDefaultProject(project);
        TableTunnel tunnel = new TableTunnel(odps); tunnel.setEndpoint(
            tunnelUrl);
        PartitionSpec partitionSpec = new PartitionSpec(partition); DownloadSe
            ssion downloadSession;
        try {
```

```
downloadSession = tunnel.createDownloadSession(project, table,
partitionSpec);
System.out.println("Session Status is : "
+ downloadSession.getStatus().toString());
long count = downloadSession.getRecordCount(); System.out.println("
RecordCount is: " + count);
ExecutorService pool = Executors.newFixedThreadPool(threadNum);
ArrayList<Callable<Long>> callers = new ArrayList<Callable<Long>>();
long start = 0;
long step = count / threadNum;
for (int i = 0; i < threadNum - 1; i++) {
RecordReader recordReader = downloadSession.openRecordReader( step * i
, step);
callers.add(new DownloadThread( i, recordReader, downloadSession.
getSchema()));
}
RecordReader recordReader = downloadSession.openRecordReader(step * (
threadNum - 1), count
+ ((threadNum - 1) * step));
callers.add(new DownloadThread( threadNum - 1, recordReader,
downloadSession.getSchema()));
Long downloadNum = 0L;
List<Future<Long>> recordNum = pool.invokeAll(callers); for (Future<
Long> num : recordNum)
downloadNum += num.get(); System.out.println("Record Count is: " +
downloadNum); pool.shutdown();
} catch (TunnelException e) { e.printStackTrace();
} catch (IOException e) { e.printStackTrace();
} catch (InterruptedException e) { e.printStackTrace();
} catch (ExecutionException e) { e.printStackTrace();
}
}
}
}
```

1.6.3 Appendix

1.6.3.1 Tunnel upload/download FAQ

This topic describes frequently asked questions (FAQs) about tunnel upload and download.

What is MaxCompute Tunnel?

Tunnel is data channel of MaxCompute, you are available to upload or download data through Tunnel to or from MaxCompute. You can upload and download only table data (excluding view data) with MaxCompute Tunnel.

Can block IDs be repeated?

Each block ID in an Upload session must be unique. After a block ID is used to start RecordWriter in an upload session, data is written, and the session is closed and committed, this block ID cannot be used to start another RecordWriter. A maximum of 20,000 blocks are supported, with the block IDs ranging from 0 to 19999.

Is there a limit on block size?

The maximum size of a block is 100 GB. We strongly recommend that 64 MB or more data is written into each block. Each block is a file. A file smaller than 64 MB is a small file. Excessive small files will affect the computing performance.

Can a session be shared? Does a session have a life cycle?

Each session has a 24-hour life cycle on the server. It can be used within 24 hours after being created, and can be shared among processes or threads. The block ID of each session must be unique. The procedure for distributed uploading is as follows: Create a session > Evaluate data volume > Assign blocks (for example, thread 1 uses blocks 0–100 and thread 2 uses blocks 100–200) > Prepare data > Upload data > Commit all blocks with data written.

How to process write/read timeout or I/O exceptions?

During data uploading, a network action is triggered every time the Writer writes 8 KB data. If no network action is triggered within 120 seconds, the server closes the connection and the Writer becomes unavailable. You have to start a new Writer.

The Reader in data downloading works in a similar way. If no network I/O occurs for a period of time, the connection is closed. We suggest that you run Read without inserting any interfaces from other systems.

Which languages of SDK are available for MaxCompute Tunnel?

MaxCompute Tunnel provides the Java and C++ editions of SDK.

Does MaxCompute Tunnel allow multiple consoles to upload the same table at the same time?

Yes.

Is MaxCompute Tunnel suitable for batch uploading or stream uploading?

MaxCompute Tunnel is more suitable for batch uploading.

Are partitions required for data uploading through MaxCompute Tunnel?

Yes, MaxCompute Tunnel does not automatically build partitions.

What is the relationship between Dship and MaxCompute Tunnel?

Dship is a tool that uploads and downloads data through MaxCompute Tunnel.

Does data uploaded with MaxCompute Tunnel append to the existing file or overwrite the data?

The uploaded data appends to the file.

What is the routing function of MaxCompute Tunnel?

The routing function allows the Tunnel SDK to get the tunnel endpoint by setting MaxCompute. That is, you can run the Tunnel SDK properly by setting the endpoint of MaxCompute.

What is the preferred size of a block when data is uploaded with MaxCompute Tunnel?

The block size depends on factors such as the network situation, real-time performance requirement, data usage, and number of small files in a cluster. If a large volume of data is uploaded continuously, the preferred block size is 64–256 MB. If the data is uploaded in a batch once every day, the block size can be 1 GB.

Why is the timeout error often reported during data downloading with MaxCompute Tunnel?

This may have occurred due to an endpoint error. Check the endpoint configuration . A simple method is to run telnet to check the network connectivity to the endpoint .

Why does the following error occur during data downloading with MaxCompute Tunnel?

```
You have NO privilege 'odps:Select' on {acs:odps:*:projects/XXX/tables/XXX}. project 'XXX' is protected
```

The data protection function has been enabled for the project. Only the project owner has the right to transfer data from one project to another if the project data is protected.

Why does the following error occur during data uploading with MaxCompute Tunnel?

```
ErrorCode=FlowExceeded, ErrorMessage=Your flow quota is exceeded. **
```

The maximum number of concurrent requests is exceeded. By default, MaxCompute Tunnel allows a maximum of 2,000 concurrent upload and download requests (quota). Each request, once it is sent, occupies one quota unit until it ends. Try the following solutions:

- **Put the system to sleep, and try again after it awakes.**
- **Change the concurrency quota to a greater number for the project after obtaining the forecast flow pressure from the administrator.**
- **Report the problem to the project owner to check and control the top requests occupying a large quota.**

1.6.3.2 Common tunnel error codes

This topic describes common tunnel error codes.

Common tunnel error codes are as follows.

Table 1-34: Common error codes

Error code	Error	Processing recommendations
NoSuchPartition	The partition does not exist .	Tunnel doesn't create partitions, you need to create partitions and then upload or download.
InvalidProjectTable	Invalid project name or table name.	Check related names.
NoSuchProject	The project does not exist.	Check related names.
NoSuchTable	The table does not exist.	Check related names.
StatusConflict	The session expires or has been committed.	Re-create the session and upload it .
MalformedDataStream	Data format error.	Normally created because network is disconnected, or Schema and Table are different.
InvalidPartitionSpec	Invalid partition information.	Check partition information. An example of a correct value is pt='1',ct='2017'.
InvalidRowRange	Invalid designated row range, normally it exceeds the max. size or it is 0.	Check related parameters.
Unauthorized	Account information error.	Normally it is wrong AccessId or AccessKey, or local device time has a 15-minute gap with server.
DataStoreError	Storage error.	Contact the administrator.
NoPermission	No permission normally because no related permission or IP whitelist has been set.	Check whether permission is correct.
MissingPartitionSpec	Missing partition information, partition table operation must carry partition information.	Add partition information.

Error code	Error	Processing recommendations
TableModified	Data in the table is modified by other tasks while upload or download.	Re-create the session and re-try.
FlowExceeded	Exceed concurrency quota limit.	Check and control volume of concurrency. If it is needed to add concurrency, please contact the project owner or administrator to evaluate flow pressure.
InvalidResourceSpec	Normally because the project, table or partition information is different from the session.	Check related information and re-try.
MethodNotAllowed	Method is not allowed, normally try to export the view.	Exporting the view is not supported currently.
InvalidColumnSpec	Invalid column information.	Normally it is column name error while download designated column.
DataVersionConflict	It is cross-cluster coping.	Re-try later.
InternalServerError	Internal error.	Re-try or contact the administrator.

1.7 MaxCompute MapReduce

1.7.1 Overview

1.7.1.1 MapReduce

MaxCompute provides a MapReduce programming API. You can use the API to write MapReduce programs to process MaxCompute data.

MapReduce is a distributed data processing model initially proposed by Google. It later gained extensive attention in the industry and was widely used in a variety of business scenarios.

A MapReduce program processes data in two stages: the Map stage and the Reduce stage. It executes the Map stage first and then the Reduce stage. Although you can

define the processing logics of Map and Reduce, they need to follow the conventions of the MapReduce framework.

The following is a detailed procedure of how MapReduce processes data:

1. Before you formally start Map, ensure that `partition` is set for input data. The input data is divided into equal-sized blocks, which are partitions. Each partition is processed as the input of a single Map worker so that multiple Map workers can work together.
2. After partitioning, multiple Map Workers start working simultaneously. Each Map Worker reads its respective shard, computes the shard, and works out the result to Reduce.



Note:

During data output, each Map worker needs to specify one key for each output data. The key decides the Reduce worker for which the data is targeted. Multiple keys may correspond to a single Reduce worker. Data of the same key is sent to the same Reduce worker, and a single Reduce worker may receive data with different keys.

3. Before entering the Reduce stage, the MapReduce framework will sort the data Key values to make data with the same Key values adjacent. If you specify `Combiner`, the framework will call `Combiner` and aggregate data with the same Key.



Note:

You can customize the `Combiner` logic. Unlike the typical MapReduce framework protocol, MaxCompute requires the input and output parameters be consistent with those of Reduce. This process is generally called `Shuffle`.

4. When entering the Reduce stage, data with same Key will be in the same Reduce Worker. A single Reduce Worker may receive data from multiple Map Workers. Each Reduce worker performs the Reduce operation on multiple values with the same key. Finally, the multiple data entries with 1 Key will become 1 Value after the Reduce operation.

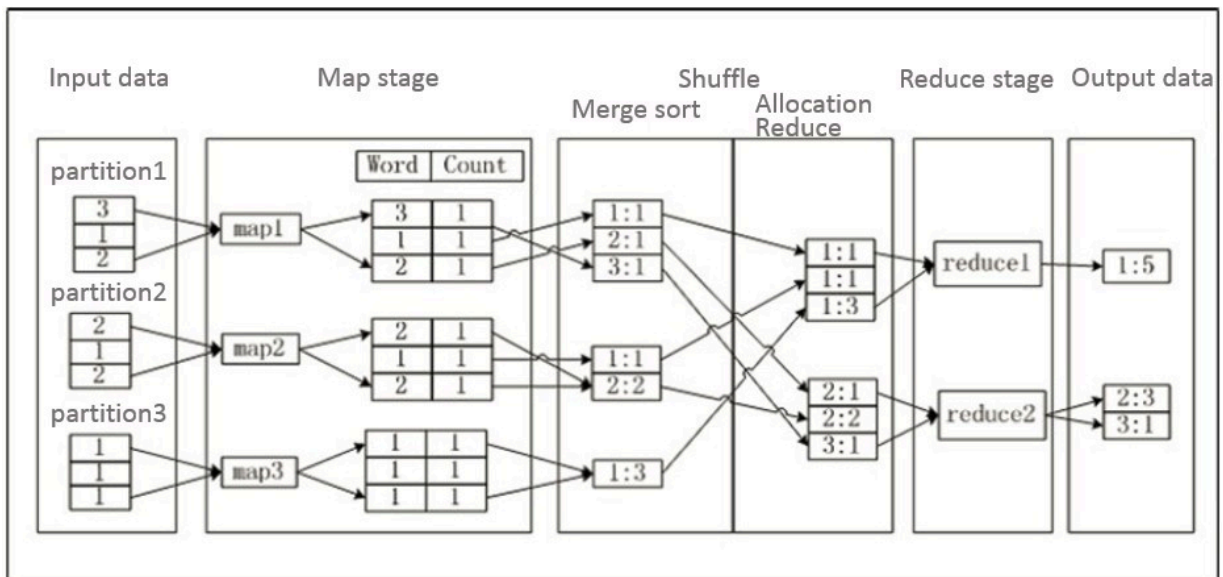


Note:

The preceding section is only a brief introduction to MapReduce. For more details, see related documentation.

The following uses WordCount as an example to explain the related concepts of MaxCompute MapReduce in different stages.

Assume there is a file a.txt, and in each line in the text is a digit. You need to count the number of times each digit appears. Each number is called a Word, and the number of its occurrences is called the Count. To use MaxCompute MapReduce for this purpose, perform the steps shown in the following figure.



1. Partition a.txt data and use data in each partition as input of a single Map worker.
2. For the Map processing input, the Count parameter is set to 1 for each obtained number. The Word | Count pair is output as a Word data key.
3. At the start of the Shuffle stage, the output of each Map worker is sorted by key value (Word value). MapReduce performs the COMBINER operation on the sorted outputs, combining data of the same key (Word value) to form a new Word | Count pair. The process is called combine sorting.
4. Later in the Shuffle stage, data is sent to the Reduce side. The Reduce Worker sorts the received data again depending on Key value.
5. During the Reduce stage, each Reduce Worker uses the same logic as Combiner while processing data, and adds the Count with the same Key value (Word value) to obtain the output result.



Note:

Because all the MaxCompute data is saved in the table, the input and output of MaxCompute MapReduce can only be a table. Customizing the output format is not permitted, and similar file system APIs are not provided.

1.7.1.2 Extended MapReduce

In a traditional MapReduce model, data must be stored in a distributed file system (such as an HDFS or a MaxCompute table) after each round of MapReduce operations. A typical MapReduce application is composed of multiple MapReduce jobs. Data is written to a disk after each job is completed. Subsequent map tasks usually only read data once to prepare for the following Shuffle stage. This results in redundant I/O operations.

The computing scheduling logic of MaxCompute supports more complicated programming models. In the preceding case, a reduce operation can be followed by the next reduce operation without having a map operation in between. An extended MapReduce model is provided. This model supports any number of reduce operations after map, such as Map-Reduce-Reduce.

Hadoop Chain Mapper and Chain Reducer also support similar serialized map or reduce operations. However, they are essentially different from the extended MapReduce (MR2) model. Chain Mapper and Chain Reducer are based on the traditional MapReduce model. They support one or more mapper operations, not reducer operations, after the original mapper or reduce operation. One benefit is that you can reuse the preceding mapper business logic by splitting a map or reduce operation into multiple mapper stages. This, however, does not change the underlying scheduling or I/O model.

1.7.2 Features

1.7.2.1 Run command

The MaxCompute client provides a jar command for running MapReduce jobs.


Command syntax:

```
Usage: jar [<GENERIC_OPTIONS>] <MAIN_CLASS> [ARGS]
-conf <configuration_file> Specify an application configuration file
-classpath <local_file_list> classpaths used to run mainClass
-D <name>=<value> Property value pair, which will be used to run
mainClass
-local Run job in local mode
```

`-resources <resource_name_list>` file/table resources used in mapper or reducer, separate by comma

The following table describes the parameters.

Table 1-35: Parameters

Parameter	Description
<code>-conf <configuration file></code>	Indicates a JobConf file.
<code>-classpath <local_file_list></code>	Indicates the classpath for local execution. It specifies the local paths (including relative path and absolute path) of the jar packet where the main function is located.
<code>-D <prop_name>=<prop_value></code>	Indicates the java attribute of <code><mainClass></code> in local execution. You can define multiple attributes.
<code>-local</code>	Indicates that the MapReduce job is run locally. It is mainly used for program debugging.
<code>-resources <resource_name_list></code>	<p>Declares the resources used by a MapReduce job. Typically, you must specify the name of the resource where the Map or Reduce function is located in <code>resource_name_list</code>.</p> <div> Notice: If the Map or Reduce function reads from other MaxCompute resources, you must add the names of these resource to <code>resource_name_list</code>. Multiple resources are separated by commas. If you want to use resources in another project, you must append <code>PROJECT_NAME/resources/</code> before the resource name, such as <code>-resources otherproject/resources/resfile</code>.</div>



Note:

The preceding optional parameters are included in `<GENERIC_OPTIONS>`.

You can use the `-conf` option to specify the JobConf file. This file can affect the settings of JobConf in the SDK. For more information about JobConf, see the introduction of MapReduce core interfaces. The following is an example of a JobConf file.

Example:

```
<configuration>
<property>
<name>import.filename</name>
<value>resource.txt</value>
</property>
</configuration>
```

In the preceding example, the JobConf file is used to define a variable named **import.filename**. The value of this variable is **resource.txt**. You can use the JobConf API in MapReduce to obtain the value of this variable. You can also use the JobConf API in the SDK for the same purpose.

Example:

```
jar -resources mapreduce-examples.jar -classpath mapreduce-examples.jar
org.alidata.odps.mr.examples.WordCount wc_in wc_out
add file data/src.txt
jar -resources src.txt,mapreduce-examples.jar -classpath mapreduce-examples.jar org.alidata.odps.mr.examples.WordCount wc_in wc_out
add file data/a.txt
add table wc_in as test_table add jar work.jar
jar -conf odps-mapred.xml -resources a.txt,test_table,work.jar
-classpath work.jar:otherlib.jar
-D import.filename=resource.txt org.alidata.odps.mr.examples.WordCount
args
```

1.7.2.2 Concepts

1.7.2.2.1 MapReduce

Map and Reduce functions support setup and cleanup methods for their corresponding **map()** and **reduce()** methods. A setup method is called prior to the **Map()** or **Reduce()** method. Each worker calls it once. A cleanup method is called after the **Map()** or **Reduce()** method. Each worker calls it once.

**Note:**

For more information about the usage example, see [Example program](#).

1.7.2.2.2 Sorting

Columns in the key records output by map can be set as sort columns. Custom comparators are not supported. You can select a few sort columns as group columns. Custom group comparators are not supported. Sort columns are generally used to sort user data, while group columns are used for secondary sorting.

**Note:**

For detailed examples, see [Secondary sorting source code](#).

1.7.2.2.3 Partition

MaxCompute supports partition columns and custom partitioners. Partition columns take precedence over custom partitioners. Partitioners are used to allocate Map output data to different Reduce workers based on the partition logic.

1.7.2.2.4 Combiner

The Combiner function combines adjacent records in the Shuffle stage. You can choose to use the Combiner function based on your business logic. The Combiner function is an optimized MapReduce computing framework. The Combiner logic is the same as the Reduce logic. After map outputs data, the framework merges data with the same key value on the map side locally.

1.7.2.2.5 Input and output

- MaxCompute MapReduce supports the following build-in data types: Bigint, Double, String, Datetime, Boolean, Decimal, Tinyint, Smallint, Int, Float, Varchar, Timestamp, Binary, Array, Map, and Struct. MapReduce does not support custom data types.
- MapReduce supports input from multiple tables with different schemas. You can use the map function to obtain the table information corresponding to the current record.
- MapReduce supports NULL as input, but does not support views as input.
- Reduce can write output to different tables or different partitions of a table. The target tables can have different schemas. Each output is identified by a label. An output is not labeled by default. At least one output is generated.

**Note:**

For examples, see [Example programs](#).

1.7.2.2.6 Read data from resources

You can use the Map or Reduce function to read data from MaxCompute resources. Any Map or Reduce worker loads resources to the memory for you to write code.

**Note:**

For a detailed example, see [Resource utilization example](#).

1.7.2.2.7 Run MapReduce tasks locally

You can specify the `-local` parameter in the `jar` command to enable local debugging by simulating the running of a local MapReduce instance. During local running, the client downloads from MaxCompute the metadata and data of the input table, required resources, and the metadata of the output table needed for local debugging. The client saves the data to a local directory named `warehouse`. When MapReduce running is complete, MapReduce saves the computing results to a file in the `warehouse` directory. If the input table and required resources are already downloaded to the local `warehouse` directory, MapReduce directly references the data and files in the `warehouse` directory the next time it runs, instead of downloading the data again.

A MapReduce instance that is running locally may start multiple map or reduce processes, which run serially rather than concurrently. The simulated running process is different from an actual distributed running process in the following aspects:

- **Input table row count:** Up to 100 rows of data can be downloaded.
- **Resource usage:** In a distributed environment, MaxCompute limits the size of referenced resources. For more information, see [Application limits](#). During local running, the size of resources is unlimited.
- **Security limits:** MaxCompute MapReduce and UDF programs running in a distributed environment are subject to Java sandbox restrictions. There is no such restrictions during local running.

The following is a simple local running example.

Example:

```
odps@my_project> jar -l com.aliyun.odps.mapred.example.WordCount wc_in wc_out;
Summary:
counters: 10
map-reduce framework combine_input_groups=2 combine_output_records=2
map_input_bytes=4 map_input_records=1 map_output_records=2 map_output_[wc_out]_bytes=0 map_output_[wc_out]_records=0 reduce_input_groups=2 reduce_output_[wc_out]_bytes=8 reduce_output_[wc_out]_records=2
OK
```



Note:

For a WordCount example, see [WordCount example](#).

If you run the local debugging command for the first time, a directory named warehouse is created in the current path after the command is complete. The following figure shows the directory structure of warehouse.

```
<warehouse>
|__my_project
|   |__<_tables_>
|   |   |__wc_in
|   |   |   |__data
|   |   |   |__<_schema_>
|   |   |__wc_out
|   |   |   |__data
|   |   |   |__<_schema_>
|   |__<_resources_>
|   |   |__table_resource_name
|   |   |   |__<_ref_>
|   |   |__file_resource_name
```

Directories of the same level as my_project are projects. wc_in and wc_out are data tables. The table file data that you read or write using the jar command is downloaded to directories of this level. The schema file contains the metadata of a table in the following format:

```
project=local_project_name
table=local_table_name
columns=col1_name:col1_type,col2_name:col2_type
partitions=p1:STRING,p2:BIGINT
-- This field is not required in this example.
```



Note:

Separate the name and type of a column with a colon (:), and separate columns with commas (.). You need to declare the project name and table name at the top of the schema files as projectname.tablename. Separate the declaration from column definition with a comma (.). The data file stores table data. The number of columns and data values must match the definition in the schema file. Separate columns with commas (.).

Example of wc_in schema file:

```
my_project.wc_in,key:STRING,value:STRING
```

Example of the corresponding data file:

```
0,2
```

**Note:**

The client downloads the metadata and part of the data of a table from MaxCompute, and saves the data to the preceding files. The next time you run this example, the client directly uses the data in the wc_in directory, instead of downloading it again. Note that you can download data from MaxCompute only when running MapReduce locally. If you use the Eclipse plugin for local debugging, you cannot download data from MaxCompute.

Example of wc_out schema file:

```
my_project.wc_out,key:STRING,cnt:BIGINT
```

Example of the corresponding data file:

```
0,1  
2,1
```

**Note:**

The client downloads the metadata of we_out from MaxCompute, and saves the data to the schema file. The data file is generated to store the local running results. You can also compile schema and data files, and save them in the corresponding table directory. Then, when you run MapReduce locally, the client detects these files in the table directory, and will not download data from MaxCompute. The local table directory does not have to correspond to an actual table in MaxCompute.

The following table compares the features of Hadoop MapReduce with MaxCompute MapReduce.

Table 1-36: Feature comparison

Feature	Hadoop MapReduce	MaxCompute MapReduce
Task progress report	The map stage calculates the read data volume. The Reduce step monitors the progress of various phases. Based on this information, the overall task progress can be estimated. The client can obtain real-time task progress. Task progress is not controlled by users.	Real-time task progress reporting is supported in a different way.
Statistics	Custom statistics, real-time summary, and real-time updates are supported. Real-time statistics updates are not controlled by users.	Real-time statistics updates are not supported during runtime . Calculation and summary are performed after a task is completed.
File compression	Users have an option to specify compressed storage.	File compression is not supported . Users cannot directly store files.
Speculative execution	/	Speculative execution is enabled by default, and cannot be configured by users.
End of task notification	The server notifies the client of task completion.	No notifications are provided. When using the SDK to submit tasks, users must poll task status constantly until the tasks are completed.

1.7.3 SDK introduction

1.7.3.1 Major API overview

Table 1-37: Major APIs

API	Description
MapperBase	User-defined Map functions must inherit this class. It converts the record objects in the input table into key-value pairs, and outputs them to the Reduce stage or directly to the result table by skipping the Reduce stage. The jobs that skip the Reduce stage and directly output calculation results are also called Map-Only jobs.

API	Description
ReducerBase	User-defined Reduce functions must inherit this class. It reduces a set of values associated with a key.
TaskContext	One of the input parameters of multiple member functions of MapperBase and ReducerBase. It contains the contextual information of task execution.
JobClient	It is used to submit and manage jobs, including the blocking mode (synchronous) and non-blocking mode (asynchronous).
RunningJob	Job runtime objects for tracking running MapReduce job instances.
JobConf	Describes the configuration of a MapReduce task. The JobConf object is generally defined in the main program (main function). Then, JobClient submits the job to MaxCompute.

1.7.3.2 API description

1.7.3.2.1 MapperBase

The following table lists the major function APIs.

Table 1-38: Major APIs

API	Description
void cleanup(TaskContext context)	The Map method is called after the map stage ends.
void map(long key, Record record, TaskContext context)	Map method for processing records in the input table.
void setup(TaskContext context)	The Map method is called before the map stage begins.

1.7.3.2.2 ReducerBase

The following table lists the major function APIs.

Table 1-39: Major APIs

API	Description
void cleanup(TaskContext context)	The Reduce method is called after the reduce stage ends.
void reduce(Record key, Iterator<Record > values, TaskContext context)	Reduce method, processing records in input table.

API	Description
void setup(TaskContext context)	The Reduce method is called before the reduce stage begins.

1.7.3.2.3 TaskContext

The following table lists the major function APIs.

Table 1-40: Major APIs

API	Description
TableInfo[] getOutputTableInfo()	Get output table information.
Record createOutputRecord()	Create record object of default output table.
Record createOutputRecord(String label)	Create record object of given label output table.
Record createMapOutputKeyRecord()	Create record object of Map output Key.
Record createMapOutputValueRecord()	Create record object of Map output Value.
void write(Record record)	Write record to default output for writing data at Reduce end. Can be called multiple times at Reduce end.
void write(Record record, String label)	Write record to given label output for writing data at Reduce end. Can be called multiple times at Reduce end.
void write(Record key, Record value)	Write record to intermediate result for writing data at Map end. Can be called multiple times at Map end.
BufferedInputStream readResourceFileAsStream(String resourceName)	Read file type resource.
Iterator<Record > readResourceTable(String resourceName)	Reads table type resources.
Counter getCounter(Enum<? > name)	Get Counter object of given name.
Counter getCounter(String group, String name)	Get Counter object of given group name and name.

API	Description
void progress()	Report heartbeat information to the MapReduce framework. If the processing time of your method is extended, and you have not called the framework during the current process, call it to avoid a task timeout period. 600 seconds is set as the timeout period by default.

**Note:**

TaskContext API has a progress function which prevents it from being forced out due to time-out if a worker runs for a long time. This API sends heartbeats to the framework rather than reporting the Worker progress. The default worker timeout period for MaxCompute MapReduce is 10 minutes (which cannot be modified by the user). If the worker does not send a heartbeat (calling the progress API) in 10 minutes, the framework terminates the worker and the MapReduce task fails. We recommend that you let the worker call the progress API periodically in the Mapper/Reducer function to prevent the framework from terminating the task unexpectedly.

1.7.3.2.4 JobConf

The following table lists the major function APIs.

Table 1-41: Major APIs

API	Description
void setResources(String resourceNames)	Declare resources this job uses. Only declared resources are available to be read through TaskContext object while run Mapper/Reducer.
void setMapOutputKeySchema(Column[] schema)	Set attribute of Key output from Mapper to Reducer.
void setMapOutputValueSchema(Column[] schema)	Set attribute of Value output from Mapper to Reducer.
void setOutputKeySortColumns(String[] cols)	Set sorting columns of Key output from Mapper to Reducer.

API	Description
void setOutputGroupingColumns(String [] cols)	Set Key grouping columns.
void setMapperClass(Class<? extends Mapper > > theClass)	Set Mapper function of job.
void setPartitionColumns(String[] cols)	Set partition columns designated by job . It is all the columns of Mapper output Key by default.
void setReducerClass(Class<? extends Reducer > theClass)	Set the job reducer.
void setCombinerClass(Class<? extends Reducer > theClass)	Set job combiner. When run at Map side , the function is similar to a single Map and local Key value of the local Reduce.
void setSplitSize(long size)	Set input split size, unit: MB, default value: 640.
void setNumReduceTasks(int n)	Set number of Reducer task, it is 1/4 number of Mapper task by default.
void setMemoryForMapTask(int mem)	Set memory size of single Worker in Mapper tasks, unit: MB, default value: 2048.
void setMemoryForReduceTask(int mem)	Set memory size of single Worker in Reduce tasks, unit: MB, default value: 2048.
void setOutputSchema(Column[] schema, String label)	Set output attribute of designated label. While multiplexed output, each output corresponds to 1 label.



Note:

- Normally, GroupingColumns is included in KeySortColumns, and KeySortColumns is included in Key.
- At the Map side, the Record output from Mapper calculates Hash value and then , based on the set PartitionColumns, determines which Reducer to distribute to for sorting Records based on KeySortColumns.
- At the Reduce side, after sorting input Records based on KeySortColumns, Records are grouped based on columns designated by GroupingColumns,

and are input in traversal sequence. The use of the same Records in columns designated by GroupingColumns as 1 input called by reduce function.

1.7.3.2.5 JobClient

The following table lists the major function APIs.

Table 1-42: Major APIs

API	Description
static RunningJob runJob(JobConf job)	Use blocking (synchronous) mode to submit MapReduce jobs and return immediately.
static RunningJob submitJob(JobConf job)	Use non-blocking (asynchronous) mode to submit MapReduce jobs and return immediately.

1.7.3.2.6 RunningJob

The following table lists the major function APIs.

Table 1-43: Major APIs

API	Description
String getInstanceID()	Get instance ID for checking run log and job management.
boolean isComplete()	Check whether job is complete.
boolean isSuccessful()	Check whether job instance is successful.
void waitForCompletion()	Wait until job instance is complete. It is typically is used for jobs submitted is asynchronous mode.
JobStatus getJobStatus()	Check job instance status.
void killJob()	End the job.
Counters getCounters()	Get Counter information.

1.7.3.2.7 InputUtils

The following table lists the major function APIs.

Table 1-44: Major APIs

API	Description
static void addTable(TableInfo table, JobConf conf)	Add table to task input. It can be called more than once. Newly added tables are appended to input queue.
static void setTables(TableInfo [] tables, JobConf conf)	Add tables to task input.

1.7.3.2.8 OutputUtils

The following table lists the major function APIs.

Table 1-45: Major APIs

API	Description
static void addTable(TableInfo table, JobConf conf)	Adds a table to the task output. It can be called more than once. Newly added tables are appended to the output queue.
static void setTables(TableInfo [] tables, JobConf conf)	Adds multiple tables to the task output.

1.7.3.2.9 Pipeline

Pipeline is the main class of MR2. A Pipeline can be built with Pipeline.Builder.

Major Pipeline APIs are as follows:

```
public Builder addMapper(Class<? extends Mapper> mapper)
public Builder addMapper(Class<? extends Mapper> mapper, Column[]
keySchema, Column[] valueSchema, String[] sortCols, SortOrder[] order
, String[] partCols, Class<? extends Partitioner> theClass, String[]
groupCols)
public Builder addReducer(Class<? extends Reducer> reducer)
public Builder addReducer(Class<? extends Reducer> reducer, Column[]
keySchema, Column[] valueSchema, String[] sortCols, SortOrder[] order
, String[] partCols, Class<? extends Partitioner> theClass, String[]
groupCols)
public Builder setOutputKeySchema (Column [] keySchema)
public Builder setOutputValueSchema (Column [] valueSchema)
public Builder setOutputKeySortColumns (String [] sortCols)
public Builder setOutputKeySortOrder (SortOrder [] order)
public Builder setPartitionColumns (String [] partCols)
public Builder setPartitionerClass(Class<? extends Partitioner>
theClass)
public Builder setOutputGroupingColumns(String[] cols)
```

Example:

```
job job = new job ();
```



```

pipeline pipeline = pipeline. builder ()
. Addmapper (maid. Class)
.setOutputKeySchema(
new Column[] { new Column("word", OdpsType.STRING) })
.setOutputValueSchema(
new Column[] { new Column("count", OdpsType.BIGINT) })
. addreducer (Sumreducer. class)
. setoutputkeyschema (
new Column[] { new Column("count", OdpsType.BIGINT) })
.setOutputValueSchema(
new column [] {new column ("word", OdpsType. string),
new column ("count", OdpsType. bigint)})
. addreducer (Identityreducer. class). createPipeline ();
job.setPipeline(pipeline); job.addInput(...)
job.addOutput(...) job.submit();

```

As shown in the preceding example, you can build MapReduce tasks of a Map operation followed by two Reduce operations in the MAIN function. If you are familiar with the basic features of MapReduce, you can use MR2 easily . We also recommend that you learn the basic features of MapReduce before using MR2, namely configuring MapReduce tasks through JobConf. JobConf can only configure MapReduce tasks of a Map operation followed by a single Reduce operation.

1.7.4 Data types

MapReduce supports the following data types: bigint, double, string, datetime, boolean, decimal, tinyint, smallint, int, float, varchar, timestamp, binary, array, map, and struct. The following table lists the mappings between MaxCompute data types and Java types.

Table 1-46: Data type mapping

MaxCompute type	Java type
Tinyint	java.lang.Byte
Smallint	java.lang.Short
Int	java.lang.Integer
Bigint	java.lang.Long
Float	java.lang.Float
Double	java.lang.Double
Decimal	java.math.BigDecimal
Boolean	java.lang.Boolean
String	java.lang.String
Varchar	com.aliyun.odps.data.Varchar

MaxCompute type	Java type
Binary	com.aliyun.odps.data.Binary
Datetime	java.util.Date
Timestamp	java.sql.Timestamp
Array	java.util.List
Map	java.util.Map
Struct	com.aliyun.odps.data.Struct

1.7.5 Limits

The following limits apply to MapReduce:

- A Map or Reduce worker consumes 2,048 MB memory by default. The value range is 256 MB to 12 GB.
- Each task can reference up to 256 resources. Each partitioned table is considered as one unit.
- Each task can have up to 1,024 inputs and up to 256 outputs.
- Each task can have up to 64 custom counters.
- The number of Map instances in a job is calculated by the framework based on the split size. If no input table is specified, you can use `odps.stage.mapper.num` to set the number of Map instances. The number is in the range of 1 to 100,000.
- The default number of Reduce instances in a job is 1/4 of the number of Map instances. You can set this number in the range of 0 to 2,000. The following situation may occur: Reduce instances process much more data than Map instances, which results in a slow Reduce stage. Furthermore, a maximum of 2,000 Reduce instances can be created.
- Each Map or Reduce instance can retry three times after failure. Exceptions that do not allow retries can cause a job to fail.
- For local jobs, the number of Map or Reduce workers cannot exceed 100, while the number of downloads for an input is 100 by default.
- Each Map or Reduce worker can read a resource a maximum of 64 times.
- A task can reference a maximum of 2 GB resources.
- The framework determines the number of Map instances based on the split size.
- The length of string columns in MaxCompute tables cannot exceed 8 MB.

- If no data access operations or heartbeat packets are sent through context.
progress(), the default timeout period of a Map or Reduce worker is 600s.

1.7.6 Sample programs

1.7.6.1 WordCount example

Example:

```
package com.aliyun.odps.mapred.open.example;
import java.io.IOException; import java.util.Iterator;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
public class WordCount {
    public static class TokenizerMapper extends MapperBase {
        private Record word;
        private Record one;
        @Override
        public void setup(TaskContext context) throws IOException {
            word = context.createMapOutputKeyRecord();
            one = context.createMapOutputValueRecord();
            one.set(new Object[] { 1L });
            System.out.println("TaskID:" + context.getTaskID().toString());
        }
        @Override
        public void map(long recordNum, Record record, TaskContext context)
            throws IOException {
            for (int i = 0; i < record.getColumnCount(); i++) {
                word.set(new Object[] { record.get(i).toString() });
                context.write(word, one);
            }
        }
    }
    /**
     *A combiner class that combines map output by sum them.
     */
    public static class SumCombiner extends ReducerBase {
        private Record count;
        @Override
        public void setup(TaskContext context) throws IOException {
            count = context.createMapOutputValueRecord();
        }
        @Override
        public void reduce(Record key, Iterator<Record> values, TaskContext
            context)
            throws IOException {
            long c = 0;
            while (values.hasNext()) {
                Record val = values.next();
                c += (Long) val.get(0);
            }
            count.set(0, c);
        }
    }
}
```

```

context.write(key, count);
}
}
/**
 * A reducer class that just emits the sum of the input values.
 */
public static class SumReducer extends ReducerBase {
    private Record result = null;
    @Override
    public void setup(TaskContext context) throws IOException { result =
        context.createOutputRecord();
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext
        context)
        throws IOException {
        long count = 0;
        while (values.hasNext()) {
            Record val = values.next();
            count += (Long) val.get(0);
        }
        result.set(0, key.get(0));
        result.set(1, count);
        context.write(result);
    }
}
public static void main(String[] args)
    throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: WordCount <in_table> <out_table>");
        System.exit(2);
    }
    JobConf job = new JobConf();
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(SumCombiner.class);
    job.setReducerClass(SumReducer.class);
    job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
    job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
    InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
        job);
    OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
        job);
    JobClient.runJob(job);
}
}

```

1.7.6.2 MapOnly example

For MapOnly jobs, Map directly sends < Key, Value > pairs to tables on MaxCompute . You only need to specify output tables. You do not need to specify the key-value metadata for map output.

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.conf.JobConf;

```

```
import com.aliyun.odps.mapred.utils.SchemaUtils;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.data.TableInfo;
public class MapOnly {
    public static class MapperClass extends MapperBase {
        @Override
        public void setup(TaskContext context)
            throws IOException {
            boolean is = context.getJobConf().getBoolean("option.mapper.setup",
                false);
            if (is) {
                Record result = context.createOutputRecord();
                result.set(0, "setup");
                result.set(1, 1L); context.write(result);
            }
        }
        @Override
        public void map(long key, Record record, TaskContext context) throws
            IOException {
            boolean is = context.getJobConf().getBoolean("option.mapper.map",
                false);
            if (is) {
                Record result = context.createOutputRecord();
                result.set(0, record.get(0));
                result.set(1, 1L); context.write(result);
            }
        }
        @Override
        public void cleanup(TaskContext context) throws IOException {
            boolean is = context.getJobConf().getBoolean("option.mapper.cleanup",
                false);
            if (is) {
                Record result = context.createOutputRecord();
                result.set(0, "cleanup");
                result.set(1, 1L); context.write(result);
            }
        }
    }
    public static void main(String[] args) throws Exception {
        if (args.length != 2 && args.length != 3) {
            System.err.println("Usage: OnlyMapper <in_table> <out_table> [setup|
            map|cleanup]");
            System.exit(2);
        }
        JobConf job = new JobConf();
        job.setMapperClass(MapperClass.class);
        job.setNumReduceTasks(0);
        InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
            job);
        OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
            job);
        if (args.length == 3) {
            String options = new String(args[2]);
            if (options.contains("setup")) {
                job.setBoolean("option.mapper.setup", true);
            }
            if (options.contains("map")) {
                job.setBoolean("option.mapper.map", true);
            }
            if (options.contains("cleanup")) {
                job.setBoolean("option.mapper.cleanup", true);
            }
        }
    }
}
```

```

JobClient.runJob(job);
}
}

```

1.7.6.3 Example: Input and output data to multiple objects

MaxCompute jobs can read data from multiple input tables, and write data to multiple output tables. All input tables for a job must have the same schema (number and type of columns). The output tables for a job can have different schemas (number and type of columns).

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException; import java.util.Iterator;
import java.util.LinkedHashMap;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 *Multi input & output example.
 *
 *To run: jar -resources odps-mapred-example-0.12.0.jar com.aliyun.odps
.mapred.open.example.MultipleInOut
 *mr_src,mr_src1,mr_srcpart|pt=1,mr_srcpart|pt=2/ds=2
 *mr_multiinout_out1,mr_multiinout_out2|a=1/b=1|out1,mr_multiinout_out2
 |a=2/b=2|out2;
 *
 **/
public class MultipleInOut {
    public static class TokenizerMapper extends MapperBase {
        private Record word;
        private Record one;
        @Override
        public void setup(TaskContext context) throws IOException {
            word = context.createMapOutputKeyRecord();
            one = context.createMapOutputValueRecord();
            one.set(new Object[] { 1L });
        }
        @Override
        public void map(long recordNum, Record record, TaskContext context)
            throws IOException {
            for (int i = 0; i < record.getColumnCount(); i++) {
                word.set(new Object[] { record.get(i).toString() });
                context.write(word, one);
            }
        }
    }
    public static class SumReducer extends ReducerBase {
        private Record result;
        private Record result1;
        private Record result2;

```

```
@Override
public void setup(TaskContext context) throws IOException {
    result = context.createOutputRecord();
    result1 = context.createOutputRecord("out1");
    result2 = context.createOutputRecord("out2");
}

@Override
public void reduce(Record key, Iterator<Record> values, TaskContext
context)
throws IOException { long count = 0;
while (values.hasNext()) {
    Record val = values.next();
    count += (Long) val.get(0);
}
long mod = count % 3;
if (mod == 0) {
    result.set(0, key.get(0));
    result.set(1, count);
    // No label is specified. Default output is adopted.
    context.write(result);
} else if (mod == 1) {
    result1.set(0, key.get(0));
    result1.set(1, count);
    context.write(result1, "out1");
} else {
    result2.set(0, key.get(0));
    result2.set(1, count);
    context.write(result2, "out2");
}
}

@Override
public void cleanup(TaskContext context) throws IOException {
    Record result = context.createOutputRecord();
    result.set(0, "default");
    result.set(1, 1L);
    context.write(result);
    Record result1 = context.createOutputRecord("out1");
    result1.set(0, "out1");
    result1.set(1, 1L); context.write(result1, "out1");
    Record result2 = context.createOutputRecord("out2");
    result2.set(0, "out1");
    result2.set(1, 1L); context.write(result2, "out2");
}
}

public static LinkedHashMap<String, String> convertPartSpecToMap(
String partSpec) {
    LinkedHashMap<String, String> map = new LinkedHashMap<String, String
>();
    if (partSpec != null && ! partSpec.trim().isEmpty()) {
        String[] parts = partSpec.split("/");
        for (String part : parts) {
            String[] ss = part.split("=");
            if (ss.length != 2) {
                throw new RuntimeException("ODPS-0730001: error part spec format: "+
partSpec);
            }
            map.put(ss[0], ss[1]);
        }
    }
    return map;
}

public static void main(String[] args) throws Exception {
    String[] inputs = null;
    String[] outputs = null;
```

```

if (args.length == 2) {
    inputs = args[0].split(",");
    outputs = args[1].split(",");
} else {
    System.err.println("MultipleInOut in... out...");
    System.exit(1);
}
JobConf job = new JobConf();
job.setMapperClass(TokenizerMapper.class);
job.setReducerClass(SumReducer.class);
job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
// Parse input table strings.
for (String in : inputs) { String[] ss = in.split("\\|");
    if (ss.length == 1) {
        InputUtils.addTable(TableInfo.builder().tableName(ss[0]).build(), job);
    } else if (ss.length == 2) {
        LinkedHashMap<String, String> map = convertPartSpecToMap(ss[1]);
        InputUtils.addTable(TableInfo.builder().tableName(ss[0]).partSpec(map).build(), job);
    } else {
        System.err.println("Style of input: " + in + " is not right");
        System.exit(1);
    }
}
// Parse output table strings.
for (String out : outputs) { String[] ss = out.split("\\|");
    if (ss.length == 1) {
        OutputUtils.addTable(TableInfo.builder().tableName(ss[0]).build(), job);
    } else if (ss.length == 2) {
        LinkedHashMap<String, String> map = convertPartSpecToMap(ss[1]);
        OutputUtils.addTable(TableInfo.builder().tableName(ss[0]).partSpec(map).build(), job);
    } else if (ss.length == 3) {
        if (ss[1].isEmpty()) {
            LinkedHashMap<String, String> map = convertPartSpecToMap(ss[2]);
            OutputUtils.addTable(TableInfo.builder().tableName(ss[0]).partSpec(map).build(), job);
        } else {
            LinkedHashMap<String, String> map = convertPartSpecToMap(ss[1]);
            OutputUtils.addTable(TableInfo.builder().tableName(ss[0]).partSpec(map).label(ss[2]).build(), job);
        }
    } else {
        System.err.println("Style of output: " + out + " is not right");
        System.exit(1);
    }
}
JobClient.runJob(job);
}
}

```

1.7.6.4 Multi-task example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;

```



```
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * MultiJobs
 *
 * Running multiple job
 *
 * To run: jar -resources > multijobs_res_table,odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.mapred.open.example.MultiJobs mr_multijobs_out;
 */
public class MultiJobs {
    public static class InitMapper extends MapperBase {
        @Override
        public void setup(TaskContext context) throws IOException { Record
            record = context.createOutputRecord();
            long v = context.getJobConf().getLong("multijobs.value", 2);
            record.set(0, v);
            context.write(record);
        }
    }
    public static class DecreaseMapper extends MapperBase {
        @Override
        public void cleanup(TaskContext context) throws IOException {
            // Obtain the variable values defined by the main function from
            JobConf.
            long expect = context.getJobConf().getLong("multijobs.expect.value", -1);
            long v = -1;
            int count = 0;
            Iterator<Record> iter = context.readResourceTable("multijobs_res_table");
            while (iter.hasNext()) {
                Record r = iter.next();
                v = (Long) r.get(0);
                if (expect != v) {
                    throw new IOException("expect: " + expect + ", but: " + v);
                }
                count++;
            }
            if (count != 1) {
                throw new IOException("res_table should have 1 record, but: " + count);
            }
            Record record = context.createOutputRecord();
            v--;
            record.set(0, v);
            context.write(record);
            context.getCounter("multijobs", "value").setValue(v);
        }
    }
    public static void main(String[] args) throws Exception { if (args.
        length != 1) {
            System.err.println("Usage: TestMultiJobs <table>");
            System.exit(1);
        }
        String tbl = args[0];
```

```

long iterCount = 2;
System.err.println("Start to run init job.");
JobConf initJob = new JobConf();
initJob.setLong("multijobs.value", iterCount);
initJob.setMapperClass(InitMapper.class);
InputUtils.addTable(TableInfo.builder().tableName("mr_empty").build(), initJob);
OutputUtils.addTable(TableInfo.builder().tableName(tbl).build(), initJob);
initJob.setMapOutputKeySchema(SchemaUtils.fromString("key:string"));
initJob.setMapOutputValueSchema(SchemaUtils.fromString("value:string"));
initJob.setNumReduceTasks(0); JobClient.runJob(initJob);
while (true) {
    System.err.println("Start to run iter job, count: " + iterCount);
    JobConf decJob = new JobConf();
    decJob.setLong("multijobs.expect.value", iterCount);
    decJob.setMapperClass(DecreaseMapper.class);
    InputUtils.addTable(TableInfo.builder().tableName("mr_empty").build(), decJob);
    OutputUtils.addTable(TableInfo.builder().tableName(tbl).build(), decJob);
    decJob.setNumReduceTasks(0);
    RunningJob rJob = JobClient.runJob(decJob); iterCount--;
    if (rJob.getCounters().findCounter("multijobs", "value").getValue() == 0) { break; }
}
if (iterCount != 0) {
    throw new IOException("Job failed.");
}
}
}
}

```

1.7.6.5 Secondary sorting example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.SchemaUtils;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.data.TableInfo;
/**
 * This is an example ODPS Map/Reduce application. It reads the input
 * > table that
 * must contain two integers per record. The output is sorted by the >
 * first and
 * second number and grouped by the first number.
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.mapred.open.example.SecondarySort mr_sort_in >
 * mr_secondarysort_out;
 */

```

```
*/  
public class SecondarySort {  
/**  
 * Read two integers from each line and generate a key, value pair as >  
 * ((left,  
 * right), right).  
**/  
public static class MapClass extends MapperBase { private Record key;  
private Record value;  
@Override  
public void setup(TaskContext context) throws IOException { key =  
context.createMapOutputKeyRecord();  
value = context.createMapOutputValueRecord();  
}  
@Override  
public void map(long recordNum, Record record, TaskContext context)  
throws IOException {  
long left = 0;  
long right = 0;  
if (record.getColumnCount() > 0) { left = (Long) record.get(0);  
if (record.getColumnCount() > 1) { right = (Long) record.get(1);  
}  
key.set(new Object[] { (Long) left, (Long) right });  
value.set(new Object[] { (Long) right });  
context.write(key, value);  
}  
}  
}  
/**  
 * A reducer class that just emits the sum of the input values.  
**/  
public static class ReduceClass extends ReducerBase {  
private Record result = null;  
@Override  
public void setup(TaskContext context) throws IOException { result =  
context.createOutputRecord();  
}  
@Override  
public void reduce(Record key, Iterator<Record> values, TaskContext  
context) throws IOException {  
result.set(0, key.get(0));  
while (values.hasNext()) {  
Record value = values.next();  
result.set(1, value.get(0));  
context.write(result);  
}  
}  
}  
public static void main(String[] args) throws Exception {  
if (args.length != 2) {  
System.err.println("Usage: secondarystrot <in> <out>");  
System.exit(2);  
}  
JobConf job = new JobConf();  
job.setMapperClass(MapClass.class);  
job.setReducerClass(ReduceClass.class);  
// Set multiple columns as keys.  
// compare first and second parts of the pair  
job.setOutputKeySortColumns(new String[] { "i1", "i2" });  
// partition based on the first part of the pair  
job.setPartitionColumns(new String[] { "i1" });  
// grouping comparator based on the first part of the pair  
job.setOutputGroupingColumns(new String[] { "i1" });  
// the map output is LongPair, Long
```

```

job.setMapOutputKeySchema(SchemaUtils.fromString("i1:bigint,i2:bigint"));
job.setMapOutputValueSchema(SchemaUtils.fromString("i2x:bigint"));
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
job);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
JobClient.runJob(job); System.exit(0);
}
}

```

1.7.6.6 Resource usage example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.BufferedInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * Upload
 *
 * Import data from text file into table
 *
 * To run: jar -resources > odps-mapred-example-0.12.0.jar,mr_join_src1.txt
 * com.aliyun.odps.mapred.open.example.Upload mr_join_src1.txt >
mr_join_src1;
 */
public class Upload {
    public static class UploadMapper extends MapperBase {
        @Override
        public void setup(TaskContext context) throws IOException { Record
record = context.createOutputRecord();
StringBuilder importdata = new StringBuilder();
BufferedInputStream bufferedInput = null;
try {
byte[] buffer = new byte[1024]; int bytesRead = 0;
String filename = context.getJobConf().get("import.filename");
bufferedInput = context.readResourceFileAsStream(filename);
while ((bytesRead = bufferedInput.read(buffer)) != -1) {
String chunk = new String(buffer, 0, bytesRead);
importdata.append(chunk);
}
String lines[] = importdata.toString().split("\n"); for (int i = 0; i
< lines.length; i++) {
String[] ss = lines[i].split(",");
record.set(0, Long.parseLong(ss[0].trim()));
record.set(1, ss[1].trim()); context.write(record);
}
} catch (FileNotFoundException ex) { throw new IOException(ex);
} catch (IOException ex) { throw new IOException(ex);
}

```

```

    } finally {
    }
    }
    @Override
    public void map(long recordNum, Record record, TaskContext context)
        throws IOException {
    }
    }
    public static void main(String[] args) throws Exception { if (args.
length != 2) {
System.err.println("Usage: Upload <import_txt> <out_table>");
System.exit(2);
}
JobConf job = new JobConf();
job.setMapperClass(UploadMapper.class);
job.set("import.filename", args[0]);
job.setNumReduceTasks(0);
job.setMapOutputKeySchema(SchemaUtils.fromString("key:bigint"));
job.setMapOutputValueSchema(SchemaUtils.fromString("value:string"));
InputUtils.addTable(TableInfo.builder().tableName("mr_empty").build
(), job);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
JobClient.runJob(job);
}
}

```

**Note:**

You can use the following methods to configure JobConf:

- **Use the JobConf API in the SDK. This method has the highest priority.**
- **Use the -conf parameter in a jar command to specify a new JobConf file. This method has the lowest priority. For how to use -Conf, refer to the relevant command.**

1.7.6.7 Example for using counters

Three counters are defined in this example: map_outputs, reduce_outputs, and global_counts. You can use the setup, map, reduce and cleanup APIs of the Map or Reduce function to obtain and operate custom counters.

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException; import java.util.Iterator;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.counter.Counters;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.SchemaUtils;
import com.aliyun.odps.mapred.utils.InputUtils;

```

```
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.data.TableInfo;
/**
 * User Defined Counters
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.mapred.open.example.UserDefinedCounters mr_src >
 mr_testcounters_out;
 */
public class UserDefinedCounters {
    enum MyCounter {
        TOTAL_TASKS, MAP_TASKS, REDUCE_TASKS
    }
    public static class TokenizerMapper extends MapperBase {
        private Record word;
        private Record one;
        @Override
        public void setup(TaskContext context) throws IOException { super.
            setup(context);
            Counter map_tasks = context.getCounter(MyCounter.MAP_TASKS);
            Counter total_tasks = context.getCounter(MyCounter.TOTAL_TASKS);
            map_tasks.increment(1);
            total_tasks.increment(1);
            word = context.createMapOutputKeyRecord();
            one = context.createMapOutputValueRecord();
            one.set(new Object[] { 1L });
        }
        @Override
        public void map(long recordNum, Record record, TaskContext context)
            throws IOException {
            for (int i = 0;
                i < record.getColumnCount();
                i++) {
                word.set(new Object[] { record.get(i).toString() });
                context.write(word, one);
            }
        }
        public static class SumReducer extends ReducerBase {
            private Record result = null;
            @Override
            public void setup(TaskContext context) throws IOException { result =
                context.createOutputRecord();
                Counter reduce_tasks = context.getCounter(MyCounter.REDUCE_TASKS);
                Counter total_tasks = context.getCounter(MyCounter.TOTAL_TASKS);
                reduce_tasks.increment(1);
                total_tasks.increment(1);
            }
            @Override
            public void reduce(Record key, Iterator<Record> values, TaskContext
                context) throws IOException {
                long count = 0;
                while (values.hasNext()) {
                    Record val = values.next();
                    count += (Long) val.get(0);
                }
                result.set(0, key.get(0));
                result.set(1, count);
                context.write(result);
            }
        }
        public static void main(String[] args) throws Exception { if (args.
            length != 2) {
```

```

System.err.println("Usage: TestUserDefinedCounters <in_table> <
out_table>");
System.exit(2);
}
JobConf job = new JobConf();
job.setMapperClass(TokenizerMapper.class);
job.setReducerClass(SumReducer.class);
job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
job);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
RunningJob rJob = JobClient.runJob(job);
Counters counters = rJob.getCounters();
long m = counters.findCounter(MyCounter.MAP_TASKS).getValue();
long r = counters.findCounter(MyCounter.REDUCE_TASKS).getValue();
long total = counters.findCounter(MyCounter.TOTAL_TASKS).getValue();
System.exit(0);
}
}

```

1.7.6.8 grep example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.Mapper;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * Extracts matching regexs from input files and counts them.
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.mapred.open.example.Grep mr_src mr_grep_tmp >
mr_grep_out val;
 */
public class Grep {
/**
 * RegexMapper
 */
public class RegexMapper extends MapperBase { private Pattern pattern;
private int group;
private Record word;
private Record one;
@Override
public void setup(TaskContext context) throws IOException {
JobConf job = (JobConf) context.getJobConf();
pattern = Pattern.compile(job.get("mapred.mapper.regex"));

```

```
group = job.getInt("mapred.mapper.regex.group", 0);
word = context.createMapOutputKeyRecord();
one = context.createMapOutputValueRecord();
one.set(new Object[] { 1L });
}
@Override
public void map(long recordNum, Record record, TaskContext context)
throws IOException {
    for (int i = 0; i < record.getColumnCount(); ++i) {
        String text = record.get(i).toString();
        Matcher = pattern.matcher(text);
        while (matcher.find()) {
            word.set(new Object[] { matcher.group(group) });
            context.write(word, one);
        }
    }
}
/**
 * LongSumReducer
 */
public class LongSumReducer extends ReducerBase {
    private Record result = null;
    @Override
    public void setup(TaskContext context) throws IOException { result =
context.createOutputRecord();
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext
context) throws IOException {
        Long count = 0;
        while (values.hasNext()) {
            Record val = values.next();
            count += (Long) val.get(0);
        }
        result.set(0, key.get(0));
        result.set(1, count);
        context.write(result);
    }
}
/**
 * A {@link Mapper} that swaps keys and values.
 */
public class InverseMapper extends MapperBase {
    private Record word;
    private Record count;
    @Override
    public void setup(TaskContext context) throws IOException {
        word = context.createMapOutputValueRecord();
        count = context.createMapOutputKeyRecord();
    }
}
/**
 * The inverse function. Input keys and values are swapped.
 */
@Override
public void map(long recordNum, Record record, TaskContext context)
throws IOException {
    word.set(new Object[] { record.get(0).toString() });
    count.set(new Object[] { (Long) record.get(1) });
    context.write(count, word);
}
}
/**
 * IdentityReducer
 */
```



```
*/  
public class IdentityReducer extends ReducerBase {  
    private Record result = null;  
    @Override  
    public void setup(TaskContext context) throws IOException {  
        result = context.createOutputRecord();  
    }  
    /** Writes all keys and values directly to output. */  
    @Override  
    public void reduce(Record key, Iterator<Record> values, TaskContext  
        context) throws IOException {  
        result.set(0, key.get(0));  
        while (values.hasNext()) {  
            Record val = values.next();  
            result.set(1, val.get(0));  
            context.write(result);  
        }  
    }  
}  
public static void main(String[] args) throws Exception {  
    if (args.length < 4) {  
        System.err.println("Grep <inDir> <tmpDir> <outDir> <regex> [<group  
>]"); System.exit(2);  
    }  
    JobConf grepJob = new JobConf();  
    grepJob.setMapperClass(RegexMapper.class);  
    grepJob.setReducerClass(LongSumReducer.class);  
    grepJob.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));  
    grepJob.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint  
"));  
    InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),  
        grepJob);  
    OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),  
        grepJob);  
    grepJob.set("mapred.mapper.regex", args[3]);  
    if (args.length == 5) {  
        grepJob.set("mapred.mapper.regex.group", args[4]);  
    }  
    @SuppressWarnings("unused")  
    RunningJob rjGrep = JobClient.runJob(grepJob);  
    JobConf sortJob = new JobConf();  
    sortJob.setMapperClass(InverseMapper.class);  
    sortJob.setReducerClass(IdentityReducer.class);  
    sortJob.setMapOutputKeySchema(SchemaUtils.fromString("count:bigint  
"));  
    sortJob.setMapOutputValueSchema(SchemaUtils.fromString("word:string  
"));  
    InputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),  
        sortJob);  
    OutputUtils.addTable(TableInfo.builder().tableName(args[2]).build(),  
        sortJob);  
    sortJob.setNumReduceTasks(1); // write a single file  
    sortJob.setOutputKeySortColumns(new String[] { "count" });  
    // sort by  
    // decreasing  
    // freq  
    @SuppressWarnings("unused")  
    RunningJob rjSort = JobClient.runJob(sortJob);  
}
```

}

1.7.6.9 JOIN example

The MaxCompute MapReduce framework does not support the JOIN logic. However, you can use custom Map and Reduce functions to complete JOIN operations. This requires extra work.

Table `mr_join_src1`(key bigint, value string) must be joined with `mr_join_src2`(key bigint, value string). The output table is `mr_join_out`(key bigint, value1 string, value2 string), where `value1` indicates the value of `mr_join_src1` and `value2` indicates the value of `mr_join_src2`.

Example:

```
package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util. arraylist;
import java.util.Iterator;
import java.util.List;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * Join
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.mapred.open.example.Join mr_join_src1 > mr_join_src2 mr_join_out;
 *
 */
public class Join {
    public static class JoinMapper extends MapperBase {
        private Record mapkey;
        private Record mapvalue;
        @Override
        public void setup(TaskContext context) throws IOException {
            mapkey = context.createMapOutputKeyRecord();
            mapvalue = context.createMapOutputValueRecord();
        }
        @Override
        public void map(long key, Record record, TaskContext context) throws IOException {
            /* Determine the source table of the record based on the value field.
             This is the user code logic. If the source table of the record cannot
             be determined based on the value field, you can add a field in the
             input table. The tag generated based on the value field is used in
             connection operations in the Reduce stage. */
            long tag = 1;
```

```
String val = record.get(1).toString();
if (val.startsWith("valb_")) {
    tag = 2;
}
mapkey.set(0, Long.parseLong(record.get(0).toString()));
mapkey.set(1, tag);
mapvalue.set(0, tag);
for (int i = 1; i < record.getColumnCount(); i++) {
    mapvalue.set(i, record.get(i));
}
context.write(mapkey, mapvalue);
}
}

public static class JoinReducer extends ReducerBase {
    private Record result = null;
    @Override
    public void setup(TaskContext context) throws IOException {
        result = context.createOutputRecord();
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext
context) throws IOException {
        long k = (Long) key.get(0);
        List<Object[]> list1 = new ArrayList<Object[]>();
        Counter cnt = context.getCounter("MyCounters", "reduce_outputs");
        cnt.increment(1);
        while (values.hasNext()) {
            Record value = values.next();
            long tag = (Long) value.get(0);
            if (tag == 1) {
                //If the data comes from the first table, the data is cached in the
list.
                //Data is sorted by key and tag, so the value with tag==1 is sorted on
the top.
                //We recommended that you exercise caution when using this method in
practice. If a key has many values,
                //the memory usage of Reduce stage increases. When the memory usage
exceeds the value set by JobConf::setMemoryForReduceTask,
                //the Reduce stage may be terminated by the system due to memory
overflow.
                list1.add(value.toArray().clone());
            } else {
                //If the data comes from the second table, the data is sorted by key
and tag.
                //All values in the first table have been saved in list1.
                //For the values in the first table
                for (Object r1: list1) { int index = 0;
                //Set the key first.
                result.set(index++, k);
                Object[] s_arr = (Object[])r1;
                result.set(index++, s_arr[1].toString());
                result.set(index++, value.get(1).toString());
                context.write(result);
                }
            }
        }
    }
}

public static void main(String[] args) throws Exception {
    if (args.length != 3) {
        System.err.println("Usage: Join <input table1> <input table2> <out>");
        System.exit(2);
    }
    JobConf job = new JobConf();
```

```

job.setMapperClass(JoinMapper.class);
job.setReducerClass(JoinReducer.class);
job.setMapOutputKeySchema(SchemaUtils.fromString("key:bigint,tag:
bigint"));
job.setMapOutputValueSchema(SchemaUtils.fromString("tagx:bigint,value:
string"));
job.setPartitionColumns(new String[] { "key" });
//Sort data by key and tag. The JOIN operation can be performed in the
Reduce stage only after data sorting by key.
//The tag indicates the source table of the current record.
job.setOutputKeySortColumns(new String[] { "key", "tag" });
job.setOutputGroupingColumns(new String[] { "key" });
//The Reduce stage uses lists to cache data. Therefore, we recommend
that you increase the memory size for Reduce workers.
job.setMemoryForReduceTask(4096);
job.setInt("table.counter", 0);
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
job);
InputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
OutputUtils.addTable(TableInfo.builder().tableName(args[2]).build(),
job);
JobClient.runJob(job);
}
}

```

1.7.6.10 Sleep example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Map;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import com.aliyun.odps.OdpsException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * Dummy class for testing MR framefork. Sleeps for a defined period of
 * > time in
 * mapper and reducer. Generates fake input for map / reduce jobs. Note
 * > that
 * generated number of input pairs is in the order of
 * <code>numMappers
 * mapSleepTime / 100</code>, so the > job uses some disk
 * space.
 * To run: jar -resources odps-mapred-example-0.12.0.jar > com.aliyun.
odps.mapred.open.example.SleepJob
 * -m 1 -r 1 -mt 1 -rt 1;
 *
 */

```

```
public class SleepJob {
    private static Log LOG = LogFactory.getLog(SleepJob.class);
    public static class SleepMapper extends MapperBase {
        private LinkedHashMap<Integer, Integer> inputs = new LinkedHashMap<
        Integer, Integer>();
        private long mapSleepDuration = 100;
        private int mapSleepCount = 1;
        private int count = 0;
        private Record key;
        @Override
        public void setup(TaskContext context) throws IOException{
            LOG.info("map setup called");
            JobConf conf = (JobConf) context.getJobConf();
            mapSleepCount = conf.getInt("sleep.job.map.sleep.count", 1);
            if (mapSleepCount < 0)
                throw new IOException("Invalid map count: " + mapSleepCount);
            mapSleepDuration = conf.getLong("sleep.job.map.sleep.time", 100)
            / mapSleepCount;
            LOG.info("mapSleepCount = " + mapSleepCount + ", mapSleepDuration = "
            + mapSleepDuration);
            final int redcount = conf.getInt("sleep.job.reduce.sleep.count", 1);
            if (redcount < 0)
                throw new IOException("Invalid reduce count: " + redcount);
            final int emitPerMapTask = (redcount * conf.getNumReduceTasks());
            int records = 0;
            int emitCount = 0;
            while (records++ < mapSleepCount) {
                int key = emitCount;
                int emit = emitPerMapTask / mapSleepCount;
                if ((emitPerMapTask) % mapSleepCount > records) {
                    ++emit;
                }
                emitCount += emit;
                int value = emit;
                inputs.put(key, value);
            }
            key = context.createMapOutputKeyRecord();
        }
        @Override
        public void cleanup(TaskContext context) throws IOException {
            // it is expected that every map processes mapSleepCount number of
            // records.
            LOG.info("map run called");
            for (Map.Entry<Integer, Integer> entry : inputs.entrySet()) {
                LOG.info("Sleeping... (" + (mapSleepDuration * (mapSleepCount - count
                )) + ") ms left");
                try {
                    Thread.sleep(mapSleepDuration);
                } catch (InterruptedException e) { throw new IOException(e); }
            }
            ++count;
            // output reduceSleepCount * numReduce number of random values, so
            that
            // each reducer will get reduceSleepCount number of keys.
            int k = entry.getKey();
            int v = entry.getValue();
            for (int i = 0; i < v; ++i) {
                key.set(new Object[] { (Long) ((long) (k + i)) });
                context.write(key, key);
            }
        }
    }
    public static class SleepReducer extends ReducerBase {
```

```
private long reduceSleepDuration = 100;
private int reduceSleepCount = 1;
private int count = 0;
@Override
public void setup(TaskContext context) throws IOException {
    LOG.info("reduce setup called");
    JobConf conf = (JobConf) context.getJobConf();
    reduceSleepCount = conf.getInt("sleep.job.reduce.sleep.count",
    reduceSleepCount);
    reduceSleepDuration = conf.getLong("sleep.job.reduce.sleep.time", 100)
    / reduceSleepCount;
    LOG.info("reduceSleepCount = " + reduceSleepCount
    + ", reduceSleepDuration = " + reduceSleepDuration);
}
@Override
public void reduce(Record key, Iterator<Record> values, TaskContext
context) throws IOException {
    LOG.info("reduce called");
    LOG.info("Sleeping... ("
    + (reduceSleepDuration * (reduceSleepCount - count)) + ") ms left");
    try {
        Thread.sleep(reduceSleepDuration);
    } catch (InterruptedException e) {
        throw new IOException(e);
    }
    count++;
}
}

public static int run(int numMapper, int numReducer, long mapSleepTime
, int mapSleepCount, long reduceSleepTime, int reduceSleepCount)
throws OdpsException {
    JobConf job = setupJobConf(numMapper, numReducer, mapSleepTime,
    mapSleepCount, reduceSleepTime, reduceSleepCount);
    JobClient.runJob(job);
    return 0;
}

public static JobConf setupJobConf(int numMapper, int numReducer, long
    mapSleepTime, int mapSleepCount, long reduceSleepTime, int reduceSlee
    pCount) {
    JobConf job = new JobConf();
    InputUtils.addTable(TableInfo.builder().tableName("mr_empty").build
    (), job);
    OutputUtils.addTable(TableInfo.builder().tableName("mr_sleep_out").
    build(), job);
    job.setNumReduceTasks(numReducer);
    job.setMapperClass(SleepMapper.class);
    job.setReducerClass(SleepReducer.class);
    job.setMapOutputKeySchema(SchemaUtils.fromString("int1:bigint"));
    job.setMapOutputValueSchema(SchemaUtils.fromString("int2:bigint"));
    job.setPartitionColumns(new String[] { "int1" });
    job.setLong("sleep.job.map.sleep.time", mapSleepTime);
    job.setLong("sleep.job.reduce.sleep.time", reduceSleepTime);
    job.setInt("sleep.job.map.sleep.count", mapSleepCount);
    job.setInt("sleep.job.reduce.sleep.count", reduceSleepCount);
    return job;
}

private static void printUsage() {
    System.err.println("SleepJob [-m numMapper] [-r numReducer]"
    + " [-mt mapSleepTime (msec)] [-rt reduceSleepTime (msec)]"
    + " [-recordt recordSleepTime (msec)]");
}

public static void main(String[] args) throws Exception {
    if (args.length < 1) {
        printUsage();
    }
}
```

```

return;
}
int numMapper = 1, numReducer = 1;
long mapSleepTime = 100, reduceSleepTime = 100, recSleepTime = 100;
int mapSleepCount = 1, reduceSleepCount = 1;
for (int i = 0; i < args.length; i++) { if (args[i].equals("-m")) {
numMapper = Integer.parseInt(args[++i]);
} else if (args[i].equals("-r")) {
numReducer = Integer.parseInt(args[++i]);
} else if (args[i].equals("-mt")) {
mapSleepTime = Long.parseLong(args[++i]);
} else if (args[i].equals("-rt")) {
reduceSleepTime = Long.parseLong(args[++i]);
} else if (args[i].equals("-recordt")) { recSleepTime = Long.parseLong
(args[++i]);
}
}
// sleep for *SleepTime duration in Task by recSleepTime per record
mapSleepCount = (int) Math.ceil(mapSleepTime / ((double) recSleepTime
));
reduceSleepCount = (int) Math.ceil(reduceSleepTime
/ ((double) recSleepTime));
run(numMapper, numReducer, mapSleepTime, mapSleepCount, reduceSleepTime, reduceSleepCount);
}
}

```

1.7.6.11 unique example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * Unique Remove duplicate words
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar
 * com.aliyun.odps.open.mapred.example.Unique mr_sort_in > mr_unique_
out
 * key|value|all;
 *
 */
public class Unique {
public static class OutputSchemaMapper extends MapperBase {
private Record key;
private Record value;
@Override
public void setup(TaskContext context) throws IOException {
key = context.createMapOutputKeyRecord();
value = context.createMapOutputValueRecord();
}
}
}

```

```
@Override
public void map(long recordNum, Record record, TaskContext context)
throws IOException {
    long left = 0;
    long right = 0;
    if (record.getColumnCount() > 0) { left = (Long) record.get(0);
    if (record.getColumnCount() > 1) { right = (Long) record.get(1);
    }
    key.set(new Object[] { (Long) left, (Long) right });
    value.set(new Object[] { (Long) left, (Long) right });
    context.write(key, value);
}
}
}

public static class OutputSchemaReducer extends ReducerBase {
    private Record result = null;
    @Override
    public void setup(TaskContext context) throws IOException {
        result = context.createOutputRecord();
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext
    context) throws IOException {
        result.set(0, key.get(0));
        while (values.hasNext()) {
            Record value = values.next();
            result.set(1, value.get(1));
        }
        context.write(result);
    }
}

public static void main(String[] args) throws Exception {
    if (args.length > 3 || args.length < 2) {
        System.err.println("Usage: unique <in> <out> [key|value|all]");
        System.exit(2);
    }
    String ops = "all";
    if (args.length == 3) { ops = args[2];
    }
    // Key Unique
    if (ops.equals("key")) {
        JobConf job = new JobConf();
        job.setMapperClass(OutputSchemaMapper.class);
        job.setReducerClass(OutputSchemaReducer.class);
        job.setMapOutputKeySchema(SchemaUtils.fromString("key:bigint,value:
        bigint"));
        job.setMapOutputValueSchema(SchemaUtils.fromString("key:bigint,value:
        bigint"));
        job.setPartitionColumns(new String[] { "key" });
        job.setOutputKeySortColumns(new String[] { "key", "value" });
        job.setOutputGroupingColumns(new String[] { "key" });
        job.set("tablename2", args[1]); job.setNumReduceTasks(1);
        job.setInt("table.counter", 0);
        InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
        job);
        OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
        job);
        JobClient.runJob(job);
    }
    // Key&Value Unique
    if (ops.equals("all")) {
        JobConf job = new JobConf();
        job.setMapperClass(OutputSchemaMapper.class);
        job.setReducerClass(OutputSchemaReducer.class);
```



```

job.setMapOutputKeySchema(SchemaUtils.fromString("key:bigint,value:
bigint"));
job.setMapOutputValueSchema(SchemaUtils.fromString("key:bigint,value:
bigint"));
job.setPartitionColumns(new String[] { "key" });
job.setOutputKeySortColumns(new String[] { "key", "value" });
job.setOutputGroupingColumns(new String[] { "key", "value" });
job.set("tablename2", args[1]); job.setNumReduceTasks(1);
job.setInt("table.counter", 0);
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
job);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
JobClient.runJob(job);
}
// Value Unique
if (ops.equals("value")) { JobConf job = new JobConf();
job.setMapperClass(OutputSchemaMapper.class);
job.setReducerClass(OutputSchemaReducer.class);
job.setMapOutputKeySchema(SchemaUtils.fromString("key:bigint,value:
bigint"));
job.setMapOutputValueSchema(SchemaUtils.fromString("key:bigint,value:
bigint"));
job.setPartitionColumns(new String[] { "value" });
job.setOutputKeySortColumns(new String[] { "value" });
job.setOutputGroupingColumns(new String[] { "value" });
job.set("tablename2", args[1]); job.setNumReduceTasks(1);
job.setInt("table.counter", 0);
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
job);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
JobClient.runJob(job);
}
}
}
}

```

1.7.6.12 Sort example

Example:

```

package com.aliyun.odps.mapred.open.example;
import java.io.IOException; import java.util.Date;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.TaskContext;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.example.lib.IdentityReducer;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
/**
 * This is the trivial map/reduce program that does absolutely nothing
 * > other
 * than use the framework to fragment and sort the input values.
 *
 * To run: jar -resources odps-mapred-example-0.12.0.jar > com.aliyun.
odps.mapred.open.example.Sort
 * mr_sort_in mr_sort_out;
 *

```

```

/**/
public class Sort {
static int printUsage() {
System.out.println("sort <input> <output>"); return -1;
}
/**
 * Implement the identity function, mapping record's first two columns
 * > to
 * outputs.
 */
public static class IdentityMapper extends MapperBase {
private Record key;
private Record value;
@Override
public void setup(TaskContext context) throws IOException {
key = context.createMapOutputKeyRecord();
value = context.createMapOutputValueRecord();
}
@Override
public void map(long recordNum, Record record, TaskContext context)
throws IOException {
Key.set (new object [] {(long) record.get (0 )});
value.set(new Object[] { (Long) record.get(1) });
context.write(key, value);
}
}
/**
 * The main driver for sort program. Invoke this method to submit the
 * map/reduce job.
 *
 * @throws IOException
 * When there is communication problems with the job tracker.
 */
public static void main(String[] args) throws Exception {
JobConf jobConf = new JobConf();
jobConf.setMapperClass(IdentityMapper.class);
jobConf.setReducerClass(IdentityReducer.class);
jobConf.setNumReduceTasks(1);
Jobconf.setmapoutputkeyschema schemautils schemeutils.fromString ("
key: bigint ");
jobConf.setMapOutputValueSchema(SchemaUtils.fromString("value:bigint
"));
InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(),
jobConf);
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
jobConf);
Date startTime = new Date(); System.out.println("Job started: " +
startTime);
JobClient.runJob(jobConf);
Date end_time = new Date(); System.out.println("Job ended: " +
end_time); System.out.println("The job took " + (end_time.getTime() -
startTime.getTime()) / 1000 + " seconds." ) ;
}
}

```

1.7.6.13 Example of using partitioned table as an input

The following examples use partitions as an input.

Example 1:

```
public static void main(String[] args) throws Exception {
```

```

JobConf job = new JobConf();
...
LinkedHashMap<String, String> input = new LinkedHashMap<String, String>();
input.put("pt", "123456");
InputUtils.addTable(TableInfo.builder().tableName("input_table").
partSpec(input).build(), job);
LinkedHashMap<String, String> output = new LinkedHashMap<String,
String>(); output.put("ds", "654321");
OutputUtils.addTable(TableInfo.builder().tableName("output_table").
partSpec(output).build(), job);
JobClient.runJob(job);
}

```

Example 2:

```

package com.aliyun.odps.mapred.open.example;
...
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Usage: WordCount <in_table> <out_table>");
System.exit(2);
}
JobConf job = new JobConf();
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(SumCombiner.class);
job.setReducerClass(SumReducer.class);
job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
Account account = new AliyunAccount("my_access_id", "my_access_key");
Odps odps = new Odps(account);
odps.setEndpoint("odps_endpoint_url");
odps.setDefaultProject("my_project");
Table table = odps.tables().get(tblname);
TableInfoBuilder builder = TableInfo.builder().tableName(tblname);
for (Partition p : table.getPartitions()) { if (applicable(p)) {
LinkedHashMap<String, String> partSpec = new LinkedHashMap<String,
String>();
for (String key : p.getPartitionSpec().keys()) {
partSpec.put(key, p.getPartitionSpec().get(key));
}
InputUtils.addTable(builder.partSpec(partSpec).build(), conf);
}
}
OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(),
job);
JobClient.runJob(job);
}

```

**Note:**

In the preceding example, the MaxCompute SDK and MapReduce SDK are used together to allow MapReduce tasks to read data from partitions. The preceding code cannot be compiled for execution. It is just an example of the main function. In the preceding example, the applicable function is the user logic that determines whether the partitions can be used as an input of the MapReduce job.

1.7.6.14 Pipeline example

Example:

```
package com.aliyun.odps.mapred.example;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.Column;
import com.aliyun.odps.OdpsException;
import com.aliyun.odps.OdpsType;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.Job;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.pipeline.Pipeline;
public class Histogram {
    public static class TokenizerMapper extends MapperBase {
        Record word;
        Record one;
        @Override
        public void setup(TaskContext context) throws IOException {
            word = context.createMapOutputKeyRecord();
            one = context.createMapOutputValueRecord();
            one.setBigint(0, 1L);
        }
        @Override
        public void map(long recordNum, Record record, TaskContext context)
            throws IOException {
            for (int i = 0; i < record.getColumnCount(); i++) {
                String[] words = record.get(i).toString().split("\\s+");
                for (String w : words) {
                    word.setString(0, w);
                    context.write(word, one);
                }
            }
        }
        public static class SumReducer extends ReducerBase { private Record
            num;
            private Record result;
            @Override
            public void setup(TaskContext context) throws IOException {
                num = context.createOutputKeyRecord();
                result = context.createOutputValueRecord();
            }
            @Override
            public void reduce(Record key, Iterator<Record> values, TaskContext
                context) throws IOException {
                long count = 0;
                while (values.hasNext()) {
                    Record val = values.next();
                    count += (Long) val.get(0);
                }
                result.set(0, key.get(0));
                num.set(0, count);
                context.write(num, result);
            }
        }
        public static class IdentityReducer extends ReducerBase {
            @Override
            public void reduce(Record key, Iterator<Record> values, TaskContext
                context) throws IOException {
```

```

while (values.hasNext()) {
    context.write(values.next());
}
}
}
}
public static void main(String[] args) throws OdpsException {
    if (args.length != 2) {
        System.err.println("Usage: orderedwordcount <in_table> <out_table>");
        System.exit(2);
    }
    Job job = new Job();
    /**
     * In the process of constructing pipeline, if you do not specify
     * mapper's OutputKeySortColumns, PartitionColumns, OutputGroupingColumns
     * ,
     * the framework defaults to its OutputKey as the default configuration
     * for the three
     */
    Pipeline pipeline = Pipeline.builder()
        .addMapper(TokenizerMapper.class)
        .setOutputKeySchema(
            new Column[] { new Column("word", OdpsType.STRING) })
        .setOutputValueSchema(
            new Column[] { new Column("count", OdpsType.BIGINT) })
        .setOutputKeySortColumns(new String[] { "word" })
        .setPartitionColumns(new String[] { "word" })
        .setOutputGroupingColumns(new String[] { "word" })
        .addReducer(SumReducer.class)
        .setOutputKeySchema(
            new Column[] { new Column("count", OdpsType.BIGINT) })
        .setOutputValueSchema(
            new Column[] { new Column("word", OdpsType.STRING)})
        .addReducer(IdentityReducer.class).createPipeline();
    job.setPipeline(pipeline);
    job.addInput(TableInfo.builder().tableName(args[0]).build());
    job.addOutput(TableInfo.builder().tableName(args[1]).build());
    job.submit(); job.waitForCompletion();
    System.exit(job.isSuccessful() == true ? 0 : 1);
}
}

```

1.8 MaxCompute Graph

1.8.1 Graph overview

1.8.1.1 Graph overview

MaxCompute Graph is a processing framework designed for iterative graph computing. Graph computing jobs use graphs to build models. Graphs are composed of vertices and edges with values. MaxCompute Graph supports the following operations to edit a graph:

- **Editing the value of vertex or edge.**
- **Adding/deleting vertex.**
- **Adding/deleting edge.**

**Note:**

When editing the vertex or edge, you must maintain the relationship between the two items.

After performing iterative graph editing and evolution, you can get the final result. Typical applications include [PageRank](#), [SSSP algorithm](#), and [K-Means algorithm](#). Furthermore, you can use the Java SDK provided by MaxCompute Graph to compile computing programs.

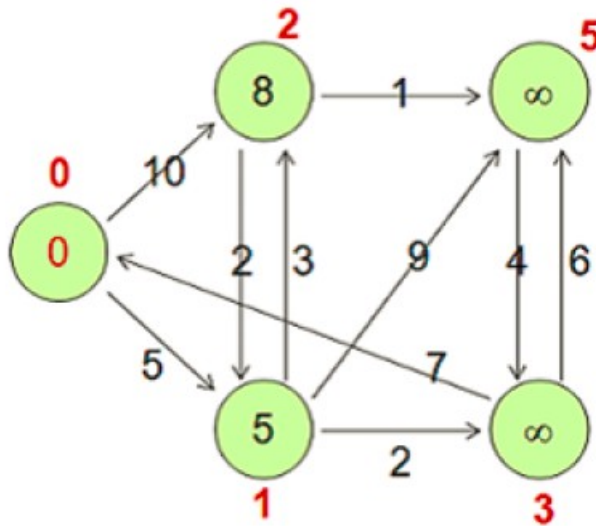
1.8.1.2 Graph data structure

MaxCompute Graph processes directed graphs, or digraphs, that consist of a vertex and an edge. MaxCompute stores data in two-dimensional tables, so you must convert graph data into two-dimensional tables and store them in MaxCompute. To perform graph analysis, you must use a custom GraphLoader to convert two-dimensional table data to vertexes and edges in the MaxCompute Graph engine. You can then determine how to break down and analyze your graph data based on your business requirements. In the following chapter, the examples provided use different tabular expressions to represent the data structure of a graph.

The vertex structure can be expressed as `< ID, Value, Halted, Edges >`, indicating respectively the vertex identifier, the value, the state (Halted, meaning whether to stop iteration), and the edge set (Edges, indicating lists of all edges starting from the vertex). The edge structure can be described as `< DestVertexID, Value >`, indicating

respectively the destination vertex (DestVertexID) and value (Value). The following figure shows Graph data structure.

Figure 1-7: Graph data structure



The preceding figure involves the following vertexes.

Table 1-47: Graph data structure

Vertex	<ID, Value, Halted, Edges>
v0	<0, 0, false, [<1, 5>, <2, 10>]>
v1	<1, 5, false, [<2, 3>, <3, 2>, <5, 9>]>
v2	<2, 8, false, [<1, 2>, <5, 1>]>
v3	<3, Long.MAX_VALUE, false, [<0, 7>, <5, 6>]>
v5	<5, Long.MAX_VALUE, false, [<3, 4>]>

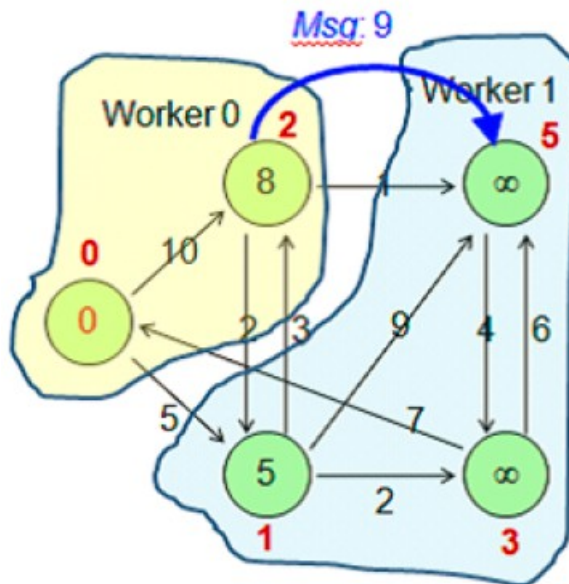
1.8.1.3 Graph logic

1.8.1.3.1 Load graph

- **Graph load:** The framework calls the custom GraphLoader to parse the records in the input table into vertices or edges.
- **Partitioning:** The framework calls the custom Partitioner to partition vertices (default partition logic: the hash value of the vertex ID modulo the number of

workers). Then, the framework distributes the partitions to the corresponding workers.

Figure 1-8: Load graph



Based on the preceding figure, if there are two workers, v0 and v2 are distributed to Worker0, because the result of the ID modulo 2 (total number of workers) is 0. v1, v3, and v5 are distributed to Worker1, because the result of the ID modulo 2 is 1.

1.8.1.3.2 Iterative computation

An iteration is a superstep. During a superstep, all vertices not in the halted state (Halted value is false) or vertices that receive messages (vertices in the halted state automatically wake up when receiving a message) are traversed. The compute method (ComputeContext context, Iterable messages) of these vertices is called.

In a custom compute method (ComputeContext context, Iterable messages):

- The messages sent by the previous superstep to the current vertex are processed.
- The graph is edited as required:
 - The values of vertices or edges are modified.
 - Messages are sent to some vertices.
 - Vertices or edges are added or deleted.
- The aggregator aggregates information to obtain global information.
- The current vertex is set to the halted or non-halted state.

- During each iteration, the framework automatically sends messages to the corresponding workers asynchronously. The messages are processed in the next superstep.

1.8.1.3.3 End of iteration

An iteration ends if any of the following conditions is satisfied.

- All vertices are in the halted state (Halted value is true), and no new messages are generated.
- The maximum number of iterations is reached.
- The terminate method of an aggregator returns true.

Example:

```
// 1. Load
for each record in input_table { GraphLoader.load();
}
// 2. Setup
WorkerComputer.setup();
for each aggr in aggregators { aggr.createStartupValue();
}
for each v in vertices { v.setup();
}
// 3. Superstep
for (step = 0; step < max; step ++ ) { for each aggr in aggregators {
aggr.createInitialValue();
}
for each v in vertices { v.compute();
}
}
// 4. Cleanup
for each v in vertices { v.cleanup();
}
WorkerComputer.cleanup();
```

1.8.2 Graph feature overview

1.8.2.1 Run a job

The MaxCompute client provides a jar command for running MaxCompute Graph jobs. This command is used in the same way as the jar command in MapReduce.


Command syntax:

```
Usage: jar [<GENERIC_OPTIONS>] <MAIN_CLASS> [ARGS]
-conf <configuration_file> Specify an application configuration file
-classpath <local_file_list> classpaths used to run mainClass
-D <name>=<value> Property value pair, which will be used to run
mainClass
-local Run job in local mode
```

-resources <resource_name_list> file/table resources used in graph, separate by comma

The following table describes the parameters.

Table 1-48: Parameters

Parameter	Description
-conf <configuration file>	Indicates a JobConf file.
-classpath <local_file_list>	Indicates the classpath for local execution. It specifies the local paths (including relative path and absolute path) of the JAR package where the main function is located.
-D <prop_name>=<prop_value>	Indicates the Java attribute of <mainClass> in local execution. You can define multiple attributes.
-local	Runs the MapReduce job locally, mainly for program debugging.
-resources <resource_name_list>	<p>Declares the resources used for running the Graph job. You typically need to specify the name of the resource where the Graph job is located in resource_name_list.</p> <div> Notice: If you read other MaxCompute resources while running the Graph job, you also need to add those resource names to resource_name_list. Multiple resources must be separated by commas (.). If you need to use resources of another project, add the prefix PROJECT_NAME/resources/, for example, -resources otherproject/resources/resfile.</div>



Note:

The preceding optional parameters are included in <GENERIC_OPTIONS>.

You can directly run the main function in the Graph job to submit the job to MaxCompute, instead of submitting the job through the MaxCompute client. Take the PageRank algorithm as an example.

Example:

```

public static void main(String[] args) throws IOException {
    if (args.length < 2)
        printUsage();
    GraphJob job = new GraphJob();
    job.setGraphLoaderClass(PageRankVertexReader.class);
    job.setVertexClass(PageRankVertex.class);
    job.addInput(TableInfo.builder().tableName(args[0]).build());
    job.addOutput(TableInfo.builder().tableName(args[1]).build());
    // Add the resources used in the job to cache resource. These
    // resources correspond to those specified by -resources and libjars in
    // the jar command.
    job.addCacheResource("mapreduce-examples.jar");
    // Add the used JAR file and other files to the class cache resource.
    // These resources correspond to those specified by -libjars in the jar
    // command.
    job.addCacheResourceToClassPath("mapreduce-examples.jar");
    // Set the configuration item corresponding to odps_config.ini in the
    // client. Replace it with the actual one in your configuration file.
    Account account = new AliyunAccount(accessId, accessKey);
    Odps odps = new Odps(account);
    odps.setDefaultProject(project);
    odps.setEndpoint(endpoint);
    SessionState.get().setOdps(odps);
    SessionState.get().setLocalRun(false); // default max iteration is 30
    job.setMaxIteration(30);
    if (args.length >= 3)
        job.setMaxIteration(Integer.parseInt(args[2]));
    long startTime = System.currentTimeMillis(); job.run();
    System.out.println("Job Finished in "
        + (System.currentTimeMillis() - startTime) / 1000.0 + " seconds");
}

```

1.8.2.2 Input and output

The input and output of MaxCompute Graph jobs must be tables. You cannot customize the input or output format.

Job input definition:

```

GraphJob job = new GraphJob();
job.addInput(TableInfo.builder().tableName("tblname").build());
// Tables are used as input.
job.addInput(TableInfo.builder().tableName("tblname").partSpec("pt1=a/
pt2=b").build());
// Partitions are used as input.
job.addInput(TableInfo.builder().tableName("tblname").partSpec("pt1=a/
pt2=b").build(), new String[]{"col2", "col0 "});
// Read only col2 and col0 of the input table. Use record.get(0) to
// obtain col2 in the load() method of GraphLoader. Both are read in the
// same sequence.

```

**Note:**

- **Multiple inputs are supported.**

- Partition filtering is not supported. For more application limits, see [Application limits](#).
- For more information about job input definition, see the `addInput` method description in `GraphJob`.
- The framework reads records from the input table and transfers the records to the user-defined `GraphLoader` to load graph data.

Job output definition:

```
GraphJob job = new GraphJob();
job.addOutput(TableInfo.builder().tableName("table_name").partSpec("
pt1=a/pt2=b").build());
// If the output table is a partitioned table, the last level of
partitions must be provided.
job.addOutput(TableInfo.builder().tableName("table_name").partSpec("
pt1=a/pt2=b").label("output1").build(), true);
// True indicates that the code will overwrite partitions specified in
tableinfo, which is similar to the INSERT OVERWRITE operation. False
is similar to the INSERT INTO operation.
```



Note:

- Multiple outputs are supported, and each output is identified by a label.
- A Graph job can use the `Write` method of `WorkContext` to write data to an output table during runtime. Multiple outputs must be labeled.
- For more information about job output definition, see the `addOutput` method description in `GraphJob`.

1.8.2.3 Read data from resources

1.8.2.3.1 Add resource in Graph program

In addition to the `jar` command, the following two methods of `GraphJob` can be used to specify the resources read by Graph:

```
void addCacheResources(String resourceNames)
void addCacheResourcesToClassPath(String resourceNames)
```

1.8.2.3.2 Use resources in Graph

In Graph, you can use the following methods of the corresponding context object `WorkerContext` to read resources:

```
public byte[] readCacheFile(String resourceName) throws IOException;
public Iterable<byte[]> readCacheArchive(String resourceName) throws
IOException;
```

```
public Iterable<byte[]> readCacheArchive(String resourceName, String
relativePath)throws IOException;
public Iterable<WritableRecord> readResourceTable(String resourceName
);
public BufferedInputStream readCacheFileAsStream(String resourceName)
throws IOException;
public Iterable<BufferedInputStream> readCacheArchiveAsStream(String
resourceName) throws IOException;
public Iterable<BufferedInputStream> readCacheArchiveAsStream(String
resourceName, String relativePath) throws IOException;
```

**Note:**

- Normally, resources are read in the setup of WorkerComputer, saved in WorkerValue, and obtained through getWorkerValue.
- The preceding stream API is recommended while reading and processing to reduce memory consumption.
- For more information about limits, see [Application limits](#).

1.8.3 Graph SDK introduction

Table 1-49: Major APIs

API	Description
GraphJob	GraphJob is inherited from JobConf to define, submit, and manage a MaxCompute Graph job.
Vertex	Vertex is an abstract of a graph and has the following attributes: id , value, halted, and edges. It is implemented through the setVertexC lass API in GraphJob.
Edge	Edge is an abstract of a graph and has the following attributes: destVertexId and value. The graph data structure is maintained by an adjacency list. The outgoing edges of a vertex are stored in its edges attribute.
GraphLoader	GraphLoader is used to load graphs. It is set through the setGraphLoaderClass API in GraphJob.
VertexReso lver	VertexResolver is used to customize the collision processing logic of the revising graph topology. It provides this logic through the setLoadingVertexResolverClass and setComputingVertexRe solverClass APIs in GraphJob for graph loading and iteration computing.

API	Description
Partitioner	Partitioner is used to partition graphs for partition computing. It is set through the setPartitionerClass API in GraphJob. By default, the HashPartitioner is used to first obtain the Vertex ID Hash value, and then to model the number of Workers.
WorkerComputer	WorkerComputer allows customized logic to be executed while Worker starts and exits. WorkerComputer is set through the setWorkerComputerClass API in GraphJob.
Aggregator	Allows you to define one or multiple Aggregators through the setAggregatorClass(Class ...) API in Aggregator.
Combiner	You can set Combiner through the setCombinerClass API in Combiner.
Counters	In the job operating logic, counters can be taken and counted through the WorkerContext API, and the framework will automatically summarize them.
WorkerContext	Context objects encapsulate the functions provided by the framework, such as revising the topology of graphs, sending messages, writing results, reading resources, and so on.

1.8.4 Development and debugging

1.8.4.1 Development procedure

MaxCompute does not provide plug-ins for Graph development. Instead, you can develop MaxCompute Graph programs in Eclipse. The recommended development procedure is as follows:

1. Write Graph code and perform local debugging to test basic functions.
2. Perform cluster debugging to verify results.

1.8.4.2 Development example

Context

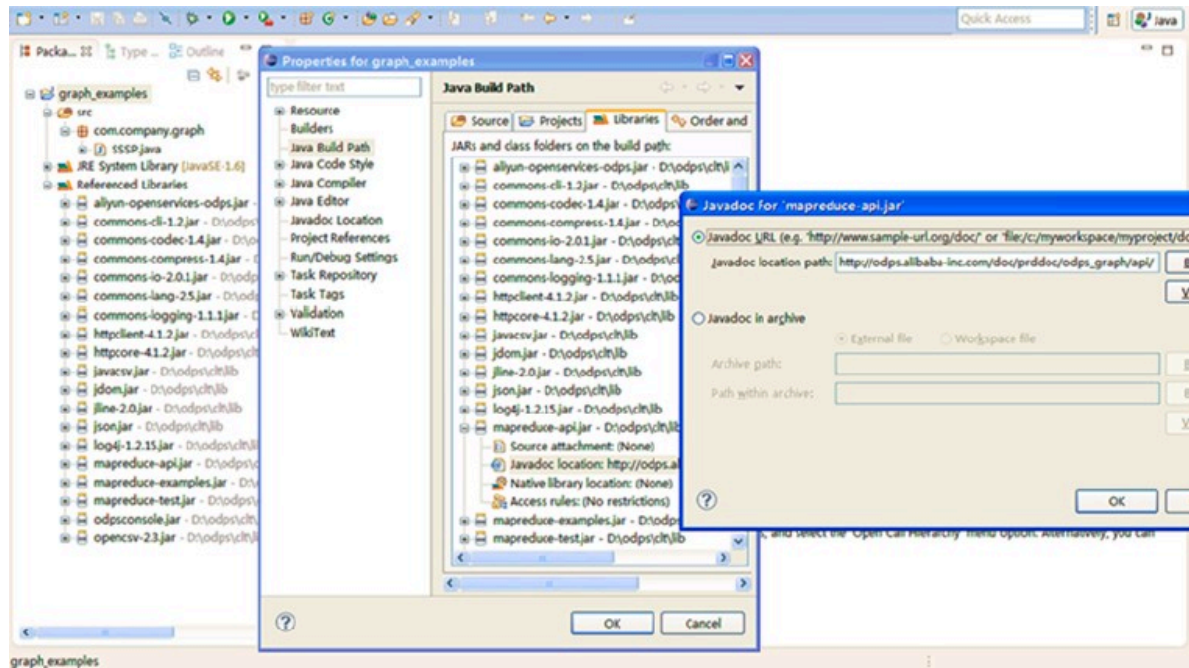
This topic uses the SSSP algorithm as an example to describe how to use Eclipse to develop and debug a Graph program. The development procedure is as follows:

Procedure

1. Create a Java project.

In this example, the project is `graph_examples`. Add the JAR package in the `lib` directory of the MaxCompute client to Build Path of the Eclipse project. The following figure shows a configured Eclipse project.

Figure 1-9: Create a Java project



2. Develop a MaxCompute Graph program.

In the actual development process, you can copy a sample program (such as SSSP) and then modify it as required. In this example, only the package path is changed to `package com.aliyun.odps.graph.example`.

3. Compile and build the package. In an Eclipse environment, right-click the source code directory (the `src` directory in the figure) and choose **Export > Java > JAR** file to generate a JAR package. Select the path for storing the target JAR package, such as `D:\odps\clt\odps-graph-example-sssp.jar`.

4. Use the MaxCompute client to run SSSP. For more information, see [Compile and run a Graph job](#).

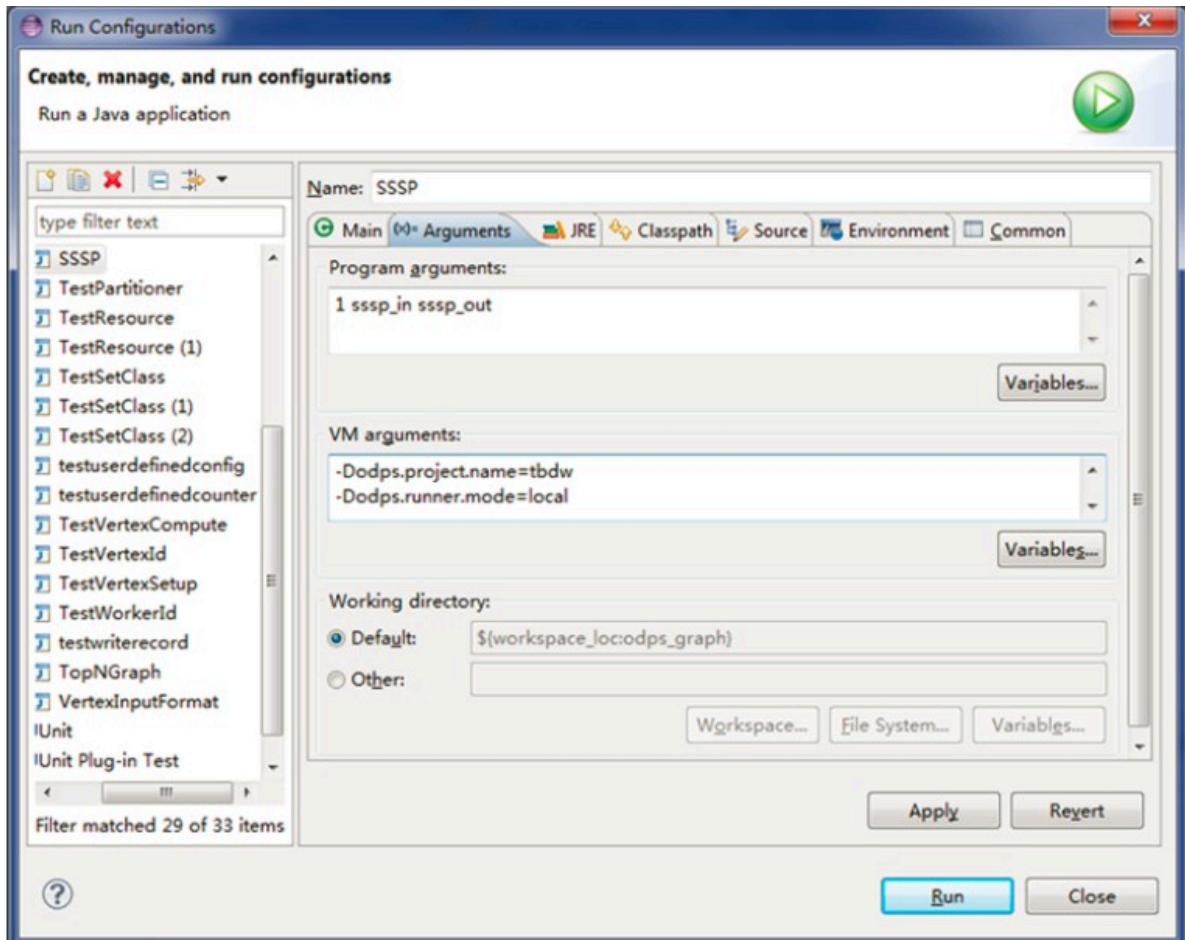
1.8.4.3 Local debugging

MaxCompute Graph supports the local debugging mode. You can use Eclipse for breakpoint debugging. The breakpoint debugging procedure is as follows:

Procedure

1. Select an Eclipse project. Right-click the Graph job main program file (the file that contains the main function), and configure its execution arguments, as shown in the following figure.

Figure 1-10: Local debugging



2. On the Arguments tab page, set the following arguments as the input parameters of the main program:

- **Program arguments:** 1 sssp_in sssp_out.
- **VM arguments:** Dodps.runner.mode=local, Dodps.project.name=<project .name>, Dodps.end.point=<end.point>, Dodps.access.id=<access.id>, and Dodps.access.key=<access.key>.
- **For the local mode (odps.end.point not specified), you need to create the sssp_in and sssp_out tables in the warehouse, and add the following data to sssp_in:**

```
1,"2:2,3:1,4:4"
2,"1:2,3:2,4:1"
3,"1:1,2:2,5:1"
4,"1:4,2:1,5:1"
```



```
5,"3:1,4:1"
```

**Note:**

For more information about the warehouse, see [Run MapReduce tasks locally](#).

3. Click Run to run SSSP on the local machine.

Refer to `conf/odps_config.ini` in the MaxCompute client for the settings of common parameters. The other parameters are described as follows:

- **odps.runner.mode:** The value is `local`. It is required for the local debugging feature.
- **odps.project.name:** specifies the current project, which is required.
- **odps.end.point:** specifies the current MaxCompute service address, which is optional. If this parameter is not specified, SSSP only reads metadata and data from the tables or resources in the warehouse. If the data does not exist, an error is returned. If this parameter is specified, SSSP first reads data from the warehouse. If the data does not exist, it reads data from the remote MaxCompute service.
- **odps.access.id:** specifies the AccessKey ID for accessing the MaxCompute service. It is valid only if `odps.end.point` is specified.
- **odps.access.key:** specifies the AccessKey secret for accessing the MaxCompute service. It is effective only if `odps.end.point` is specified.
- **odps.cache.resources:** specifies the resource list to be used. This parameter is the same as `-resources` of the `jar` command.
- **odps.local.warehouse:** specifies the local warehouse path. It is `./warehouse` by default.

The output is as follows:

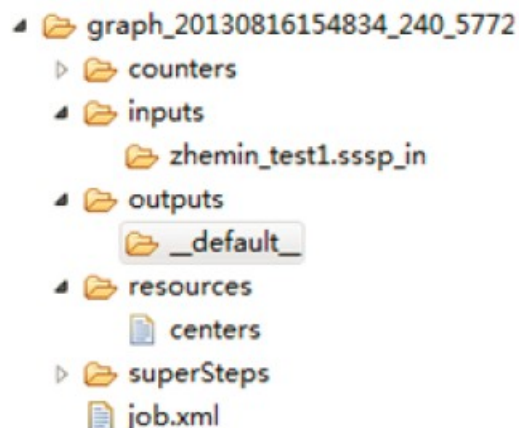
```
Counters: 3
com.aliyun.odps.graph.local.COUNTER
TASK_INPUT_BYTE=211
TASK_INPUT_RECORD=5
TASK_OUTPUT_BYTE=161
TASK_OUTPUT_RECORD=5
graph task finish
```

**Notice:**

In the preceding examples, the local warehouse must contain the `sssp_in` and `sssp_out` tables. For more information about `sssp_in` and `sssp_out`, see [Compile and run Graph](#).

1.8.4.4 Temporary directory for local jobs

Each time MaxCompute Graph runs a local debugging job, it creates a temporary directory in the Eclipse project directory, as shown in the following figure.



The temporary directory for a local Graph job contains the following directories and files:

- **counters:** stores the counter information that is generated during the running of the job.
- **inputs:** stores the input data of the job. Graph first reads data from the local warehouse. If no data is found, Graph uses the MaxCompute SDK to read data from the server (if `odps.end.point` is configured). An input operation reads 10 records by default. You can use the `Dodps.mapred.local.record.limit` parameter to modify the number of records read during each input operation. Up to 10,000 records can be read each time.
- **outputs:** stores the output data of the job. If there is an output table in the local warehouse, the results in outputs will overwrite data in that table after the job is executed.
- **resources:** stores the resources used by the job. Similarly, Graph first reads data from the local warehouse. If no data is found, Graph uses the MaxCompute SDK to read data from the server (if `odps.end.point` is configured).
- **job.xml:** stores job configurations.
- **superstep:** stores persistent message information from each iteration.

**Notice:**

If you need to output detailed logs during local debugging, place the log4j configuration file named `log4j.properties_odps_graph_cluster_debug` in the `src` directory.

1.8.4.5 Cluster debugging

After you perform local debugging, you can perform the following steps to submit a job for cluster testing:

1. Configure the MaxCompute client.
2. Run the `add jar /path/work.jar -f;` command to update the JAR package.
3. Run a JAR command to execute the job and check the operation log and command output.

**Notice:**

For more information about running a Graph job in a cluster, see [Compile and run a Graph job](#).

1.8.4.6 Performance optimization

1.8.4.6.1 Configure job parameters

The following table lists the GraphJob configuration parameters that affect job performance.

Table 1-50: GraphJob configuration parameters

Parameter	Description
<code>setSplitSize(long)</code>	Indicates the split size in MB. The value must be greater than 0. Default value: 64.
<code>setNumWorkers(int)</code>	Indicates the number of job workers. Value range: 1 to 1,000. Default value: 1. The number of workers is determined by the number of input bytes and split size.
<code>setWorkerCPU(int)</code>	Indicates the CPU resources for a map job. 100 resources are equivalent to one CPU core. Value range: 50 to 800. Default value: 200.
<code>setWorkerMemory(int)</code>	Indicates the memory resources in MB for a map job. Value range: 256M to 12G. Default value: 4,096.

Parameter	Description
<code>setMaxIteration(int)</code>	Indicates the maximum number of iterations. Default value: -1. If the value is equal to or smaller than 0, the job is not terminated after the maximum number of iterations.
<code>setJobPriority(int)</code>	Indicates the job priority. Value range: 0 to 9. Default value: 9. A greater value indicates a lower priority.

Recommendations:

1. Use `setNumWorkers` to increase the number of workers.
2. Use `setSplitSize` to reduce the split size and increase the data loading speed.
3. Use `setWorkerCPU` or `setWorkerMemory` to increase the CPU or memory resources for workers.
4. Use `setMaxIteration` to set the maximum number of iterations. For applications that do not require precise results, you can reduce the number of iterations to accelerate the iterating process.

Use `setNumWorkers` and `setSplitSize` together to accelerate data loading. If `setNumWorkers` is `workerNum`, `setSplitSize` is `splitSize`, and the total number of input bytes is `inputSize`, `splitNum` equals `inputSize` divided by `splitSize`. The relationship between `workerNum` and `splitNum` is as follows:

1. If `splitNum` is equal to `workerNum`, each worker loads one split.
2. If `splitNum` is greater than `workerNum`, each worker loads one or more splits.
3. If `splitNum` is smaller than `workerNum`, each worker loads zero or one split.

Therefore, you can adjust `workerNum` and `splitSize` to obtain a suitable loading speed. In the first two cases, data loading is faster. In the iteration phase, you only need to adjust `workerNum`. If you set runtime partitioning to false, we recommend that you either use `setSplitSize` to adjust the number of workers, or ensure the conditions in the first two cases are met. In the third case, the number of vertices in some of the workers is 0. In this case, you can use `set odps.graph.split.size=<m>; set odps.graph.worker.num=<n>;` before the JAR command, which achieves the same effect as `setNumWorkers` and `setSplitSize`.

Another common performance problem is data skew. As indicated by the counters, some workers process much more vertices or edges than others.

Data skew usually occurs when the number of vertices, edges, or messages corresponding to certain keys is far greater than that corresponding to other keys. These keys are distributed for processing by a small number of workers, resulting in long execution time of these workers. You can use the following methods to resolve this issue:

- Use Combiner to aggregate the messages of vertices corresponding to the keys, to reduce the number of generated messages.
- Improve the business logic.

1.8.4.6.2 Use Combiner

Developers can set Combiner to reduce the memory and network traffic consumed by message storage, and reduce job execution time. For more information, see the Combiner description in [Graph SDK overview](#).

1.8.4.6.3 Reduce data input

If a disk stores large volumes of data, reading data from the disk may prolong the processing time. You can reduce the number of bytes to be read to increase the overall throughput and improve job performance. You can use either of the following methods:

- Reduce data input: For some decision-making applications, processing sampled data only affects the precision of the results, not the overall accuracy. In this case, you can sample specific data to the input table for further processing.
- Avoid reading unnecessary fields: The TableInfo class provided in the MaxCompute Graph framework can read specific columns (sent through a column name array), instead of the entire table or partition. This effectively reduces the amount of input data and improves job performance.

1.8.4.6.4 JAR packages

The following JAR packages are loaded on the JVM that runs Graph programs by default. You do not need to manually upload these resources, or use -libjars to specify them in a command.

- commons-codec-1.3.jar
- commons-io-2.0.1.jar
- commons-lang-2.5.jar
- commons-logging-1.0.4.jar

- commons-logging-api-1.0.4.jar
- guava-14.0.jar
- json.jar
- log4j-1.2.15.jar
- slf4j-api-1.4.3.jar
- slf4j-log4j12-1.4.3.jar
- xmlenc-0.52.jar



Notice:

In CLASSPATH of the running JVM, the preceding JAR packages are loaded before your JAR package. This may cause version conflicts. For example, your program calls a certain class function of commons-codec-1.5.jar, but the function is not included in the current MaxCompute packages. In this case, you can choose to call a similar function in version 1.3 or wait until MaxCompute is upgraded to the required version.

1.8.5 Application limits

- Each job can reference up to 256 resources. Each table or archive is considered as one unit.
- The total resource size referenced by a job cannot exceed 512 MB.
- Each job can have up to 1,024 inputs (the number of input tables cannot exceed 64). Each job can have up to 256 outputs.
- Labels specified for multiple outputs cannot be null or empty strings. The label length cannot exceed 256, and the label can contain only upper-case letters (A to Z), lower-case letters (a to z), digits (0 to 9), underlines (_), pound signs (#), periods (.), and hyphens (-).
- The number of custom counters in a job cannot exceed 64. The counter group name and counter name cannot contain pound signs (#), and the total length of both names cannot exceed 100.
- The number of workers for each job is calculated by the framework. The maximum number of workers is 1,000. An error is returned when the number of workers exceeds this value.
- Each worker consumes 200 units of CPU resources by default. The range of resources consumed is 50 to 800.

- Each worker consumes 4,096 MB memory by default. The range of memory consumed is 256 MB to 12 GB.
- Each worker can read from a single resource up to 64 times.
- The split size is 64 MB by default, but can be defined by the user. The split size must be greater than 0, and the maximum value is in the range of 20 to 9223372036854775807.
- GraphLoader/Vertex/Aggregator in MaxCompute Graph are restricted by Java Sandbox (however, the main program of a Graph job is not subject to this restriction) while they run in a cluster. For more information, see [Java sandbox restrictions](#).

1.8.6 Sample programs

1.8.6.1 SSSP

Dijkstra's algorithm is a typical algorithm for calculating the Single Source Shortest Path (SSSP) in a directed graph.

Shortest path: For a weighted directed graph $G = (V, E)$, many paths are available from source vertex s to sink vertex v . The path with the smallest sum of edge weights is called the shortest path from s to v . The algorithm is implemented as follows:

- **Initialization:** The distance from s to s is 0 ($d[s] = 0$), and the distance from u to s is infinite ($d[u] = \infty$).
- **Iteration:** If an edge from u to v exists, the shortest distance from s to v is updated to $d[v] = \min(d[v], d[u] + \text{weight}(u, v))$. The iteration ends until the distance from all vertices to s does not change.



Note:

The implementation process determines that the algorithm is applicable to the MaxCompute Graph program. Each vertex maintains the current shortest distance to the source vertex. If the value changes, a message containing the new value and the edge weight is sent to the adjacent vertices. In the next iteration, the adjacent vertices update the current shortest distance based on the received message. The iteration ends when the current shortest distance values of all vertices do not change.

Example:

```

import java.io.IOException;
import com.aliyun.odps.io.WritableRecord;
import com.aliyun.odps.graph.Combiner;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.Edge;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.data.TableInfo;
public class SSSP {
    public static final String START_VERTEX = "sssp.start.vertex.id";
    /**Define SSSPVertex, where:
    * The vertex value indicates the current shortest distance from this
    vertex to source vertex startVertexId.
    * The compute() method uses the iteration formula  $d[v] = \min(d[v], d[u] + \text{weight}(u, v))$  to update the vertex value.
    * The cleanup() method writes the vertex and its shortest distance to
    the source vertex to the result table.
    */
    public static class SSSPVertex extends
    Vertex<LongWritable, LongWritable, LongWritable, LongWritable> {
        private static long startVertexId = -1;
        public SSSPVertex() {
            this.setValue(new LongWritable(Long.MAX_VALUE));
        }
        public boolean isStartVertex(
        ComputeContext<LongWritable, LongWritable, LongWritable, LongWritable> context) {
            if (startVertexId == -1) {
                String s = context.getConfiguration().get(START_VERTEX);
                startVertexId = Long.parseLong(s);
            }
            return getId().get() == startVertexId;
        }
        @Override
        public void compute(
        ComputeContext<LongWritable, LongWritable, LongWritable, LongWritable> context,
        Iterable<LongWritable> messages) throws IOException {
            long minDist = isStartVertex(context) ? 0 : Integer.MAX_VALUE;
            for (LongWritable msg : messages) { if (msg.get() < minDist) {
                minDist = msg.get();
            }
            }
            if (minDist < this.getValue().get()) {
                this.setValue(new LongWritable(minDist));
                if (hasEdges()) {
                    for (Edge<LongWritable, LongWritable> e : this.getEdges()) {
                        context.sendMessage(e.getDestVertexId(), new LongWritable(minDist + e.getValue().get()));
                    }
                } else {
                    voteToHalt();
                }
                // If the vertex value does not change, voteToHalt() is called to
                notify the framework that this vertex enters the halted state. The
                calculation ends when all vertices enter the halted state.
            }
        }
    }
}

```



```
@Override
public void cleanup(
WorkerContext<LongWritable, LongWritable, LongWritable, LongWritable>
context) throws IOException {
context.write(getId(), getValue());
}
}
/** Define MinLongCombiner and combine messages sent to the same
vertex to optimize performance and reduce memory usage.**/
public static class MinLongCombiner extends
Combiner<LongWritable, LongWritable> {
@Override
public void combine(LongWritable vertexId, LongWritable combinedMe
ssage, LongWritable messageToCombine) throws IOException {
if (combinedMessage.get() > messageToCombine.get()) {
combinedMessage.set(messageToCombine.get());
}
}
}
/** Define the SSSPVertexReader class, load a graph, and parse each
record in the table into a vertex. The first column of the record is
the vertex ID, and the second column stores all edge sets starting
from the vertex, such as 2:2,3:1,4:4.**/
public static class SSSPVertexReader extends
GraphLoader<LongWritable, LongWritable, LongWritable, LongWritable> {
@Override
public void load(LongWritable recordNum, WritableRecord record,
MutationContext<LongWritable, LongWritable, LongWritable, LongWritable>
context) throws IOException {
SSSPVertex vertex = new SSSPVertex();
vertex.setId((LongWritable) record.get(0));
String[] edges = record.get(1).toString().split(",");
for (int i = 0; i < edges.length; i++) {
String[] ss = edges[i].split(":");
vertex.addEdge(new LongWritable(Long.parseLong(ss[0])), new LongWritab
le(Long.parseLong(ss[1])));
}
context.addVertexRequest(vertex);
}
}
public static void main(String[] args) throws IOException { if (args.
length < 2) {
System.out.println("Usage: <startnode> <input> <output>");
System.exit(-1);
}
GraphJob job = new GraphJob();
// Define GraphJob, specify the implementation of Vertex/GraphLoader/
Combiner, and specify input and output tables.
job.setGraphLoaderClass(SSSPVertexReader.class);
job.setVertexClass(SSSPVertex.class);
job.setCombinerClass(MinLongCombiner.class);
job.set(START_VERTEX, args[0]);
job.addInput(TableInfo.builder().tableName(args[1]).build());
job.addOutput(TableInfo.builder().tableName(args[2]).build());
long startTime = System.currentTimeMillis(); job.run();
System.out.println("Job Finished in " + (System.currentTimeMillis() -
startTime) / 1000.0 + " seconds");
}
```

}

1.8.6.2 PageRank

PageRank is an algorithm for Web page ranking. For more information, see [PageRank](#). The input of the algorithm is a digraph G, where Vertex represents pages. If there is a link between page A to page B, there is an Edge linking A and B. Basic principles of the algorithm are as follows:

- **Initialization:** Vertex value means rank value of PageRank (double type). Initially, the value of all Vertices is $1/\text{TotalNumVertices}$.
- **Iteration formula:** $\text{PageRank}(i) = 0.15/\text{TotalNumVertices} + 0.85 \times \text{sum}$. Sum indicates the sum of $\text{PageRank}(j)/\text{out_degree}(j)$. (j indicates all vertices pointing to vertex i.)



Note:

The PageRank algorithm is best suited to be run on MaxCompute Graph as each j point maintains its PageRank value and sends $\text{PageRank}(j)/\text{out_degree}(j)$ to its adjacent vertex (to vote it) per iteration. Upon the next iteration, each vertex recalculates the PageRank value using the iteration formula.

Example:

```
import java.io.IOException;
import org.apache.log4j.Logger;
import com.aliyun.odps.io.WritableRecord;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.DoubleWritable;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.io.Text;
import com.aliyun.odps.io.Writable;
public class PageRank {
    private final static Logger LOG = Logger.getLogger(PageRank.class);
    /**
     * Defines PageRankVertex, where:
     * The vertex value indicates the current PageRank value of the vertex
     * (web page).
     * The compute() method uses the iteration formula  $\text{PageRank}(i) = 0.15/\text{TotalNumVertices} + 0.85 \times \text{sum}$  to update the vertex value.
     * The cleanup() method writes the vertex and its PageRank value to the
     * result table.
     */
    public static class PageRankVertex extends
        Vertex<Text, DoubleWritable, NullWritable, DoubleWritable> {
```

```
@Override
public void compute(
    ComputeContext<Text, DoubleWritable, NullWritable, DoubleWritable>
    context, Iterable<DoubleWritable> messages) throws IOException {
    if (context.getSuperstep() == 0) {
        setValue(new DoubleWritable(1.0 / context.getTotalNumVertices()));
    } else if (context.getSuperstep() >= 1) { double sum = 0;
        for (DoubleWritable msg : messages) { sum += msg.get();
        }
        DoubleWritable vertexValue = new DoubleWritable( (0.15f / context.
        getTotalNumVertices()) + 0.85f * sum);
        setValue(vertexValue);
    }
    if (hasEdges()) {
        context.sendMessageToNeighbors(this, new DoubleWritable(getValue().
        get() / getEdges().size()));
    }
}

@Override
public void cleanup(
    WorkerContext<Text, DoubleWritable, NullWritable, DoubleWritable>
    context) throws IOException {
    context.write(getId(), getValue());
}

/** Define the PageRankVertexReader class, load a graph, and resolve
    each record in the table into a vertex. The first column of the record
    is the start vertex and other columns are the destination vertices
    .**/
public static class PageRankVertexReader extends
    GraphLoader<Text, DoubleWritable, NullWritable, DoubleWritable> {
    @Override public void load(
        LongWritable recordNum, WritableRecord record,
        MutationContext<Text, DoubleWritable, NullWritable, DoubleWritable>
        context) throws IOException {
        PageRankVertex vertex = new PageRankVertex();
        vertex.setValue(new DoubleWritable(0));
        vertex.setId((Text) record.get(0));
        System.out.println(record.get(0));
        for (int i = 1; i < record.size(); i++) {
            Writable edge = record.get(i);
            System.out.println(edge.toString());
            if (!(edge.equals(NullWritable.get()))) {
                vertex.addEdge(new Text(edge.toString()), NullWritable.get());
            }
        }
        LOG.info("vertex eds size: " + (vertex.hasEdges() ? vertex.getEdges
        ().size() : 0));
        context.addVertexRequest(vertex);
    }
}

private static void printUsage() {
    System.out.println("Usage: <in> <out> [Max iterations (default 30
    )]");
    System.exit(-1);
}

public static void main(String[] args) throws IOException { if (args.
length < 2)
    printUsage();
    GraphJob job = new GraphJob();
    // Define GraphJob and specify the implementation method of Vertex/
    GraphLoader, the maximum number of iterations (> 30 by default), and
    input and output tables.
    job.setGraphLoaderClass(PageRankVertexReader.class);
```

```

job.setVertexClass(PageRankVertex.class);
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());
// default max iteration is 30
job.setMaxIteration(30); if (args.length >= 3)
job.setMaxIteration(Integer.parseInt(args[2]));
long startTime = System.currentTimeMillis(); job.run();
System.out.println("Job Finished in "
+ (System.currentTimeMillis() - startTime) / 1000.0 + " seconds");
}
}

```

1.8.6.3 K-means clustering

K-means clustering is a basic macro-clustering algorithm. Basic principles of the K-means clustering algorithm are as follows: Clustering is performed around k points in space, and the closest vertices are classified. The values of the clustering centers are successively updated through iterations until the optimal clustering result is obtained.

Assuming the sample set is divided into k sets or categories, the steps in the algorithm are as follows:

- **Select initial center of k classes.**
- **In the ith iteration, select any sample, solve its path to k center, and then classify the sample into the class of shortest path to center.**
- **Use the mean method to update the center value of the class.**
- **For all k clustering centers, if the value remains unchanged or is less than a certain threshold after iteration of the first two steps, the iteration ends. Otherwise, the iteration continues.**

Example:

```

import java.io.DataInput; import java.io.DataOutput;
import java.io.IOException;
import org.apache.log4j.Logger;
import com.aliyun.odps.io.WritableRecord;
import com.aliyun.odps.graph.Aggregator;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.DoubleWritable;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.io.Text;
import com.aliyun.odps.io.Tuple;
import com.aliyun.odps.io.Writable;
public class Kmeans {

```

```
private final static Logger LOG = Logger.getLogger(Kmeans.class);
/**Define KmeansVertex. The compute() method is simple. It calls the
aggregate() method of the context object and transmits the value of
the current vertex (in Tuple type and expressed by vector).**/
public static class KmeansVertex extends
Vertex<Text, Tuple, NullWritable, NullWritable> {
@Override
public void compute(
ComputeContext<Text, Tuple, NullWritable, NullWritable> context
, Iterable<NullWritable> messages) throws IOException { context.
aggregate(getValue());
}
}
/** Define the KmeansVertexReader class, load a graph, and parse each
record in the table as a vertex. The vertex ID does not matter, and
transmitted recordNum is used as the ID. The vertex value is the Tuple
consisting of all columns of the record.**/
public static class KmeansVertexReader extends
GraphLoader<Text, Tuple, NullWritable, NullWritable> {
@Override
public void load(LongWritable recordNum, WritableRecord record,
MutationContext<Text, Tuple, NullWritable, NullWritable> context)
throws IOException {
KmeansVertex vertex = new KmeansVertex();
vertex.setId(new Text(String.valueOf(recordNum.get())));
vertex.setValue(new Tuple(record.getAll()));
context.addVertexRequest(vertex);
}
}
public static class KmeansAggrValue implements Writable {
Tuple centers = new Tuple();
Tuple sums = new Tuple();
Tuple counts = new Tuple();
@Override
public void write(DataOutput out) throws IOException {
centers.write(out);
sums.write(out); counts.write(out);
}
@Override
public void readFields(DataInput in) throws IOException {
centers = new Tuple();
centers.readFields(in);
sums = new Tuple();
sums.readFields(in);
counts = new Tuple();
counts.readFields(in);
}
@Override
public String toString() {
return "centers " + centers.toString() + ", sums " + sums.toString()
+ ", counts " + counts.toString();
}
}
/**
* Defines KmeansAggregator. This class encapsulates the main logic of
the Kmeans algorithm, where,
* createInitialValue creates an initial value for each iteration (k-
class center point). In first iteration (superstep equals to 0), the
value is the initial center point. Otherwise, the value is the new
center point when the last iteration ends.
* The aggregate() method calculates the distance from each vertex to
centers of different classes, classifies the vertex as the class of
the nearest center, and updates sum and count of the class.
```

```

* The merge() method combines sums and counts collected by each Worker
.
* The terminate() method calculates the new central point based on
the sum and count of each class. If the distance between the new and
old central points is less than a threshold value or the number of
iterations reaches the upper limit, the iteration ends (false is
returned). The final central point is written to the resulting table.
**/
public static class KmeansAggregator extends Aggregator<KmeansAggr
Value> {
@SuppressWarnings("rawtypes")
@Override
public KmeansAggrValue createInitialValue(WorkerContext context)
throws IOException {
KmeansAggrValue aggrVal = null;
if (context.getSuperstep() == 0) {
aggrVal = new KmeansAggrValue();
aggrVal.centers = new Tuple();
aggrVal.sums = new Tuple();
aggrVal.counts = new Tuple();
byte[] centers = context.readCacheFile("centers");
String lines[] = new String(centers).split("\n");
for (int i = 0;
i < lines.length; i++) { String[] ss = lines[i].split(",");
Tuple center = new Tuple();
Tuple sum = new Tuple();
for (int j = 0; j < ss.length; ++j) {
center.append(new DoubleWritable(Double.valueOf(ss[j].trim())));
sum.append(new DoubleWritable(0.0));
}
LongWritable count = new LongWritable(0);
aggrVal.sums.append(sum); aggrVal.counts.append(count);
aggrVal.centers.append(center);
}
} else {
aggrVal = (KmeansAggrValue) context.getLastAggregatedValue(0);
}
return aggrVal;
}
@Override
Public void aggregate (glasvalue, object item ){
int min = 0;
double mindist = Double.MAX_VALUE;
Tuple point = (Tuple) item;
for (int i = 0;
i < value.centers.size();
i++) { Tuple center = (Tuple) value.centers.get(i);
// use Euclidean Distance, no need to calculate sqrt
double dist = 0.0d;
for (int j = 0; j < center.size(); j++) {
double v = ((DoubleWritable) point.get(j)).get()
- ((DoubleWritable) center.get(j)).get();
dist += v * v;
}
if (dist < mindist) { mindist = dist; min = i;
}
}
// update sum and count
Tuple sum = (Tuple) value.sums.get(min);
for (int i = 0;
i < point.size(); i++) {
DoubleWritable s = (DoubleWritable) sum.get(i); s.set(s.get() + ((
DoubleWritable) point.get(i)).get());
}
}

```

```
LongWritable count = (LongWritable) value.counts.get(min);
count.set(count.get() + 1);
}
@Override
public void merge(KmeansAggrValue value, KmeansAggrValue partial) {
    for (int i = 0; i < value.sums.size(); i++) {
        Tuple sum = (Tuple) value.sums.get(i);
        Tuple that = (Tuple) partial.sums.get(i);
        for (int j = 0; j < sum.size(); j++) {
            DoubleWritable s = (DoubleWritable) sum.get(j);
            s.set(s.get() + ((DoubleWritable) that.get(j)).get());
        }
    }
    for (int i = 0; i < value.counts.size(); i++) {
        LongWritable count = (LongWritable) value.counts.get(i);
        count.set(count.get() + ((LongWritable) partial.counts.get(i)).get());
    }
}
@SuppressWarnings("rawtypes")
@Override
public boolean terminate(WorkerContext context, KmeansAggrValue value
) throws IOException {
    // compute new centers
    Tuple newCenters = new Tuple(value.sums.size());
    for (int i = 0; i < value.sums.size(); i++) {
        Tuple sum = (Tuple) value.sums.get(i);
        Tuple newCenter = new Tuple(sum.size());
        LongWritable c = (LongWritable) value.counts.get(i);
        for (int j = 0; j < sum.size(); j++) {
            DoubleWritable s = (DoubleWritable) sum.get(j);
            double val = s.get() / c.get();
            newCenter.set(j, new DoubleWritable(val));
        }
        // reset sum for next iteration
        s.set(0.0d);
    }
    // reset count for next iteration
    c.set(0);
    newCenters.set(i, newCenter);
}
// update centers
Tuple oldCenters = value.centers; value.centers = newCenters;
LOG.info("old centers: " + oldCenters + ", new centers: " + newCenters
);
// compare new/old centers
boolean converged = true;
for (int i = 0; i < value.centers.size() && converged; i++) {
    Tuple oldCenter = (Tuple) oldCenters.get(i);
    Tuple newCenter = (Tuple) newCenters.get(i); double sum = 0.0d;
    for (int j = 0; j < newCenter.size(); j++) {
        double v = ((DoubleWritable) newCenter.get(j)).get() - ((DoubleWrit
        able) oldCenter.get(j)).get();
        sum += v * v;
    }
    double dist = Math.sqrt(sum);
    LOG.info("old center: " + oldCenter + ", new center: " + newCenter +
        ", dist: " + dist);
    // converge threshold for each center: 0.05
    converged = dist < 0.05d;
}
if (converged || context.getSuperstep() == context.getMaxIteration()
    - 1) {
    // converged or reach max iteration, output centers
    for (int i = 0; i < value.centers.size(); i++) { context.write(((Tuple
    ) value.centers.get(i)).toArray());
```

```

}
// true means to terminate iteration
return true;
}
// false means to continue iteration
return false;
}
}
private static void printUsage() {
System.out.println("Usage: <in> <out> [Max iterations (default 30)]");
System.exit(-1);
}
/**Define GraphJob, and specify the implementation method of Vertex,
GraphLoader, or Aggregator, the maximum number of iterations (30 by
default), and input and output tables. */
public static void main(String[] args) throws IOException {
if (args.length < 2)
printUsage();
GraphJob job = new GraphJob();
job.setGraphLoaderClass(KmeansVertexReader.class);
job.setRuntimePartitioning(false);
// Specify job.setRuntimePartitioning(false). For the K-means
algorithm, vertices do not need to be distributed during graph loading
. If RuntimePartitioning is set to false, the performance for graph
loading is improved.
job.setVertexClass(KmeansVertex.class);
job.setAggregatorClass(KmeansAggregator.class);
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());
// default max iteration is 30
job.setMaxIteration(30); if (args.length >= 3)
job.setMaxIteration(Integer.parseInt(args[2]));
long start = System.currentTimeMillis(); job.run();
System.out.println("Job Finished in " + (System.currentTimeMillis() -
start) / 1000.0 + " seconds");
}
}

```

1.8.6.4 BiPartiteMatching

In a bipartite graph, all vertices can be divided into two sets, to which the two vertices of each edge respectively belong. For a bipartite graph G , M is its subgraph. If any two edges of M 's edge set are not attached to the same vertex, then M is a match. The bipartite graph matching is usually used for information matching in scenarios with clear supply and demand relationships (such as online dating websites).

The procedure is as follows:

- Start from the first vertex on the left, select unmatched vertex to search for augmented path.
- If it goes through an unmatched vertex, the search is successful.
- Update path information, match number of Edge +1, and stop searching.
- If no augmented path is found, it does not search again from the specific vertex.

Example:

```
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import java.util.Random;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.io.Text;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.io.WritableRecord;
public class BipartiteMatching {
    private static final Text UNMATCHED = new Text("UNMATCHED");
    public static class TextPair implements Writable {
        public Text first; public Text second;
        public TextPair() { first = new Text();
            second = new Text();
        }
        public TextPair(Text first, Text second) {
            this.first = new Text(first);
            this.second = new Text(second);
        }
        @Override
        public void write(DataOutput out) throws IOException {
            first.write(out);
            second.write(out);
        }
        @Override
        public void readFields(DataInput in) throws IOException {
            first = new Text();
            first.readFields(in);
            second = new Text();
            second.readFields(in);
        }
        @Override
        public String toString() { return first + ": " + second;
        }
    }
    public static class BipartiteMatchingVertexReader extends
        GraphLoader<Text, TextPair, NullWritable, Text> {
        @Override
        public void load(LongWritable recordNum, WritableRecord record,
            MutationContext<Text, TextPair, NullWritable, Text> context) throws
            IOException {
            BipartiteMatchingVertex vertex = new BipartiteMatchingVertex();
            vertex.setId((Text) record.get(0));
            vertex.setValue(new TextPair(UNMATCHED, (Text) record.get(1)));
            String[] adj = record.get(2).toString().split(",");
            for (String adj:adj) {
                vertex.addEdge(new Text(adj), null);
            }
            context.addVertexRequest(vertex);
        }
    }
    public static class BipartiteMatchingVertex extends
        Vertex<Text, TextPair, NullWritable, Text> {
```

```
private static final Text LEFT = new Text("LEFT");
private static final Text RIGHT = new Text("RIGHT");
private static Random rand = new Random();
@Override
public void compute(
    ComputeContext<Text, TextPair, NullWritable, Text> context, Iterable<
    Text> messages) throws IOException {
    if (isMatched()) { voteToHalt();
    return;
    }
    switch ((int) context.getSuperstep() % 4) {
    case 0:
        if (isLeft()) {
            context.sendMessageToNeighbors(this, getId());
        }
        break;
    case 1:
        if (isRight()) {
            Text luckyLeft = null;
            for (Text message : messages) { if (luckyLeft == null) {
            luckyLeft = new Text(message);
            } else {
            if (rand.nextInt(1) == 0) { luckyLeft.set(message);
            }
            }
            }
            if (luckyLeft != null) { context.sendMessage(luckyLeft, getId());
            }
        }
        break;
    case 2:
        if (isLeft()) {
            Text luckyRight = null;
            for (Text msg : messages) { if (luckyRight == null) {
            luckyRight = new Text(msg);
            } else {
            if (rand.nextInt(1) == 0) { luckyRight.set(msg);
            }
            }
            }
            if (luckyRight != null) {
            setMatchVertex(luckyRight);
            context.sendMessage(luckyRight, getId());
            }
        }
        break; case 3:
        if (isRight()) {
            for (Text msg : messages) { setMatchVertex(msg);
            }
        }
        break;
    }
}
@Override
public void cleanup(
    WorkerContext<Text, TextPair, NullWritable, Text> context) throws
    IOException {
    context.write(getId(), getValue().first);
}
private boolean isMatched() {
    return ! getValue().first.equals(UNMATCHED);
}
private boolean isLeft() {
    return getValue().second.equals(LEFT);
}
```

```
}  
private boolean isRight() {  
    return getValue().second.equals(RIGHT);  
}  
private void setMatchVertex(Text matchVertex) { getValue().first.set(  
    matchVertex);  
}  
}  
private static void printUsage() {  
    System.err.println("BipartiteMatching <input> <output> [maxIteration  
    ]");  
}  
public static void main(String[] args) throws IOException { if (args.  
    length < 2) {  
        printUsage();  
    }  
    GraphJob job = new GraphJob();  
    job.setGraphLoaderClass(BipartiteMatchingVertexReader.class);  
    job.setVertexClass(BipartiteMatchingVertex.class);  
    job.addInput(TableInfo.builder().tableName(args[0]).build());  
    job.addOutput(TableInfo.builder().tableName(args[1]).build());  
    int maxIteration = 30;  
    if (args.length > 2) {  
        maxIteration = Integer.parseInt(args[2]);  
    }  
    job.setMaxIteration(maxIteration);  
    job.run();  
}
```

1.8.6.5 Strongly-connected component

A directed graph is called a strongly-connected graph if every vertex is reachable from every other vertex. A strongly-connected sub-graph with a large number of vertices in a directed graph is called a strongly-connected component. This algorithm example is based on the parallel coloring algorithm.

Each vertex contains the following two parts:

- **colorID:** stores the color of the vertex (v) during forward traversal. At the end of computing, the vertices with the same colorID belong to one strongly-connected component.
- **transposeNeighbors:** stores neighbor IDs of v in the transpose graph of the input graph.

The algorithm is implemented as follows:

- **Transpose graph formation:** contains two supersteps. In the first superstep, each vertex sends a message with its ID to all its outgoing neighbors. These IDs are stored in transposeNeighbors in the second superstep.

- **Trimming:** contains one superstep. Each vertex with only one incoming or outgoing edge sets its colorID to its own ID, and becomes inactive. Subsequent messages sent to these vertexes are ignored.
- **Forward traversal:** contains two subphases (supersteps): Start and Rest. In the Start phase, each vertex sets its colorID to its own ID, and sends the ID to outgoing neighbors. In the Rest phase, each vertex uses the maximum colorID it received to update its own colorID, and propagates the colorID until the colorIDs converge. When the colorIDs converge, the master process sets the phase to backward traversal.
- **Backward traversal:** contains two subphases, Start and Rest. In the Start phase, each vertex whose ID equals its colorID propagates its ID to the vertices in transposeNeighbors and sets its status as inactive. Subsequent messages sent to these vertexes are ignored. In each of the Rest phase supersteps, each vertex receives a message matching its colorID, propagates its colorID in the transpose graph, and sets its status as inactive. If there are still active vertices after this step, the process goes back to the trimming phase.

Example:

```
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.Aggregator;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.BooleanWritable;
import com.aliyun.odps.io.IntWritable;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.io.Tuple;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.io.WritableRecord;
/**
 * Definition from Wikipedia:
 * In the mathematical theory of directed graphs, a graph is said
 * to be strongly connected if every vertex is reachable from every
 * other vertex. The strongly connected components of an arbitrary
 * directed graph form a partition into subgraphs that are themselves
 * strongly connected.
 *
 * Algorithms with four phases as follows.
 * 1. Transpose Graph Formation: Requires two supersteps. In the first
 * superstep, each vertex sends a message with its ID to all its >
 * outgoing
```

```

* neighbors, which in the second superstep are stored > in transposeN
eighbors.
*
* 2. Trimming: Takes one superstep. Every vertex with only in-coming
or
* only outgoing edges (or neither) sets its colorID to its own ID and
* becomes inactive. Messages subsequently sent to the vertex > are
ignored.
*
* 3. Forward-Traversal: There are two sub phases: Start and Rest. In
the
* Start phase, each vertex sets its colorID to its own ID and >
propagates
* its ID to its outgoing neighbors. In the Rest phase, vertices update
* their own colorIDs with the minimum colorID they have seen, and >
propagate
* their colorIDs, if updated, until the colorIDs converge.
* Set the phase to Backward-Traversal when the colorIDs converge.
*
* 4. Backward-Traversal: We again break the phase into Start and Rest.
* In Start, every vertex whose ID equals its colorID propagates its ID
> to
* the vertices in transposeNeighbors and sets itself inactive. >
Messages
* subsequently sent to the vertex are ignored. In each of the Rest >
phase supersteps,
* each vertex receiving a message that matches its colorID: (1) >
propagates
* its colorID in the transpose graph; (2) sets itself inactive. >
Messages
* subsequently sent to the vertex are ignored. Set the phase back to
Trimming
* if not all vertex are inactive.
*
* http://ilpubs.stanford.edu:8090/1077/3/p535-salihoglu.pdf
**/
public class StronglyConnectedComponents {
public final static int STAGE_TRANSPOSE_1 = 0;
public final static int STAGE_TRANSPOSE_2 = 1;
public final static int STAGE_TRIMMING = 2;
public final static int STAGE_FW_START = 3;
public final static int STAGE_FW_REST = 4;
public final static int STAGE_BW_START = 5;
public final static int STAGE_BW_REST = 6;
/**
* The value is composed of component id, incoming neighbors,
* active status and updated status.
**/
public static class MyValue implements Writable {
LongWritable sccID;// strongly connected component id
Tuple inNeighbors; // transpose neighbors
BooleanWritable active; // vertex is active or not
BooleanWritable updated; // sccID is updated or not
public MyValue() {
this.sccID = new LongWritable(Long.MAX_VALUE);
this.inNeighbors = new Tuple();
this.active = new BooleanWritable(true);
this.updated = new BooleanWritable(false);
}
public void setSccID(LongWritable sccID) {
this.sccID = sccID;
}
public LongWritable getSccID() {
return this.sccID;
}
}

```

```
}
public void setInNeighbors(Tuple inNeighbors) {
    this.inNeighbors = inNeighbors;
}
public Tuple getInNeighbors() {
    return this.inNeighbors;
}
public void addInNeighbor(LongWritable neighbor) {
    this.inNeighbors.append(new LongWritable(neighbor.get()));
}
public boolean isActive() {
    return this.active.get();
}
public void setActive(boolean status) {
    this.active.set(status);
}
public boolean isUpdated() {
    return this.updated.get();
}
public void setUpdated(boolean update) {
    this.updated.set(update);
}
@Override
public void write(DataOutput out) throws IOException {
    this.sccID.write(out);
    this.inNeighbors.write(out);
    this.active.write(out);
    this.updated.write(out);
}
@Override
public void readFields(DataInput in) throws IOException {
    this.sccID.readFields(in);
    this.inNeighbors.readFields(in);
    this.active.readFields(in);
    this.updated.readFields(in);
}
@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("sccID: " + sccID.get());
    sb.append(" inNeighbores: " + inNeighbors.toDelimitedString(','));
    sb.append(" active: " + active.get());
    sb.append(" updated: " + updated.get());
    return sb.toString();
}
}
public static class SCCVertex extends
    Vertex<LongWritable, MyValue, NullWritable, LongWritable> {
    public SCCVertex() {
        this.setValue(new MyValue());
    }
    @Override
    public void compute(
        ComputeContext<LongWritable, MyValue, NullWritable, LongWritable>
        context, Iterable<LongWritable> msgs) throws IOException {
        // Messages sent to inactive vertex are ignored.
        if (! this.getValue().isActive()) {
            this.voteToHalt(); return;
        }
        int stage = ((SCCAggrValue)context.getLastAggregatedValue(0)).getStage
            (); switch (stage) {
        case STAGE_TRANSPOSE_1:
            context.sendMessageToNeighbors(this, this.getId());
            break;
        }
    }
}
```

```
case STAGE_TRANSPOSE_2:
for (LongWritable msg: msgs) {
this.getValue().addInNeighbor(msg);
}
case STAGE_TRIMMING:
this.getValue().setScCID(getId());
if (this.getValue().getInNeighbors().size() == 0 || this.getNumEdges()
== 0) {
this.getValue().setActive(false);
}
break;
case STAGE_FW_START: this.getValue().setScCID(getId());
context.sendMessageToNeighbors(this, this.getValue().getScCID());
break;
case STAGE_FW_REST:
long minScCID = Long.MAX_VALUE;
for (LongWritable msg : msgs) {
if (msg.get() < minScCID) { minScCID = msg.get();
}
}
if (minScCID < this.getValue().getScCID().get()) {
this.getValue().setScCID(new LongWritable(minScCID));
context.sendMessageToNeighbors(this, this.getValue().getScCID());
this.getValue().setUpdated(true);
} else {
this.getValue().setUpdated(false);
}
break;
case STAGE_BW_START:
if (this.getId().equals(this.getValue().getScCID())) {
for (Writable neighbor : this.getValue().getInNeighbors().getAll()) {
context.sendMessage((LongWritable)neighbor, this.getValue().getScCID
());
}
this.getValue().setActive(false);
}
break;
case STAGE_BW_REST: this.getValue().setUpdated(false);
for (LongWritable msg : msgs) {
if (msg.equals(this.getValue().getScCID())) {
for (Writable neighbor : this.getValue().getInNeighbors().getAll()) {
context.sendMessage((LongWritable)neighbor, this.getValue().getScCID
());
}
this.getValue().setActive(false);
this.getValue().setUpdated(true);
}
}
break;
}
context.aggregate(0, getValue());
}
@Override
public void cleanup(
WorkerContext<LongWritable, MyValue, NullWritable, LongWritable>
context)
throws IOException {
context.write(getId(), getValue().getScCID());
}
}
/**
* The SCCAggrValue maintains global stage and graph updated and >
active status.
```

```
* updated is true only if one vertex is updated.
* active is true only if one vertex is active.
*/
public static class SCCAggrValue implements Writable {
    IntWritable stage = new IntWritable(STAGE_TRANSPOSE_1);
    BooleanWritable updated = new BooleanWritable(false);
    BooleanWritable active = new BooleanWritable(false);
    public void setStage(int stage) { this.stage.set(stage);
    }
    public int getStage() { return this.stage.get();
    }
    public void setUpdated(boolean updated) {
        this.updated.set(updated);
    }
    public boolean getUpdated() {
        return this.updated.get();
    }
    public void setActive(boolean active) {
        this.active.set(active);
    }
    public boolean getActive() {
        return this.active.get();
    }
    @Override
    public void write(DataOutput out) throws IOException {
        this.stage.write(out);
        this.updated.write(out);
        this.active.write(out);
    }
    @Override
    public void readFields(DataInput in) throws IOException {
        this.stage.readFields(in);
        this.updated.readFields(in);
        this.active.readFields(in);
    }
}
/**
 * The job of SCCAggregator is to schedule global stage in > every
 * superstep.
 */
public static class SCCAggregator extends Aggregator<SCCAggrValue> {
    @SuppressWarnings("rawtypes")
    @Override
    public SCCAggrValue createStartupValue(WorkerContext context) throws
        IOException { return new SCCAggrValue();
    }
    @SuppressWarnings("rawtypes")
    @Override
    public SCCAggrValue createInitialValue(WorkerContext context) throws
        IOException {
        return (SCCAggrValue) context.getLastAggregatedValue(0);
    }
    @Override
    public void aggregate(SCCAggrValue value, Object item) throws
        IOException { MyValue v = (MyValue)item;
        if ((value.getStage() == STAGE_FW_REST || value.getStage() ==
            STAGE_BW_REST)&& v.isUpdated()) { value.setUpdated(true);
        }
        // only active vertex invoke aggregate()
        value.setActive(true);
    }
    @Override
    public void merge(SCCAggrValue value, SCCAggrValue partial) throws
        IOException {
```



```
boolean updated = value.getUpdated() || partial.getUpdated();
value.setUpdated(updated);
boolean active = value.getActive() || partial.getActive();
value.setActive(active);
}
@SuppressWarnings("rawtypes")
@Override
public boolean terminate(WorkerContext context, SCCAggrValue value)
throws IOException {
    // If all vertices is inactive, job is over.
    if (! value.getActive()) { return true;
    }
    // state machine
    switch (value.getStage()) {
    case STAGE_TRANSPOSE_1:value.setStage(STAGE_TRANSPOSE_2);
    break;
    case STAGE_TRANSPOSE_2:value.setStage(STAGE_TRIMMING);
    break;
    case STAGE_TRIMMING:value.setStage(STAGE_FW_START);
    break;
    case STAGE_FW_START: value.setStage(STAGE_FW_REST);
    break;
    case STAGE_FW_REST:if (value.getUpdated()) {
    value.setStage(STAGE_FW_REST);
    } else {
    value.setStage(STAGE_BW_START);
    }
    break;
    case STAGE_BW_START: value.setStage(STAGE_BW_REST);
    break;
    case STAGE_BW_REST:if (value.getUpdated()) { value.setStage(STAGE_BW_R
EST);
    } else { value.setStage(STAGE_TRIMMING);
    }
    break;
    }
    value.setActive(false);
    value.setUpdated(false);
    return false;
    }
}

public static class SCCVertexReader extends
GraphLoader<LongWritable, MyValue, NullWritable, LongWritable> {
    @Override public void load(
    LongWritable recordNum, WritableRecord record,
    MutationContext<LongWritable, MyValue, NullWritable, LongWritable>
    context) throws IOException {
    SCCVertex vertex = new SCCVertex();
    vertex.setId((LongWritable) record.get(0));
    String[] edges = record.get(1).toString().split(",");
    for (int i = 0; i < edges.length; i++) { try {
    long destID = Long.parseLong(edges[i]);
    vertex.addEdge(new LongWritable(destID), NullWritable.get());
    } catch (NumberFormatException nfe) { System.err.println("Ignore " +
nfe);
    }
    }
    context.addVertexRequest(vertex);
    }
}

public static void main(String[] args) throws Exception {
    if (args.length < 2) {
    System.out.println("Usage: <input> <output>");
    System.exit(-1);
    }
```

```

}
GraphJob job = new GraphJob();
job.setGraphLoaderClass(SCCVertexReader.class);
job.setVertexClass(SCCVertex.class);
job.setAggregatorClass(SCCAggregator.class);
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());
long startTime = System.currentTimeMillis();
job.run();
System.out.println("Job Finished in " + (System.currentTimeMillis() -
startTime) / 1000.0 + " seconds");
}
}

```

1.8.6.6 Connected component

Two vertices are connected if a path exists between them. Undirected graph G is called a connected graph if every two vertices in the graph are connected. Otherwise, G is called an unconnected graph. A connected sub-graph with a large number of vertices is called a connected component. This algorithm calculates connected component members of each vertex, and outputs the connected component of the vertex value that includes the smallest vertex ID. The smallest vertex ID is propagated along edges to all vertices of the connected component.

Example:

```

import java.io.IOException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.graph.examples.SSSP.MinLongCombiner;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.io.WritableRecord;
/**
 * Compute the connected component membership of each vertex and output
 * each vertex which's value containing the smallest id in the >
 * connected
 * component containing that vertex.
 *
 * Algorithm: propagate the smallest vertex id along the edges to all
 * vertices of a connected component.
 */
public class ConnectedComponents {
    public static class CCVertex extends
        Vertex<LongWritable, LongWritable, NullWritable, LongWritable> {
        @Override
        public void compute(
            ComputeContext<LongWritable, LongWritable, NullWritable, LongWritable>
            context, Iterable<LongWritable> msgs) throws IOException {
            if (context.getSuperstep() == 0L) {
                this.setValue(getId());
            }
        }
    }
}

```

```

context.sendMessageToNeighbors(this, getValue());
return;
}
long minID = Long.MAX_VALUE;
for (LongWritable id : msgs) {
    if (id.get() < minID) { minID = id.get();
    }
}
if (minID < this.getValue().get()) {
    this.setValue(new LongWritable(minID));
    context.sendMessageToNeighbors(this, getValue());
} else {
    this.voteToHalt();
}
}
}
/**
 * Output Table Description:
 * +-----+
 * Field | Type | Comment |
 * +-----+
 * v | bigint | vertex id |
 * minID | bigint | smallest id in the connected component |
 * +-----+
 */
@Override
public void cleanup(
    WorkerContext<LongWritable, LongWritable, NullWritable, LongWritable>
    context) throws IOException {
    context.write(getId(), getValue());
}
}
/**
 * Input Table Description:
 * +-----+
 * +-----+
 * Field | Type | Comment |
 * +-----+
 * +-----+
 * v | bigint | vertex id |
 * es | string | comma separated target vertex id of outgoing edges |
 * +-----+
 * +-----+
 *
 * Example:
 * For graph:
 * 1 ----- 2
 * | |
 * 3 ----- 4
 * Input table:
 * +-----+
 * v | es |
 * +-----+
 * | 1 | 2,3 |
 * | 2 | 1,4 |
 * | 3 | 1,4 |
 * | 4 | 2,3 |
 * +-----+
 */
public static class CCVertexReader extends
    GraphLoader<LongWritable, LongWritable, NullWritable, LongWritable> {
    @Override
    public void load(
        LongWritable recordNum, WritableRecord record,

```

```

MutationContext<LongWritable, LongWritable, NullWritable, LongWritable
> context) throws IOException {
    CCVertex vertex = new CCVertex();
    vertex.setId((LongWritable) record.get(0));
    String[] edges = record.get(1).toString().split(",");
    for (int i = 0; i < edges.length; i++) {
        long destID = Long.parseLong(edges[i]);
        vertex.addEdge(new LongWritable(destID), NullWritable.get());
    }
    context.addVertexRequest(vertex);
}
}
}
public static void main(String[] args) throws IOException {
    if (args.length < 2) {
        System.out.println("Usage: <input> <output>");
        System.exit(-1);
    }
    GraphJob job = new GraphJob();
    job.setGraphLoaderClass(CCVertexReader.class);
    job.setVertexClass(CCVertex.class);
    job.setCombinerClass(MinLongCombiner.class);
    job.addInput(TableInfo.builder().tableName(args[0]).build());
    job.addOutput(TableInfo.builder().tableName(args[1]).build());
    long startTime = System.currentTimeMillis();
    job.run();
    System.out.println("Job Finished in " + (System.currentTimeMillis() -
        startTime) / 1000.0 + " seconds");
}
}

```

1.8.6.7 Topological sorting

For a directed edge (u,v), all vertex sequences satisfying $u < v$ are called topological sequences. Topological sorting is an algorithm that is used to calculate the topological sequence of a directed graph.

The algorithm is implemented as follows:

- A vertex without incoming edges in the graph is found and output.
- The output vertex and all its outgoing edges are deleted.
- The preceding steps are repeated until all vertices are output.

Example:

```

import java.io.IOException;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.Aggregator;
import com.aliyun.odps.graph.Combiner;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;

```

```
import com.aliyun.odps.io.BooleanWritable;
import com.aliyun.odps.io.WritableRecord;
public class TopologySort {
    private final static Log LOG = LogFactory.getLog(TopologySort.class);
    public static class TopologySortVertex extends
        Vertex<LongWritable, LongWritable, NullWritable, LongWritable> {
        @Override
        public void compute(
            ComputeContext<LongWritable, LongWritable, NullWritable, LongWritable>
            > context, Iterable<LongWritable> messages) throws IOException {
            // in superstep 0, each vertex sends message whose value is 1 to its
            // neighbors
            if (context.getSuperstep() == 0) { if (hasEdges()) {
                context.sendMessageToNeighbors(this, new LongWritable(1L));
            }
            } else if (context.getSuperstep() >= 1) {
                // compute each vertex's indegree
                long indegree = getValue().get();
                for (LongWritable msg : messages) {
                    indegree += msg.get();
                }
                setValue(new LongWritable(indegree));
                if (indegree == 0) {
                    voteToHalt();
                    if (hasEdges()) {
                        context.sendMessageToNeighbors(this, new LongWritable(-1L));
                    }
                }
                context.write(new LongWritable(context.getSuperstep()), getId());
                LOG.info("vertex: " + getId());
            }
            context.aggregate(new LongWritable(indegree));
        }
    }
}

public static class TopologySortVertexReader extends
    GraphLoader<LongWritable, LongWritable, NullWritable, LongWritable> {
    @Override public void load(
        LongWritable recordNum, WritableRecord record,
        MutationContext<LongWritable, LongWritable, NullWritable, LongWritable>
        > context) throws IOException {
        TopologySortVertex vertex = new TopologySortVertex();
        vertex.setId((LongWritable) record.get(0));
        vertex.setValue(new LongWritable(0));
        String[] edges = record.get(1).toString().split(",");
        for (int i = 0; i < edges.length; i++) {
            long edge = Long.parseLong(edges[i]);
            if (edge >= 0) {
                vertex.addEdge(new LongWritable(Long.parseLong(edges[i])), NullWritable.get());
            }
        }
        LOG.info(record.toString());
        context.addVertexRequest(vertex);
    }
}

public static class LongSumCombiner extends
    Combiner<LongWritable, LongWritable> {
    @Override
    public void combine(LongWritable vertexId, LongWritable combinedMessage,
        LongWritable messageToCombine) throws IOException {
        combinedMessage.set(combinedMessage.get() + messageToCombine.get());
    }
}

public static class TopologySortAggregator extends
```

```
Aggregator<BooleanWritable> {
@SuppressWarnings("rawtypes")
@Override
public BooleanWritable createInitialValue(WorkerContext context)
throws IOException {
return new BooleanWritable(true);
}
@Override
public void aggregate(BooleanWritable value, Object item) throws
IOException {
boolean hasCycle = value.get();
boolean inDegreeNotZero = ((LongWritable) item).get() == 0 ? false :
true;
value.set(hasCycle && inDegreeNotZero);
}
@Override
public void merge(BooleanWritable value, BooleanWritable partial)
throws IOException {
value.set(value.get() && partial.get());
}
@SuppressWarnings("rawtypes")
@Override
public boolean terminate(WorkerContext context, BooleanWritable value
) throws IOException {
if (context.getSuperstep() == 0) {
// since the initial aggregator value is true, and in superstep we don
't
// do aggregate
return false;
}
return value.get();
}
}
public static void main(String[] args) throws IOException { if (args.
length != 2) {
System.out.println("Usage : <inputTable> <outputTable>");
System.exit(-1);
}
// Input format
// 0 1, 2
// 1 3
// 2 3
// 3 -1
// The first column is vertexid, and the second column is the
destination vertexid of the vertex. If the value is -1, the vertex
does not have any outgoing edges.
// Output format
// 0 0
// 1 1
// 1 2
// 2 3
// The first column is the supstep value, in which the topological
sequence is hidden. The second column is vertexid.
// TopologySortAggregator is used to determine if the graph has any
loops.
// If the input graph has a loop, the iteration ends when the indegree
of all active vertices is not 0.
// You can use records in the input and output tables to determine if
the graph has loops.
GraphJob job = new GraphJob();
job.setGraphLoaderClass(TopologySortVertexReader.class);
job.setVertexClass(TopologySortVertex.class);
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());
```

```

job.setCombinerClass(LongSumCombiner.class);
job.setAggregatorClass(TopologySortAggregator.class);
long startTime = System.currentTimeMillis(); job.run();
System.out.println("Job Finished in " + (System.currentTimeMillis() -
startTime) / 1000.0 + " seconds");
}
}
}

```

1.8.6.8 Linear regression

In statistics, linear regression is a statistical analysis method used to determine the dependency between two or more variables. Compared with the classification algorithm that predicts discrete data, the regression algorithm can predict continuous value-type data. The linear regression algorithm defines the loss function as the sum of the least square errors of a sample set. It solves the weight vector by minimizing the loss function.

A common solution is the gradient descent method. It is implemented as follows:

- Initialize the weight vector to provide the descent speed and iterations (or iteration convergence condition).
- Calculate the least square error for each sample.
- Calculate the sum of the least square error, and update the weight based on the descent speed.
- Repeat iterations until convergence occurs.

Example:

```

import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.Aggregator;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.io.DoubleWritable;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.io.Tuple;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.io.WritableRecord;
/**
 * LineRegression input: y,x1,x2,x3,.....
 *
 * @author shiwan.ch
 * @update jiasheng.tjs running parameters are like: tjs_lr_in >
tjs_lr_out 1500 2
 * 0.07
 */

```

```
public class LinearRegression {
    public static class GradientWritable implements Writable {
        Tuple lastTheta;
        Tuple currentTheta;
        Tuple tmpGradient;
        LongWritable count;
        DoubleWritable lost;
        @Override
        public void readFields(DataInput in) throws IOException {
            lastTheta = new Tuple();
            lastTheta.readFields(in);
            currentTheta = new Tuple();
            currentTheta.readFields(in);
            tmpGradient = new Tuple();
            tmpGradient.readFields(in);
            count = new LongWritable();
            count.readFields(in);
            /* update 1: add a variable to store lost at every iteration */
            lost = new DoubleWritable();
            lost.readFields(in);
        }
        @Override
        public void write(DataOutput out) throws IOException {
            lastTheta.write(out);
            currentTheta.write(out);
            tmpGradient.write(out);
            count.write(out);
            lost.write(out);
        }
    }
    public static class LinearRegressionVertex extends
        Vertex<LongWritable, Tuple, NullWritable, NullWritable> {
        @Override
        public void compute(
            ComputeContext<LongWritable, Tuple, NullWritable, NullWritable>
            context, Iterable<NullWritable> messages) throws IOException {
            context.aggregate(getValue());
        }
    }
    public static class LinearRegressionVertexReader extends
        GraphLoader<LongWritable, Tuple, NullWritable, NullWritable> {
        @Override
        public void load(LongWritable recordNum, WritableRecord record,
            MutationContext<LongWritable, Tuple, NullWritable, NullWritable>
            context)
            throws IOException {
            LinearRegressionVertex vertex = new LinearRegressionVertex();
            vertex.setId(recordNum);
            vertex.setValue(new Tuple(record.getAll())); context.addVertexRequest(
            vertex);
        }
    }
    public static class LinearRegressionAggregator extends
        Aggregator<GradientWritable> {
        @SuppressWarnings("rawtypes")
        @Override
        public GradientWritable createInitialValue(WorkerContext context)
            throws IOException {
            if (context.getSuperstep() == 0) {
                /* set initial value, all 0 */
                GradientWritable grad = new GradientWritable();
                grad.lastTheta = new Tuple();
                grad.currentTheta = new Tuple();
                grad.tmpGradient = new Tuple();
            }
        }
    }
}
```



```
grad.count = new LongWritable(1);
grad.lost = new DoubleWritable(0.0);
int n = (int) Long.parseLong(context.getConfiguration().get("Dimension
"));
for (int i = 0; i < n; i++) { grad.lastTheta.append(new DoubleWritable
(0));
grad.currentTheta.append(new DoubleWritable(0));
grad.tmpGradient.append(new DoubleWritable(0));
}
return grad;
} else
return (GradientWritable) context.getLastAggregatedValue(0);
}
public static double vecMul(Tuple value, Tuple theta) {
/* perform this partial computing:  $y(i) - h\theta(x(i))$  for each sample */
/* value denote a piece of sample and value(0) is y */
double sum = 0.0;
for (int j = 1; j < value.size(); j++)
sum += Double.parseDouble(value.get(j).toString()) * Double.parseDoubl
e(theta.get(j).toString());
Double tmp = Double.parseDouble(theta.get(0).toString()) + sum -Double
.parseDouble(value.get(0).toString());
return tmp;
}
@Override
public void aggregate(GradientWritable gradient, Object value) throws
IOException {
/*
* perform on each vertex--each sample i:set theta(j) for each sample
> i
* for each dimension
*/
double tmpVar = vecMul((Tuple) value, gradient.currentTheta);
/*
* update 2:local worker aggregate(), perform like merge() below. This
* means the variable gradient denotes the previous aggregated value
*/
gradient.tmpGradient.set(0, new DoubleWritable( ((DoubleWritable)
gradient.tmpGradient.get(0)).get() + tmpVar));
gradient.lost.set(Math.pow(tmpVar, 2));
/*
* calculate  $(y(i) - h\theta(x(i)))x(i)(j)$  for each sample i for each
* dimension j
*/
for (int j = 1; j < gradient.tmpGradient.size(); j++) gradient.
tmpGradient.set(j, new DoubleWritable(
((DoubleWritable) gradient.tmpGradient.get(j)).get() + tmpVar * Double
.parseDouble(((Tuple) value).get(j).toString())));
}
@Override
public void merge(GradientWritable gradient, GradientWritable partial
) throws IOException {
/* perform SumAll on each dimension for all samples. */
Tuple master = (Tuple) gradient.tmpGradient;
Tuple part = (Tuple) partial.tmpGradient;
for (int j = 0; j < gradient.tmpGradient.size(); j++) {
DoubleWritable s = (DoubleWritable) master.get(j);
s.set(s.get() + ((DoubleWritable) part.get(j)).get());
}
gradient.lost.set(gradient.lost.get() + partial.lost.get());
}
@SuppressWarnings("rawtypes")
@Override
```

```
public boolean terminate(WorkerContext context, GradientWritable
gradient) throws IOException {
    /*
    * 1. calculate new theta 2. judge the diff between last step and this
    * step, if smaller than the threshold, stop iteration
    */
    gradient.last = new DoubleWritable(gradient.last.get() / (2 * context.
    getTotalNumVertices()));
    /*
    * we can calculate lost in order to make sure the algorithm is running
    > on
    * the right direction (for debug)
    */
    System.out.println(gradient.count + " lost:" + gradient.last);
    Tuple tmpGradient = gradient.tmpGradient;
    System.out.println("tmpGra" + tmpGradient);
    Tuple lastTheta = gradient.lastTheta;
    Tuple tmpCurrentTheta = new Tuple(gradient.currentTheta.size());
    System.out.println(gradient.count + " terminate_start_last:" +
    lastTheta);
    double alpha = 0.07; // learning rate
    // alpha =
    // Double.parseDouble(context.getConfiguration().get("Alpha"));
    /* perform theta(j) = theta(j)-alpha*tmpGradient */
    long M = context.getTotalNumVertices();
    /*
    * update 3: add (/M) on the code. The original code forget this step
    */
    for (int j = 0; j < lastTheta.size(); j++) { tmpCurrentTheta
    .set( j,
    new DoubleWritable(Double.parseDouble(lastTheta.get(j)
    .toString()) - alpha / M * Double.parseDouble(tmpGradient.get(j).
    toString())));
    }
    System.out.println(gradient.count + " terminate_start_current:" +
    tmpCurrentTheta);
    // judge if convergence is happening.
    double diff = 0.00d;
    for (int j = 0; j < gradient.currentTheta.size(); j++)
    diff += Math.pow(((DoubleWritable) tmpCurrentTheta.get(j)).get() - ((
    DoubleWritable) lastTheta.get(j)).get(), 2);
    if (
    /*
    * Math.sqrt(diff) < 0.000000000005d ||
    */
    Long.parseLong(context.getConfiguration().get("Max_Iter_Num")) ==
    gradient.count
    .get()) { context.write(gradient.currentTheta.toArray());
    return true;
    }
    gradient.lastTheta = tmpCurrentTheta;
    gradient.currentTheta = tmpCurrentTheta;
    gradient.count.set(gradient.count.get() + 1);
    int n = (int) Long.parseLong(context.getConfiguration().get("Dimension
    "));
    /*
    * update 4: Important!!! Remember this step. Graph won't reset the
    * initial value for global variables at the beginning of each
    iteration
    */
    for (int i = 0; i < n; i++) {
    gradient.tmpGradient.set(i, new DoubleWritable(0));
    }
    return false;
}
```

```

}
}
public static void main(String[] args) throws IOException { GraphJob
job = new GraphJob();
job.setGraphLoaderClass(LinearRegressionVertexReader.class); job.
setRuntimePartitioning(false);
job.setNumWorkers(3);
job.setVertexClass(LinearRegressionVertex.class);
job.setAggregatorClass(LinearRegressionAggregator.class);
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());
job.setMaxIteration(Integer.parseInt(args[2])); // Numbers of
Iteration
job.setInt("Max_Iter_Num", Integer.parseInt(args[2]));
job.setInt("Dimension", Integer.parseInt(args[3])); // Dimension
job.setFloat("Alpha", Float.parseFloat(args[4])); // Learning rate
long start = System.currentTimeMillis(); job.run();
System.out.println("Job Finished in " + (System.currentTimeMillis() -
start) / 1000.0 + " seconds");
}
}

```

1.8.6.9 Count triangles

This algorithm is used to calculate the number of triangles passing through each vertex. The algorithm is implemented as follows:

- **Each vertex sends its ID to all outgoing neighbors.**
- **Each vertex stores information about incoming and outgoing neighbors, and sends this information to outgoing neighbors.**
- **Each vertex calculates the number of endpoint intersections for each edge, calculates the sum, and outputs the results to a table.**
- **The number of triangles is the sum of the output results in the table divided by 3.**

Example:

```

import java.io.IOException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.Edge;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.WorkerContext;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.NullWritable;
import com.aliyun.odps.io.Tuple;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.io.WritableRecord;
/**
 * Compute the number of triangles passing through each vertex.
 *
 * The algorithm can be computed in three supersteps:
 * I. Each vertex sends a message with its ID to all its outgoing
 * neighbors.

```

```
* II. The incoming neighbors and outgoing neighbors are stored and
* send to outgoing neighbors.
* III. For each edge compute the intersection of the sets at
destination
* vertex and sum them, then output to table.
*
* The triangle count is the sum of output table and divide by three >
since
* each triangle is counted three times.
*
**/
public class TriangleCount {
    public static class TCVertex extends
    Vertex<LongWritable, Tuple, NullWritable, Tuple> {
    @Override
    public void setup(
    WorkerContext<LongWritable, Tuple, NullWritable, Tuple> context)
    throws IOException {
    // collect the outgoing neighbors
    Tuple t = new Tuple();
    if (this.hasEdges()) {
    for (Edge<LongWritable, NullWritable> edge : this.getEdges()) {
    t.append(edge.getDestVertexId());
    }
    }
    this.setValue(t);
    }
    @Override
    public void compute(
    ComputeContext<LongWritable, Tuple, NullWritable, Tuple> context,
    Iterable<Tuple> msgs) throws IOException {
    if (context.getSuperstep() == 0L) {
    // sends a message with its ID to all its outgoing neighbors
    Tuple t = new Tuple(); t.append(getId());
    context.sendMessageToNeighbors(this, t);
    } else if (context.getSuperstep() == 1L) {
    // store the incoming neighbors
    for (Tuple msg : msgs) {
    for (Writable item : msg.getAll()) {
    if (! this.getValue().getAll().contains((LongWritable)item)) {
    this.getValue().append((LongWritable)item);
    }
    }
    }
    // send both incoming and outgoing neighbors to all outgoing neighbors
    context.sendMessageToNeighbors(this, getValue());
    } else if (context.getSuperstep() == 2L) {
    // count the sum of intersection at each edge
    long count = 0;
    for (Tuple msg : msgs) {
    for (Writable id : msg.getAll()) {
    if (getValue().getAll().contains(id)) { count ++;
    }
    }
    }
    // output to table
    context.write(getId(), new LongWritable(count));
    this.voteToHalt();
    }
    }
    public static class TCVertexReader extends
    GraphLoader<LongWritable, Tuple, NullWritable, Tuple> {
    @Override public void load(
```

```

LongWritable recordNum, WritableRecord record,
MutationContext<LongWritable, Tuple, NullWritable, Tuple> context)
throws IOException {
    TCVertex vertex = new TCVertex();
    vertex.setId((LongWritable) record.get(0));
    String[] edges = record.get(1).toString().split(",");
    for (int i = 0; i < edges.length; i++) { try {
        long destID = Long.parseLong(edges[i]);
        vertex.addEdge(new LongWritable(destID), NullWritable.get());
    } catch (NumberFormatException nfe) { System.err.println("Ignore " +
nfe);
    }
    }
    context.addVertexRequest(vertex);
}
}
}
public static void main(String[] args) throws IOException { if (args.
length < 2) {
    System.out.println("Usage: <input> <output>"); System.exit(-1);
}
    GraphJob job = new GraphJob();
    job.setGraphLoaderClass(TCVertexReader.class);
    job.setVertexClass(TCVertex.class);
    job.addInput(TableInfo.builder().tableName(args[0]).build());
    job.addOutput(TableInfo.builder().tableName(args[1]).build());
    long startTime = System.currentTimeMillis();
    job.run();
    System.out.println("Job Finished in " + (System.currentTimeMillis() -
startTime) / 1000.0 + " seconds");
}
}
}

```

1.8.6.10 GraphLoader

The following example describes how to compile a graph job program to load data of different types. It mainly covers how GraphLoader and VertexResolver are used together to build the graph.

Example:

```

import java.io.IOException;
import com.aliyun.odps.conf.Configuration;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.graph.ComputeContext;
import com.aliyun.odps.graph.GraphJob;
import com.aliyun.odps.graph.GraphLoader;
import com.aliyun.odps.graph.Vertex;
import com.aliyun.odps.graph.VertexResolver;
import com.aliyun.odps.graph.MutationContext;
import com.aliyun.odps.graph.VertexChanges;
import com.aliyun.odps.graph.Edge;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.WritableComparable;
import com.aliyun.odps.io.WritableRecord;
/**
 * A MaxCompute Graph job uses MaxCompute tables as the input. Assume
 * that a job has two input tables, one storing vertices and the other
 * storing edges.
 * The format of the table storing vertices is as follows:
 * +-----+

```

```

* VertexID | VertexValue |
* +-----+
* | id0| 9|
* +-----+
* | id1| 7|
* +-----+
* | id2| 8|
* +-----+
*
* The format of the table storing edges is as follows:
* +-----+
* VertexID | DestVertexID| EdgeValue|
* +-----+
* | id0| id1| 1|
* +-----+
* | id0| id2| 2|
* +-----+
* | id2| id1| 3|
* +-----+
*
* The two preceding tables show that id0 has two outgoing edges
pointing to id1 and id2. id2 has an outgoing edge pointing to id1, and
id1 has no outgoing edges.
*
* For data of this type, in GraphLoader::load(LongWritable, Record
, MutationContext), > MutationContext#addVertexRequest(Vertex) can
be used to add vertices to the graph, while link MutationContext#
addEdgeRequest(WritableComparable,Edge) can be used to add edges to
the graph. In link VertexResolver#resolve(WritableComparable, Vertex,
VertexChanges, boolean), vertices and edges added in the load() method
are combined to a vertex object, which is used as the returned value
and added to the graph for participating in computation.
*
**/
public class VertexInputFormat {
private final static String EDGE_TABLE = "edge.table";
/**
* Resolve a record to vertices and edges. Each record indicates a
vertex or an edge based on its source.
*
* Enter a record to generate key-value pairs as you process com.aliyun
.odps.mapreduce.Mapper#map. The keys are vertex IDs, and the values
are vertices or edges written based on the context. These key-value
pairs are summarized based on vertex IDs using LoadingVertexResolver.
*
* Note: Vertices or edges added here are requests sent based on the
record content, and are not used in computation. Only vertices or
edges added using VertexResolver participate in computation.
**/
public static class VertexInputLoader extends
GraphLoader<LongWritable, LongWritable, LongWritable, LongWritable> {
private boolean isEdgeData;
/**
* Configure VertexInputLoader.
*
* @param conf
* Indicate the configured parameters of a job. These parameters are
configured in the MAIN function of GraphJob, or set on the console.
* @param workerId
* Indicate the serial number of the operating worker. It starts from 0
and can be used to build a unique vertex ID.
* @param inputTableInfo

```

```
* Indicate information about the input table loaded to the current
worker. The information can be used to determine the type of current
input data (record format).
**/
@Override
public void setup(Configuration conf, int workerId, TableInfo
inputTableInfo) {
    isEdgeData = conf.get(EDGE_TABLE).equals(inputTableInfo.getTableInfoName
());
}
/**
 * Based on the record content, resolve corresponding edges and send a
request to add them to the graph.
 *
 * @param recordNum
 * Indicate the record serial number, which starts from 1 and is
separately counted in each worker.
 * @param record
 * Indicate the record in the input table. It contains three columns,
indicating the first vertex, last vertex, and edge weight.
 * @param context
 * Indicate the context for adding resolved edges to the graph.
 **/
@Override public void load(
LongWritable recordNum, WritableRecord record,
MutationContext<LongWritable, LongWritable, LongWritable, LongWritable
> context) throws IOException {
    if (isEdgeData) {
        /**
         * Data comes from the table that stores edges.
         *
         * 1. The first column indicates the first vertex ID.
         */
        LongWritable sourceVertexID = (LongWritable) record.get(0);
        /**
         * 2. The second column indicates the last vertex ID.
         */
        LongWritable destinationVertexID = (LongWritable) record.get(1);
        /**
         * 3. The third column indicates the edge weight.
         */
        LongWritable edgeValue = (LongWritable) record.get(2);
        /**
         * 4. Create an edge that consists of the last vertex ID and edge
weight.
         */
        Edge<LongWritable, LongWritable> edge = new Edge<LongWritable,
LongWritable>( destinationVertexID, edgeValue);
        /**
         * 5. Send a request to add an edge to the first vertex.
         */
        context.addEdgeRequest(sourceVertexID, edge);
        /**
         * 6. If each record indicates a bidirectional edge, repeat steps 4 and
5. Edge<LongWritable, > LongWritable> edge2 = new
         * Edge<LongWritable, LongWritable>( sourceVertexID, edgeValue);
         * context.addEdgeRequest(destinationVertexID, edge2);
         */
    } else {
        /**
         * Data comes from the table that stores vertices.
         *
         * 1. The first column indicates the vertex ID.
         */
    }
```

```

LongWritable vertexID = (LongWritable) record.get(0);
/**
 * 2. The second column indicates the vertex value.
 */
LongWritable vertexValue = (LongWritable) record.get(1);
/**
 * 3. Create a vertex that consists of the vertex ID and vertex value.
 */
MyVertex vertex = new MyVertex();
/**
 * 4. Initialize the vertex.
 */
vertex.setId(vertexID); vertex.setValue(vertexValue);
/**
 * 5. Send a request for adding a vertex.
 */
context.addVertexRequest(vertex);
}
}
}
/**
 * Summarize key-value pairs generated using GraphLoader::load(
 * LongWritable, Record, > MutationContext), which is similar to
 * Reduce in com.aliyun.odps.mapreduce.Reducer. For the unique vertex
 * > ID, all actions such as
 * adding or deleting vertices or edges for the ID are stored in
 * VertexChanges.
 *
 * Note: Not only conflicting vertices or edges added by using the load
 * () method are called. (A conflict occurs when multiple same vertex
 * objects or duplicate edges are added.)
 * All IDs requested to be generated using the load() method are called
 * .
 */
public static class LoadingResolver extends
VertexResolver<LongWritable, LongWritable, LongWritable, LongWritable>
{
/**
 * Process a request for adding/deleting vertices or edges for an ID.
 *
 * VertexChanges has four APIs, which correspond to the four APIs of
 * MutationContext:
 * VertexChanges::getAddedVertexList() corresponds to
 * MutationContext::addVertexRequest(Vertex).
 * In the load() method, if vertex objects with the same ID are
 * requested to be added, such vertex objects are collected to the
 * returned list.
 * VertexChanges::getAddedEdgeList() corresponds to
 * MutationContext::addEdgeRequest(WritableComparable, Edge)
 * If edge objects with the same first vertex ID are requested to be
 * added, such edge objects are collected to the returned list.
 * VertexChanges::getRemovedVertexCount() corresponds to
 * MutationContext::removeVertexRequest(WritableComparable)
 * If vertices with the same ID are requested to be deleted, the number
 * of total deletion requests is returned.
 * VertexChanges#getRemovedEdgeList() corresponds to
 * MutationContext#removeEdgeRequest(WritableComparable, WritableCo
 * mparable)
 * If edge objects with the same first vertex ID are requested to be
 * deleted, such edge objects are collected to the returned list.
 *
 * By processing ID changes, you can state whether the ID participates
 * in computation using the returned value. If the returned vertex is not
 * NULL,

```



```

* the ID participates in subsequent computation. If the returned
vertex is NULL, the ID does not participate in subsequent computation.
*
* @param vertexId
* Indicate the ID of the vertex to be added, or the ID of the first
vertex of the edge to be added.
* @param vertex
* Indicate an existing vertex object. Its value is always NULL in the
data loading phase.
* @param vertexChanges
* Indicate the set of vertices or edges to be added/deleted for the ID
.
* @param hasMessages
* Indicate whether the ID has any input messages. Its value is always
false in the data loading phase.
**/
@Override
public Vertex<LongWritable, LongWritable, LongWritable, LongWritable>
resolve( LongWritable vertexId,
Vertex<LongWritable, LongWritable, LongWritable, LongWritable> vertex
, VertexChanges<LongWritable, LongWritable, LongWritable, LongWritable>
vertexChanges, boolean hasMessages) throws IOException {
/**
* 1. Obtain the vertex object for computation.
**/
MyVertex computeVertex = null;
if (vertexChanges.getAddedVertexList() == null
|| vertexChanges.getAddedVertexList().isEmpty()) { computeVertex = new
MyVertex(); computeVertex.setId(vertexId);
} else {
/**
* Each record indicates a unique vertex in the table that stores
vertices.
**/
computeVertex = (MyVertex) vertexChanges.getAddedVertexList().get(0);
}
/**
* 2. Add the edge, which is requested to be added to the vertex,
to the vertex object. If the data is a possible duplicate, perform
deduplication based on the algorithm needs.
**/
if (vertexChanges.getAddedEdgeList() != null) {
for (Edge<LongWritable, LongWritable> edge : vertexChanges.getAddedEd
geList()) { computeVertex.addEdge(edge.getDestVertexId(), edge.
getValue());
}
}
/**
* 3. Return the vertex object and add it to the final graph for
computation.
**/
return computeVertex;
}
}
/**
* Determine actions of the vertex that participates in computation.
*
**/
public static class MyVertex extends
Vertex<LongWritable, LongWritable, LongWritable, LongWritable> {
/**
* Write the vertex edge to the result table based on the format of the
input table. Ensure that the format and data of the input and output
tables are the same.

```

```
*
* @param context
* Indicate the runtime context.
* @param messages
* Indicate the input message.
**/
@Override
public void compute(
    ComputeContext<LongWritable, LongWritable, LongWritable, LongWritable> context,
    Iterable<LongWritable> messages) throws IOException {
    /**
    * Write the vertex ID and value to the result table that stores
    vertices.
    **/
    context.write("vertex", getId(), getValue());
    /**
    * Write the vertex edge to the result table that stores edges.
    **/
    if (hasEdges()) {
        for (Edge<LongWritable, LongWritable> edge : getEdges()) { context.
            write("edge", getId(), edge.getDestVertexId(),
            edge.getValue());
        }
    }
    /**
    * Perform one round of iteration.
    **/
    voteToHalt();
}
}
/**
* @param args
* @throws IOException
*/
public static void main(String[] args) throws IOException {
    if (args.length < 4) { throw new IOException(
        "Usage: VertexInputFormat <vertex input> <edge input> <vertex output>
        <edge output>");
    }
    /**
    * GraphJob is used to configure Graph jobs.
    */
    GraphJob job = new GraphJob();
    /**
    * 1. Specify input graph data and the table that stores edges.
    */
    job.addInput(TableInfo.builder().tableName(args[0]).build());
    job.addInput(TableInfo.builder().tableName(args[1]).build());
    job.set(EDGE_TABLE, args[1]);
    /**
    * 2. Specify the data loading mode, and resolve the records to edges.
    Similar to Map, the generated key is the vertex ID, and the value is
    the edge.
    */
    job.setGraphLoaderClass(VertexInputLoader.class);
    /**
    * 3. Specify the data loading phase, and generate the vertex that
    participates in computation. Similar to Reduce, edges generated by Map
    are combined to a vertex.
    */
    job.setLoadingVertexResolverClass>LoadingResolver.class);
    /**
    * 4. Specify actions of the vertex that participates in computation.
    The vertex.compute() method is used for each round of iteration.
    */
}
```

```
*/  
job.setVertexClass(MyVertex.class);  
/**  
 * 5. Specify the output table of the Graph job, and write the  
 * computation result to the result table.  
 */  
job.addOutput(TableInfo.builder().tableName(args[2]).label("vertex").  
build());  
job.addOutput(TableInfo.builder().tableName(args[3]).label("edge").  
build());  
/**  
 * 6. Submit the job for execution.  
 */  
job.run();  
}  
}
```

1.9 Java SDK

MaxCompute provides Java SDK with a variety of APIs to support development on MaxCompute.

For more information about the APIs, see *MaxCompute Developer Guide*.

1.10 Java sandbox limits

MaxCompute MapReduce and UDF programs in distributed environments are run in Java sandboxes. The main programs of MapReduce are not subject to these limits.

The limits include:

- **Local files cannot be accessed directly, and can only be accessed through APIs provided by MaxCompute MapReduce or Graph. You can access the resources specified by the resources option, such as files, JAR packages, and resource tables. You can use System.out and System.err to export logs and run the log command on the MaxCompute console to view log information.**
- **Direct access to the distributed file system is not allowed. You can only use MaxCompute MapReduce to Graph to access records of tables.**
- **JNI calls are not allowed.**
- **Java threads cannot be created, and Linux commands cannot be executed by sub-threads.**
- **Network access, including the operation of getting the local IP address, is prohibited.**

- **Java reflection restriction:** The `suppressAccessChecks` permission is prohibited, so you cannot set a private attribute or method accessible to read private attributes or call private methods.

Specifically, an `access denied` error is returned when you perform any of the preceding operations.

Methods for accessing local files:

java.io.File:

```
public boolean delete()
public void deleteOnExit()
public boolean exists()
public boolean canRead()
public boolean isFile()
public boolean isDirectory()
public boolean isHidden()
public long lastModified()
public long length()
public String[] list()
public String[] list(FilenameFilter filter)
public File[] listFiles()
public File[] listFiles(FilenameFilter filter)
public File[] listFiles(FileFilter filter)
public boolean canWrite()
public boolean createNewFile()
public static File createTempFile(String prefix, String suffix)
public static File createTempFile(String prefix, String suffix, File
directory)
public boolean mkdir()
public boolean mkdirs()
public boolean renameTo(File dest)
public boolean setLastModified(long time)
public boolean setReadOnly()
java.io.RandomAccessFile:
RandomAccessFile(String name, String mode)
RandomAccessFile(File file, String mode)
java.io.FileInputStream:
FileInputStream(FileDescriptor fdObj)
FileInputStream(String name) FileInputStream(File file)
java.io.FileOutputStream:
FileOutputStream(FileDescriptor fdObj)
FileOutputStream(File file)
FileOutputStream(String name)
FileOutputStream(String name, boolean append)
java.lang.Class:
public ProtectionDomain getProtectionDomain()
java.lang.ClassLoader:
ClassLoader ()
ClassLoader(ClassLoader parent)
java. lang. Runtime:
public Process exec(String command)
public Process exec(String command, String envp[])
public Process exec(String cmdarray[])
public Process exec(String cmdarray[], String envp[])
public void exit(int status)
public static void runFinalizersOnExit(boolean value)
public void addShutdownHook(Thread hook)
```

```
public boolean removeShutdownHook(Thread hook)
public void load(String lib)
public void loadLibrary(String lib)
java.lang.System:
public static void exit(int status)
public static void runFinalizersOnExit(boolean value)
public static void load(String filename)
public static void loadLibrary( String libname)
public static Properties getProperties()
public static void setProperties(Properties props)
public static String getProperty(String key)
// Only some keys are accessible.
public static String getProperty(String key, String def)
// Only some keys are accessible.
public static String setProperty(String key, String value)
public static void setIn(InputStream in)
public static void setOut(PrintStream out)
public static void setErr(PrintStream err)
public static synchronized void setSecurityManager(SecurityManager s
)
```

List of keys allowed by System.getProperty:

```
java.version
java.vendor
java.vendor.url
java.class.version
os.name os.version
os.arch
file.separator
path.separator
line.separator
java.specification.version
java.specification.vendor
java.specification.name
java.vm.specification.version
java.vm.specification.vendor
java.vm.specification.name
java.vm.version
java.vm.vendor
java.vm.name
file.encoding
user.timezone
java. lang. Thread:
Thread ()
Thread(Runnable target)
Thread(String name)
Thread(Runnable target, String name)
Thread(ThreadGroup group, ...)
public final void checkAccess()
public void interrupt()
public final void suspend()
public final void resume()
public final void setPriority (int newPriority)
public final void setName(String name)
public final void setDaemon(boolean on)
public final void stop()
public final synchronized void stop(Throwable obj)
public static int enumerate(Thread tarray[])
public void setContextClassLoader(ClassLoader cl)
java. lang. ThreadGroup:
ThreadGroup (String name)
ThreadGroup(ThreadGroup parent, String name)
```

```
public final void checkAccess()
public int enumerate(Thread list[])
public int enumerate(Thread list[], boolean recurse)
public int enumerate(ThreadGroup list[])
public int enumerate(ThreadGroup list[], boolean recurse)
public final ThreadGroup getParent()
public final void setDaemon(boolean daemon)
public final void setMaxPriority(int pri)
public final void suspend()
public final void resume()
public final void destroy()
public final void interrupt()
public final void stop()
java.lang.reflect.AccessibleObject:
public static void setAccessible (...)
public void setAccessible (...)
java.net.InetAddress:
public String getHostName ()
public static InetAddress[] getAllByName(String host)
public static InetAddress getLocalHost()
java.net.DatagramSocket:
public InetAddress getLocalAddress()
java.net.Socket:
Socket(...)
java.net.ServerSocket:
ServerSocket (...)
public Socket accept()
protected final void implAccept(Socket s)
public static synchronized void setSocketFactory(...)
public static synchronized void setSocketImplFactory(...)
java.net.DatagramSocket:
DatagramSocket (...)
public synchronized void receive(DatagramPacket p)
java.net.MulticastSocket:
MulticastSocket(...)
java.net.URL:
URL(...)
public static synchronized void setURLStreamHandlerFactory(...)
java.net.URLConnection
public static synchronized void setContentHandlerFactory(...)
public static void setFileNameMap(FileNameMap map)
java.net.HttpURLConnection:
public static void setFollowRedirects(boolean set)
java.net.URLClassLoader
URLClassLoader(...)
java.security.AccessControlContext:
public AccessControlContext(AccessControlContext acc, DomainCombiner
combiner)
public DomainCombiner getDomainCombiner()
```

1.11 Volume lifecycle management

1.11.1 Overview

In the previous version of MaxCompute, a partition in a volume does not have a lifecycle and can exist forever. You have to manage the lifecycle of a volume. In some cases, user management of volume lifecycle may cause a problem. For example, if the account used to delete a partition is different from the account

used to create the partition, the delete operation fails. The new feature of volume lifecycle management can solve this problem.

For more information about simple volume lifecycle management operations, see *Volume lifecycle operations*.

1.11.2 Volume lifecycle operations

Create a volume with a specified lifecycle

Example:

```
odps@ your_project>fs -mkv test_volume -lifecycle 7 "this is a test volume";  
OK
```

Modify the lifecycle of a volume

Example:

```
odps@ your_project>fs -alter test_volume -lifecycle 3;  
OK
```

View the lifecycle of a volume

Example:

```
odps@ your_project>fs -meta test_volume;  
Comment: "this is a test volume"  
Length: 0  
File number: 0  
Lifecycle: 3  
OK
```

1.12 Spark on MaxCompute

1.12.1 Overview

Spark on MaxCompute is a solution developed by Alibaba Cloud to enable the seamless use of Spark on the MaxCompute platform. It supplements a wide variety of features to MaxCompute.

Spark on MaxCompute provides a native Spark user experience and offers native Spark components and APIs. Spark on MaxCompute can access MaxCompute data sources and enhance security for multi-tenant scenarios. Spark on MaxCompute can also act as a management platform to share resources, storage, and user systems between jobs and ensure high performance at a low cost. Spark can work

with MaxCompute to create better and more efficient data processing solutions.

Community Spark applications can run in Spark on MaxCompute.

Spark on MaxCompute has an independent data development node in DataWorks and supports data development in DataWorks.

1.12.2 Project resources

Before you use Spark on MaxCompute, you may need to pay attention to or download the following project resources:

- **Spark on MaxCompute release package:** Download the latest release package at [Spark on MaxCompute](#).
- **Spark on MaxCompute plugin:** This is an open-source program. Download the plugin at [Aliyun Cupid SDK](#).

After you prepare the preceding project resources, you need to complete environment configuration. Then, you can run related GitHub demos.

1.12.3 Environment settings

1.12.3.1 Decompress the Spark on MaxCompute release package

Download the latest version of the Spark on MaxCompute release package and decompress it. The structure of the decompressed folder is as follows:

```
.
|-- R
|-- RELEASE
|-- __spark_libs__.zip
|-- bin
|-- conf
|-- cupid
|-- derby.log
|-- examples
|-- jars
|-- logs
|-- metastore_db
|-- python
|-- sbin
|-- yarn
```

1.12.3.2 Set environment variables

Set required environment variables.



Note:

The main environment variables are JAVA_HOME and SPARK_HOME.

Set JAVA_HOME

```
export JAVA_HOME=/path/to/jdk
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
```

Set SPARK_HOME

```
export SPARK_HOME=/path/to/spark_extracted_package
export PATH=$SPARK_HOME/bin:$PATH
```

If you use SparkR, install R in the `/home/admin/R` directory. Then, run the following command to set the path:

```
export PATH=/home/admin/R/bin/:$PATH
```

If you use PySpark, install Python 2.7. Then, run the following command to set the path:

```
export PATH=/path/to/python/bin/:$PATH
```

1.12.3.3 Configure Spark-defaults.conf

The `$SPARK_HOME/conf` directory contains a file named `spark-defaults.conf`. Before you submit a Spark task to MaxCompute, you must configure your MaxCompute account in this file.

The following content is the default configuration in the file. You only need to enter your account information in the blanks.

```
# OdpsAccount Info Setting
spark.hadoop.odps.project.name=
spark.hadoop.odps.access.id=
spark.hadoop.odps.access.key=
spark.hadoop.odps.end.point=
#spark.hadoop.odps.moye.trackurl.host=
#spark.hadoop.odps.cupid.webproxy.endpoint=
spark.sql.catalogImplementation=odps

# spark-shell Setting
spark.driver.extraJavaOptions -Dscala.repl.reader=com.aliyun.odps.
spark_repl.OdpsInteractiveReader -Dscala.usejavacp=true

# SparkR Setting
# odps.cupid.spark.r.archive=/path/to/R-PreCompile-Package.zip

# Cupid Longtime Job
# spark.hadoop.odps.cupid.engine.running.type=longtime
# spark.hadoop.odps.cupid.job.capability.duration.hours=8640
# spark.hadoop.odps.moye.trackurl.dutation=8640
```

```
# spark.r.command=/home/admin/R/bin/Rscript
# spark.hadoop.odps.cupid.disk.driver.enable=false
spark.hadoop.odps.cupid.bearer.token.enable=false
spark.hadoop.odps.exec.dynamic.partition.mode=nonstrict
```

1.12.4 Quick start

This topic describes how to use Spark on MaxCompute.

1. Download the Spark package from [Spark on MaxCompute](#) and decompress the package.

2. Set environment variables.

```
export SPARK_HOME=/path/to/spark-2.1.0-private-cloud-v3.1.0
export JAVA_HOME=/path/to/java/
```

3. Set spark-defaults.conf.

```
cp $SPARK_HOME/conf/spark-defaults.conf.template $SPARK_HOME/conf/
spark-defaults.conf
```

Edit \$SPARK_HOME/conf/spark-defaults.conf by filling information in the blanks.

```
# OdpsAccount Info Setting
spark.hadoop.odps.project.name=
spark.hadoop.odps.access.id=
spark.hadoop.odps.access.key=
spark.hadoop.odps.end.point=
#spark.hadoop.odps.moye.trackurl.host=
#spark.hadoop.odps.cupid.webproxy.endpoint=
spark.sql.catalogImplementation=odps

# spark-shell Setting
spark.driver.extraJavaOptions -Dscala.repl.reader=com.aliyun.odps.
spark_repl.OdpsInteractiveReader -Dscala.usejavacp=true

# SparkR Setting
# odps.cupid.spark.r.archive=/path/to/R-PreCompile-Package.zip

# Cupid Longtime Job
# spark.hadoop.odps.cupid.engine.running.type=longtime
# spark.hadoop.odps.cupid.job.capability.duration.hours=8640
# spark.hadoop.odps.moye.trackurl.dutation=8640

# spark.r.command=/home/admin/R/bin/Rscript
# spark.hadoop.odps.cupid.disk.driver.enable=false
spark.hadoop.odps.cupid.bearer.token.enable=false
spark.hadoop.odps.exec.dynamic.partition.mode=nonstrict
```

4. Prepare spark-example.

```
git clone https://github.com/aliyun/aliyun-cupid-sdk.git
cd aliyun-cupid-sdk
```

```
mvn -T 1C clean install -DskipTests
```

5. Run SparkPi.

```
cd $SPARK_HOME
bin/spark-submit --master yarn-cluster --class com.aliyun.odps.spark
.examples.SparkPi /path/to/aliyun-cupid-sdk/examples/spark-examples/
target/spark-examples_2.11-1.0.0-SNAPSHOT-shaded.jar
```

If you see the following output, your operation is successful. Other logs may be included in the output.

```
18/02/09 15:52:28 INFO Client: Application report for applicatio
n_1518162700322_1635034099 (state: FINISHED)
18/02/09 15:52:28 INFO Client:
  client token: N/A
  diagnostics: N/A
  ApplicationMaster host: ***
  ApplicationMaster RPC port: ***
  queue: queue
  start time: 1518162732343
  final status: SUCCEEDED
  tracking URL: http://***:80/proxyview/jobview/?h=http://***:80/api
&p=odps_smoke_test&i=20180209075148695gbkp8e01&t=spark&id=applicatio
n_1518162700322_1635034099&metaname=20180209075148695gbkp8e01&token=
YkhjNXJWZ0dvdzVScXVFQWpCQWMra1RSZHVFPsPRFBTX09CTzoxMzY10TM3MTUwNzcyMj
EzLDE1MTg0MjE5MzUseyJTdGF0ZW1lbnQiOi0lt7IkFjdGlvbiI6WyJvZHBz0lJlYWQiXSwi
RWZmZWNOIjoIQWxs3ciLCJSZXNvdXJjZSI6WyJhY3M6b2RwczoqOnByb2plY3RzL29kcH
Nfc21va2VfdGVzdC9pbN0YW5jZXMvMjAxODAyMDkwNzUxNDg2OTVnYmtwOGUwMSJdfV0s
IlZlcnNpb24iOiIxIn0=
  user: user
18/02/09 15:52:28 INFO ShutdownHookManager: Shutdown hook called
18/02/09 15:52:28 INFO ShutdownHookManager: Deleting directory /tmp/
spark-d77416ad-79a8-49f7-931d-0533663b5d85
```

1.12.5 Common cases

1.12.5.1 WordCount example

Spark runs simple WordCount.



Note:

You must download the GitHub project and compile the project before running the corresponding demos.

```
git clone https://github.com/aliyun/aliyun-cupid-sdk.git
-- Download a GitHub project.
cd aliyun-cupid-sdk
mvn -T 1C clean install -DskipTests
-- Compile the GitHub project.
```

After you complete the preceding steps, JAR packages are created. These JAR packages will be used to run the demos in this and the subsequent topics.

The following is the code for this example:

```
package com.aliyun.odps.spark.examples

import org.apache.spark.sql.SparkSession

object WordCount {
  def main(args: Array[String]) {
    val spark = SparkSession
      .builder()
      .appName("WordCount")
      .getOrCreate()
    val sc = spark.sparkContext
    try {
      sc.parallelize(1 to 100, 10).map(word => (word, 1)).reduceByKey(
        _ + _, 10).take(100).foreach(println)
    } finally {
      sc.stop()
    }
  }
}
```

Run the following command to submit the job:

```
bin/spark-submit \
--master yarn-cluster \
--class com.aliyun.odps.spark.examples.WordCount \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-
SNAPSHOT-shaded.jar
```

1.12.5.2 OSS access example

Spark can access OSS data.

Example:

```
package com.aliyun.odps.spark.examples.oss

import org.apache.spark.sql.SparkSession

object SparkUnstructuredDataCompute {
  def main(args: Array[String]) {
    val spark = SparkSession
      .builder()
      .appName("SparkUnstructuredDataCompute")
      .config("spark.hadoop.fs.oss.accessKeyId", "***")
      .config("spark.hadoop.fs.oss.accessKeySecret", "***")
      .config("spark.hadoop.fs.oss.endpoint", "oss-cn-hangzhou-zmf.
aliyuncs.com")
      .getOrCreate()

    val sc = spark.sparkContext
    try {
      val pathIn = "oss://bucket/inputdata/"
      val inputData = sc.textFile(pathIn, 5)
      val cnt = inputData.count
      println(s"count: $cnt")
    } finally {
      sc.stop()
    }
  }
}
```

```
}
}
}
```

Run the following command to submit the job:

```
./bin/spark-submit
--jars cupid/hadoop-aliyun-package-3.0.0-alpha2-odps-jar-with-
dependencies.jar
--class com.aliyun.odps.spark.examples.oss.SparkUnstructuredDat
aCompute
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar
```

1.12.5.3 MaxCompute table read/write example

Read/write a MaxCompute table and convert it to Spark RDD.



Notice:

The project or table specified in the demo must exist or be changed to the specific project or table.

Example:

```
package com.aliyun.odps.spark.examples

import com.aliyun.odps.data.Record
import com.aliyun.odps.{ PartitionSpec, TableSchema}
import org.apache.spark.odps.OdpsOps
import org.apache.spark.sql.SparkSession
import scala.util.Random

object OdpsTableReadWrite {
  def main(args: Array[String]) {

    val spark = SparkSession
      .builder()
      .appName("OdpsTableReadWrite")
      .getOrCreate()

    val sc = spark.sparkContext
    val projectName = sc.getConf.get("odps.project.name")

    try {
      val odpsOps = new OdpsOps(sc)

      // read from normal table via rdd api
      val rdd_0 = odpsOps.readTable(
        projectName,
        "cupid_wordcount",
        (r: Record, schema: TableSchema) => (r.getString(0), r.
getString(1))
      )

      //read from single partition column table via rdd api
      val rdd_1 = odpsOps.readTable(
        projectName,
        "dfctest_single_parted",
```

```
        Array("pt=20160101"),
        (r: Record, schema: TableSchema) => (r.getString(0), r.
getString(1), r.getString("pt"))
    )

    // read from multi partition column table via rdd api
    val rdd_2 = odpsOps.readTable(
        projectName,
        "dfctest_parted",
        Array("pt=20160101,hour=12"),
        (r: Record, schema: TableSchema) => (r.getString(0), r.
getString(1), r.getString("pt"), r.getString(3))
    )

    // read with multi partitionSpec definition via rdd api
    val rdd_3 = odpsOps.readTable(
        projectName,
        "cupid_partition_table1",
        Array("pt1=part1,pt2=part1", "pt1=part1,pt2=part2", "pt1=part2
,pt2=part3"),
        (r: Record, schema: TableSchema) => (r.getString(0), r.
getString(1), r.getString("pt1"), r.getString("pt2"))
    )

    // save rdd into normal table
    val transfer_0 = (v: Tuple2[String, String], record: Record,
schema: TableSchema) => {
        record.set("id", v._1)
        record.set(1, v._2)
    }
    odpsOps.saveToTable(projectName, "cupid_wordcount_empty", rdd_0
, transfer_0, true)

    // save rdd into partition table with single partition spec
    val transfer_1 = (v: Tuple2[String, String], record: Record,
schema: TableSchema) => {
        record.set("id", v._1)
        record.set("value", v._2)
    }
    odpsOps.saveToTable(projectName, "cupid_partition_table1", "pt1=
test,pt2=dev", rdd_0, transfer_1, true)

    // dynamic save rdd into partition table with multiple partition
spec
    val transfer_2 = (v: Tuple2[String, String], record: Record,
part: PartitionSpec, schema: TableSchema) => {
        record.set("id", v._1)
        record.set("value", v._2)

        val pt1_value = if (new Random().nextInt(10) % 2 == 0) "even"
else "odd"
        val pt2_value = if (new Random().nextInt(10) % 2 == 0) "even"
else "odd"
        part.set("pt1", pt1_value)
        part.set("pt2", pt2_value)
    }
    odpsOps.saveToTableForMultiPartition(projectName, "cupid_part
ition_table1", rdd_0, transfer_2, true)
    } catch {
        case ex: Exception => {
            throw ex
        }
    } finally {
        sc.stop
    }
```

```
}
}
}
```

Run the following command to submit the job:

```
bin/spark-submit \
--master yarn-cluster \
--class com.aliyun.odps.spark.examples.OdpsTableReadWrite \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar
```

You can use either of the following methods to adjust the MaxCompute table read concurrency:

- **Modify the `spark.hadoop.odps.input.split.size` parameter. A larger value results in a smaller number of Map tasks. The default value is 256 MB.**
- **Set `numPartition` in `OdpsOps.readTable`. The value determines the number of Map tasks. The number is calculated based on `spark.hadoop.odps.input.split.size`.**

1.12.5.4 MaxCompute Table Spark-SQL example

Use `sqlContext` to read/write a MaxCompute table.



Notice:

- **The project or table specified in the demo must exist or be changed to the specific project or table.**
- `Spark-defaults.conf` **must contain the setting**
`spark.sql.catalogImplementation = odps.`

Example:

```
package com.aliyun.odps.spark.examples
import org.apache.spark.sql.SparkSession
object OdpsTableReadWriteViaSQL {
  def main(args: Array[String]) {
    // please make sure spark.sql.catalogImplementation=odps in spark-
    defaults.conf
    // to enable odps catalog
    val spark = SparkSession
      .builder()
      .appName("OdpsTableReadWriteViaSQL")
      .getOrCreate()
```

```

val projectName = spark.sparkContext.getConf.get("odps.project.
name")
val tableName = "cupid_wordcount"

// get a ODPS table as a DataFrame
val df = spark.table(tableName)
println(s"df.count: ${df.count()}")

// Just do some query
spark.sql(s"select * from $tableName limit 10").show(10, 200)
spark.sql(s"select id, count(id) from $tableName group by id").
show(10, 200)

// any table exists under project could be use
// productRevenue
spark.sql(
  """
  |SELECT product,
  |        category,
  |        revenue
  |FROM
  |    (SELECT product,
  |           category,
  |           revenue,
  |           dense_rank() OVER (PARTITION BY category
  |                               ORDER BY revenue DESC) AS rank
  |    FROM productRevenue) tmp
  |WHERE rank <= 2
  """
    .stripMargin).show(10, 200)

  spark.stop()
}
}

```

Run the following command to submit the job:

```

bin/spark-submit \
--master yarn-cluster \
--class com.aliyun.odps.spark.examples.OdpsTableReadWrite \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar

```

1.12.5.5 MaxCompute self-developed Console mode example

For safety reasons, machines in MaxCompute can not be directly connected.

Therefore, the yarn-clientmode in native Spark cannot be used. To enable interaction, the MaxCompute team developed a proprietary client mode.

Example:

```

package com.aliyun.odps.spark.examples

import com.aliyun.odps.cupid.client.spark.client.CupidSparkClientRunner

object SparkClientNormalFT {
  def main(args: Array[String]) {
    val cupidSparkClient = CupidSparkClientRunner.getReadyCupidSparkClient()
  }
}

```



```

    val jarPath = args(0) //client-jobexamples jar path
    val sparkClientNormalApp = new SparkClientNormalApp(cupidSpark
Client)
    sparkClientNormalApp.runNormalJob(jarPath)
    cupidSparkClient.stopRemoteDriver()
  }
}

```

Run the following command to submit the job:

```

java -cp \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar:$SPARK_HOME/jars/* \
com.aliyun.odps.spark.examples.SparkClientNormalFT /path/to/aliyun-
cupid-sdk/examples/client-jobexamples/target/client-jobexamples_2.11-1
.0.0-SNAPSHOT.jar

```

1.12.5.6 MaxCompute Table PySpark example

Use PySpark to read/write MaxCompute tables.

Example:

```

from odps.odps_sdk import OdpsOps
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext, DataFrame

if __name__ == '__main__':
    conf = SparkConf().setAppName("odps_pyspark")
    sc = SparkContext(conf=conf)
    sql_context = SQLContext(sc)

    project_name = "cupid_testal"
    in_table_name = "cupid_wordcount"
    out_table_name = "cupid_wordcount_py"

    normal_df = OdpsOps.read_odps_table(sql_context, project_name,
in_table_name)

    for i in normal_df.sample(False, 0.01).collect():
        print i
    print "Read normal odps table finished"

    OdpsOps.write_odps_table(sql_context, normal_df.sample(False, 0.
001), project_name, out_table_name)
    print "Write normal odps table finished"

```

Run the following command to submit the job:

```

spark-submit \
--master yarn-cluster \
--jars /path/to/aliyun-cupid-sdk/external/cupid-datasource/target/
cupid-datasource_2.11-1.0.0-SNAPSHOT.jar \
--py-files /path/to/aliyun-cupid-sdk/examples/spark-examples/src/main/
python/odps.zip \

```

```
/path/to/aliyun-cupid-sdk/examples/spark-examples/src/main/python/  
odps_table_rw.py
```

1.12.5.7 Mllib example

We recommend that you use OSS for read/write operations in the Mllib model.

Example:

```
package com.aliyun.odps.spark.examples.mllib  
  
import org.apache.spark.mllib.clustering.KMeans._  
import org.apache.spark.mllib.clustering.{ KMeans, KMeansModel}  
import org.apache.spark.mllib.linalg.Vectors  
import org.apache.spark.sql.Session  
  
object KmeansModelSaveToOss {  
  val modelOssDir = "oss://bucket/kmeans-model"  
  
  def main(args: Array[String]) {  
  
    //1. train and save the model  
    val spark = SparkSession  
      .builder()  
      .appName("KmeansModelSaveToOss")  
      .config("spark.hadoop.fs.oss.accessKeyId", "***")  
      .config("spark.hadoop.fs.oss.accessKeySecret", "***")  
      .config("spark.hadoop.fs.oss.endpoint", "***")  
      .getOrCreate()  
  
    val sc = spark.sparkContext  
    val points = Seq(  
      Vectors.dense(0.0, 0.0),  
      Vectors.dense(0.0, 0.1),  
      Vectors.dense(0.1, 0.0),  
      Vectors.dense(9.0, 0.0),  
      Vectors.dense(9.0, 0.2),  
      Vectors.dense(9.2, 0.0)  
    )  
    val rdd = sc.parallelize(points, 3)  
    val initMode = K_MEANS_PARALLEL  
    val model = KMeans.train(rdd, k = 2, maxIterations = 2, runs = 1,  
initMode)  
    val predictResult1 = rdd.map(feature => "cluster id: " + model  
.predict(feature) + " feature:" + feature.toArray.mkString(",")).  
collect  
    println("modelOssDir=" + modelOssDir)  
    model.save(sc, modelOssDir)  
  
    //2. predict from the oss model  
    val modelLoadOss = KMeansModel.load(sc, modelOssDir)  
    val predictResult2 = rdd.map(feature => "cluster id: " +  
modelLoadOss.predict(feature) + " feature:" + feature.toArray.mkString  
(",")).collect  
    assert(predictResult1.size == predictResult2.size)  
    predictResult2.foreach(result2 => assert(predictResult1.contains(  
result2)))  
  }  
}
```

```
}
```

Run the following command to submit the job:

```
./bin/spark-submit  
--jars cupid/hadoop-aliyun-package-3.0.0-alpha2-odps-jar-with-  
dependencies.jar  
--class com.aliyun.odps.spark.examples.mllib.KmeansModelSaveToOss  
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-  
examples_2.11-1.0.0-SNAPSHOT-shaded.jar
```

1.12.5.8 PySpark interactive execution example

PySpark can only run on a host that can be directly connected to a computing cluster.

Install Python 2.7 and set the following path:

```
export PATH=/path/to/python/bin/:$PATH
```

Start command:

```
bin/pyspark --master yarn-client
```

Interactive execution:

```
df=spark.sql("select * from spark_user_data")  
df.show()
```

1.12.5.9 Spark-shell interactive execution example (read tables)

Spark-shell can only run on a host that can be directly connected to a computing cluster.

Start command:

```
bin/spark-shell --master yarn
```

Interactive execution:

```
sc.parallelize(0 to 100, 2).collect  
sql("show tables").show  
sql("select * from spark_user_data").show(200,100)
```

1.12.5.10 Spark-shell interactive execution example (MLlib and OSS read/write)

Spark-shell can only run on a host that can be directly connected to a computing cluster.

Add the following configuration to `conf/spark-defaults.conf`:

```
spark.hadoop.fs.oss.accessKeyId=***  
spark.hadoop.fs.oss.accessKeySecret=***  
spark.hadoop.fs.oss.endpoint=***
```

Start command:

```
bin/spark-shell --master yarn --jars cupid/hadoop-aliyun-package-3.0.0  
-alpha2-odps-jar-with-dependencies.jar
```

Interactive execution:

```
import org.apache.spark.mllib.clustering.KMeans._  
import org.apache.spark.mllib.clustering.{ KMeans, KMeansModel}  
import org.apache.spark.mllib.linalg.Vectors  
val modelOssDir = "oss://your_bucket/kmeans-model"  
val points = Seq(  
  Vectors.dense(0.0, 0.0),  
  Vectors.dense(0.0, 0.1),  
  Vectors.dense(0.1, 0.0),  
  Vectors.dense(9.0, 0.0),  
  Vectors.dense(9.0, 0.2),  
  Vectors.dense(9.2, 0.0)  
)  
val rdd = sc.parallelize(points, 3)  
val initMode = K_MEANS_PARALLEL  
val model = KMeans.train(rdd, k = 2, maxIterations = 2, runs = 1,  
  initMode)  
val predictResult1 = rdd.map(feature => "cluster id: " + model.predict  
  (feature) + " feature:" + feature.toArray.mkString(",")).collect  
println("modelOssDir=" + modelOssDir)  
model.save(sc, modelOssDir)  
val modelLoadOss = KMeansModel.load(sc, modelOssDir)  
val predictResult2 = rdd.map(feature => "cluster id: " + modelLoadO  
  ss.predict(feature) + " feature:" + feature.toArray.mkString(",")).  
collect  
assert(predictResult1.size == predictResult2.size)  
predictResult2.foreach(result2 => assert(predictResult1.contains(  
  result2)))
```

1.12.5.11 SparkR interactive execution example

SparkR can only be run on a host which can directly connect to a computing cluster.

In addition, you must install R in the `/home/admin/R` directory and set the path:

```
export PATH=/home/admin/R/bin/:$PATH
```

Start command:

```
bin/sparkR --master yarn --archives . /R/R.zip
```

Interactive execution:

```
df <- as.DataFrame(faithful)  
df
```

```
head(select(df, df$eruptions))
head(select(df, "eruptions"))
head(filter(df, df$waiting < 50))

results <- sql("FROM spark_user_data SELECT *")
head(results)
```

1.12.5.12 GraphX-PageRank example

Spark on MaxCompute supports native GraphX.

Example:

```
package com.aliyun.odps.spark.examples.graphx

import org.apache.spark.{ SparkConf, SparkContext}
import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD

object PageRank {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("pagerank")
    val sc = new SparkContext(conf)

    // construct vertices
    val users: RDD[(VertexId, Array[String])] = sc.parallelize(List(
      "1,BarackObama,Barack Obama",
      "2,ladygaga,Goddess of Love",
      "3,jeresig,John Resig",
      "4,justinbieber,Justin Bieber",
      "6,matei_zaharia,Matei Zaharia",
      "7,odersky,Martin Odersky",
      "8,anonsys"
    ).map(line => line.split(",")).map(parts => (parts.head.toLong,
      parts.tail)))

    // construct edges
    val followers: RDD[Edge[Double]] = sc.parallelize(Array(
      Edge(2L,1L,1.0),
      Edge(4L,1L,1.0),
      Edge(1L,2L,1.0),
      Edge(6L,3L,1.0),
      Edge(7L,3L,1.0),
      Edge(7L,6L,1.0),
      Edge(6L,7L,1.0),
      Edge(3L,7L,1.0)
    ))

    // construct graph
    val followerGraph: Graph[Array[String], Double] = Graph(users,
      followers)

    // restrict the graph to users with usernames and names
    val subgraph = followerGraph.subgraph(vpred = (vid, attr) => attr.
      size == 2)

    // compute PageRank
    val pageRankGraph = subgraph.pageRank(0.001)

    // get attributes of the top pagerank users
    val userInfoWithPageRank = subgraph.outerJoinVertices(pageRankGr
      aph.vertices) {
```

```

        case (uid, attrList, Some(pr)) => (pr, attrList.toList)
        case (uid, attrList, None) => (0.0, attrList.toList)
    }

    println(userInfoWithPageRank.vertices.top(5)(Ordering.by(_._2._1
)).mkString("\n"))
  }
}

```

Run the following command to submit the job:

```

bin/spark-submit \
--master yarn-cluster \
--class com.aliyun.odps.spark.examples.graphx.PageRank \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar

```

1.12.5.13 Spark Streaming - NetworkWordCount example

Spark on MaxCompute supports native Spark Streaming. To use

NetworkWordCount, you need to install Netcat on your local host, and then run the following command:

```
$ nc -lk 9999
```

The input on the console then becomes the input of Spark Streaming.

Example:

```

package com.aliyun.odps.spark.examples.streaming

import org.apache.spark.SparkConf
import org.apache.spark.examples.streaming.StreamingExamples
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{ Seconds, StreamingContext }

object NetworkWordCount {
  def main(args: Array[String]) {
    if (args.length < 2) {
      System.err.println("Usage: NetworkWordCount <hostname> <port>")
      System.exit(1)
    }

    StreamingExamples.setStreamingLogLevels()

    // Create the context with a 1 second batch size
    val sparkConf = new SparkConf().setAppName("NetworkWordCount")
    val ssc = new StreamingContext(sparkConf, Seconds(1))

    // Create a socket stream on target ip:port and count the
    // words in input stream of \n delimited text (eg. generated by '
    nc')
    // Note that no duplication in storage level only for running
    locally.
    // Replication necessary in distributed scenario for fault
    tolerance.
    val lines = ssc.socketTextStream(args(0), args(1).toInt,
    StorageLevel.MEMORY_AND_DISK_SER)

```

```
val words = lines.flatMap(_.split(" "))
val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
wordCounts.print()
ssc.start()
ssc.awaitTermination()
}
```

Run the following command to submit the job:

```
bin/spark-submit \
--master local[4] \
--class com.aliyun.odps.spark.examples.streaming.NetworkWordCount \
/path/to/aliyun-cupid-sdk/examples/spark-examples/target/spark-
examples_2.11-1.0.0-SNAPSHOT-shaded.jar localhost 9999
```

1.12.6 Maven dependencies

The GitHub project mentioned earlier can be used as your quick start template. For custom development, use the following pom.xml file.

Use Spark community edition 2.3.0 and ensure that the scope is provided.

```
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-core_2.11</artifactId>
  <version>2.3.0</version>
  <scope>provided</scope>
</dependency>
```

The MaxCompute plugin has been released to the Maven warehouse. Add the following dependencies:

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-core_2.11</artifactId>
  <version>1.0.0</version>
  <scope>provided</scope>
</dependency>

<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-client_2.11</artifactId>
  <version>1.0.0</version>
</dependency>

<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-datasource_2.11</artifactId>
  <version>1.0.0</version>
</dependency>
```

The file list in the Maven warehouse is as follows:

- **The core code of the Cupid platform encapsulates the Cupid task submission API and the parent-child process read/write table API.**

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-core_2.11</artifactId>
  <version>1.0.0</version>
</dependency>
```

- **The datasource encapsulates the spark-related MaxCompute table read/write API.**

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-datasource_2.11</artifactId>
  <version>1.0.0</version>
```

- **The SDK encapsulates the Cupid client mode.**

```
<dependency>
  <groupId>com.aliyun.odps</groupId>
  <artifactId>cupid-client_2.11</artifactId>
  <version>1.0.0</version>
</dependency>
```

1.12.7 Special notes

1.12.7.1 Streaming tasks

MaxCompute also supports Spark Streaming. To support long-running tasks, add the following special configurations to `spark-defaults.conf`.

Compared with offline jobs, streaming jobs have additional configurations, which take effect immediately after they are completed in `spark-defaults.conf`.

```
spark.hadoop.odps.cupid.engine.running.type=longtime
-- Set the type to longtime so that the job will not be reclaimed.
spark.hadoop.odps.cupid.job.capability.duration.hours=25920
-- Set the duration.
spark.yarn.maxAppAttempts=10
-- Set the maximum number of retries for a failover.
spark.streaming.receiver.writeAheadLog.enable=true
-- Determine whether to enable the writeAheadLog mode. This mode
prevents data loss but lowers the efficiency.
```

1.12.7.2 Tracking Url

After submitting the job, you will usually have the following output:

```
17/08/28 14:53:26 INFO Client:
client token: N/A
diagnostics: N/A
ApplicationMaster host: 11.137.199.2
ApplicationMaster RPC port: 57524
```



```

queue: queue
start time: 1503903179541
final status: SUCCEEDED
tracking URL: http://jobview.odps.aliyun-inc.com/proxyview/jobview/?h=
http://service.
odps.aliyun-inc.com/api&p=odps_public_dev&i=20170828065141675g5h
4t6u1&t=spark&id= application_1503903039442_1185611255&metaname
=20170828065141675g5h4t6u1&token= L0dSMHRkSlNXS2ZkeFE1UkVsckthTT
ZQWHV3PSxPRFBTX09CTzoxMDU5NTgyNzI0MzIyOT k5LDE1MDQxNjIzODMsey
JTdGF0ZWl1bnQiOlt7IkFjdGlvb2I6WyJvZHBzOlJlYWQiXSwi RWZmZWNOIj
oiQWxsY3ciLCJSZXNvdXJjZSI6WyJhY3M6b2RwczoqOnByb2ply3RzL29kcH
NfcHVibGljX2Rldi9pbN0YW5jZXMvMjAxNzA4MjgwNjUxNDE2NzVnNWg0dDZ1MSJdfV0s
IlZlcnNpb24iOiIxIn0=
user: user

```

TrackingUrl in the output indicates that your job has been submitted to the **MaxCompute** cluster.



Note:

TrackingUrl in the output is crucial because it is the URL of both **SparkWebUI** and **HistoryServer**.

1.12.8 APIs supported by Spark

1.12.8.1 Spark Shell

Run the following commands to start the application.

```

$cd $SPARK_HOME
-- Access the spark directory.
<b>Start command</b>: <codeblock>bin/spark-shell --master yarn</
codeblock>
-- Select a running mode and start the application.

```

Example:

```

sc.parallelize(0 to 100, 2).collect
sql("show tables").show
sql("select * from spark_user_data").show(200,100)

```

1.12.8.2 Spark R

Run the following commands to start the application:

```

$mkdir -p /home/admin/R && unzip . /R/R.zip -d /home/admin/R/
-- Create directory R and decompress R.zip in the directory.
$export PATH=/home/admin/R/bin/:$PATH
-- Set environment variables.
$bin/sparkR --master yarn --archives . /R/R.zip
-- Select a running mode and start the application.

```

Example:

```
df <- as.DataFrame(faithful)
```

```
df
head(select(df, df$eruptions))
head(select(df, "eruptions"))
head(filter(df, df$waiting < 50))
results <- sql("FROM spark_user_data SELECT *")
head(results)
```

1.12.8.3 Spark SQL

Run the following commands to start the application:

```
$cd $SPARK_HOME
-- Access the spark directory.
$bin/spark-sql --master yarn
-- Select a running mode and start the application.
```

Example:

```
show tables;
select * from spark_user_data limit 3;;
quit;
```

1.12.8.4 Spark JDBC

Run the following commands to start an application:

```
$sbin/stop-thriftserver.sh
-- Stop a thread.
/sbin/start-thriftserver.sh
-- Restart a thread.
$bin/beeline
-- Start an application.
```

Example:

```
! connect jdbc:hive2://localhost:10000/odps_smoke_test
show tables;
select * from mr_input limit 3;
! quit
```

1.12.9 Spark dynamic resource allocation

Background

Spark provides a large number of configuration items to implement a wide array of semantics. You can use the default values for most configuration items, but there are some items require manual configurations. Of these items, spark.executor.instances is the most complicated.

A value that is too small will cause operations to run slowly or fail, while a value that is too large will waste resources.

Even the optimal value based on full understanding of the data and logic is not reliable. For complex jobs, the number of executors required at different stages is different and different resources are required during the job execution process. Fixed configurations of resources will cause a waste. When long tail latency occurs, idle resources will be occupied by other executors even for simple jobs.

Solution

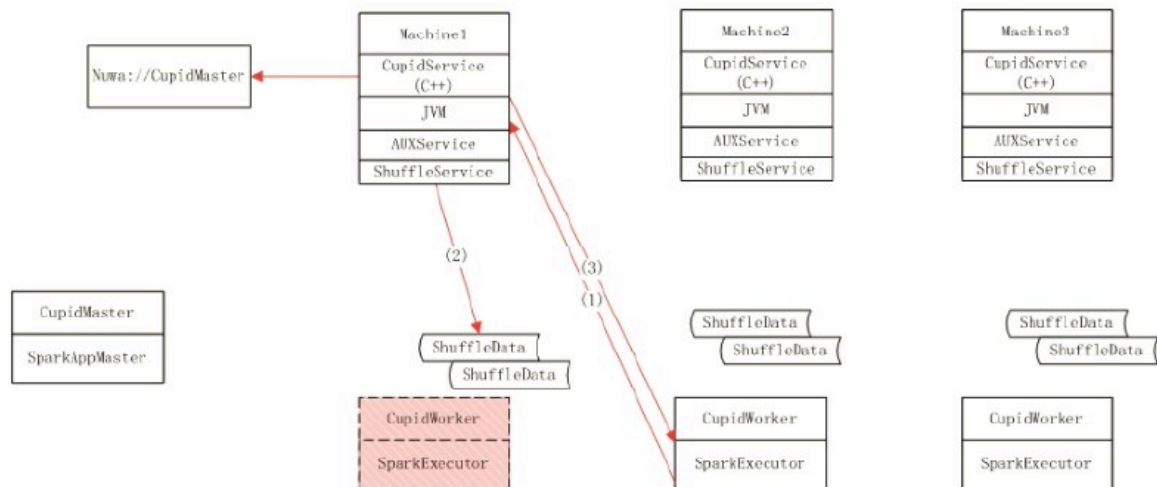
The best solution is to allocate resources as needed. Dynamic Resource Allocation (DRA) is such a solution. The CupidService developed by the Cupid team supports native DRA. Its implementation process is as shown in the following figure.



Note:

For more information about the community native DRA and related configurations, see [Dynamic Resource Allocation in Job Scheduling](#) and [Dynamic Allocation in Spark Configuration](#).

Figure 1-11: Implementation process



Enable DRA

You must add the following settings to enable DRA. You do not need to modify the code.

```
spark.hadoop.odps.cupid.shuffleservice.enable=true // Required, the
flag at the Cupid end.
spark.hadoop.odps.cupid.disk.driver.enable=true // Required, the
dependent item.
spark.dynamicAllocation.enabled=true // Required, the flag at the
Spark end.
spark.shuffle.service.enabled=true // Required, the dependent item.
```

```
spark.shuffle.service.port=7338 // Required, the shuffle service port.  
spark.authenticate=true // Required, authentication.  
spark.dynamicAllocation.maxExecutors=128 // Optional, the maximum  
number of executors.  
spark.dynamicAllocation.minExecutors=1 // Optional, the minimum number  
of executors.  
spark.dynamicAllocation.initialExecutors=1 // Optional, the initial  
number of executors.  
spark.dynamicAllocation.executorIdleTimeout=60s // Optional, the  
waiting period before idle executors are released.
```

When DRA is enabled, `spark.executor.instances` is optional and equivalent to `spark.dynamicAllocation.initialExecutors`.

1.13 Elasticsearch on Maxcompute

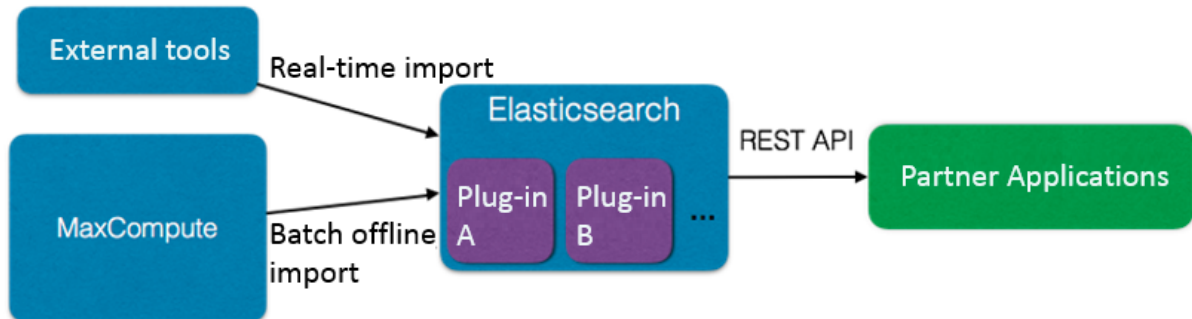
1.13.1 Overview

Elasticsearch on MaxCompute is an enterprise-class full-text retrieval system developed by Alibaba Cloud for retrieving large volumes of data. It provides near-real-time (NRT) search performance for government agencies and enterprises. Elasticsearch on MaxCompute provides elastic full-text retrieval and supports native Elasticsearch APIs. Based on the APIs, it supports data import from heterogeneous data sources as well as cluster and service OAM. Based on the centralized scheduling and management capabilities of MaxCompute, Elasticsearch provides more efficient core services for retrieving large volumes of data. Elasticsearch on MaxCompute can also work with plugins available from the Elasticsearch open source community to provide a range of retrieval features.

You can import data to Elasticsearch on MaxCompute using external tools in real time or use MaxCompute to import offline data in batches. After indexing imported data, Elasticsearch on MaxCompute provides the retrieval service

through the RESTful API. The following figure shows the usage of Elasticsearch on MaxCompute.

Figure 1-12: Elasticsearch on MaxCompute usage



1.13.2 Workflow

1.13.2.1 Overview

Elasticsearch on MaxCompute is based on the open source Elasticsearch. It can run the Elasticsearch service on MaxCompute clusters.

In the MaxCompute client, you can start and manage your Elasticsearch service as needed, including the number of nodes, disk space, memory size, and custom settings. The resources consumed by the Elasticsearch service are counted towards your MaxCompute instance quota.

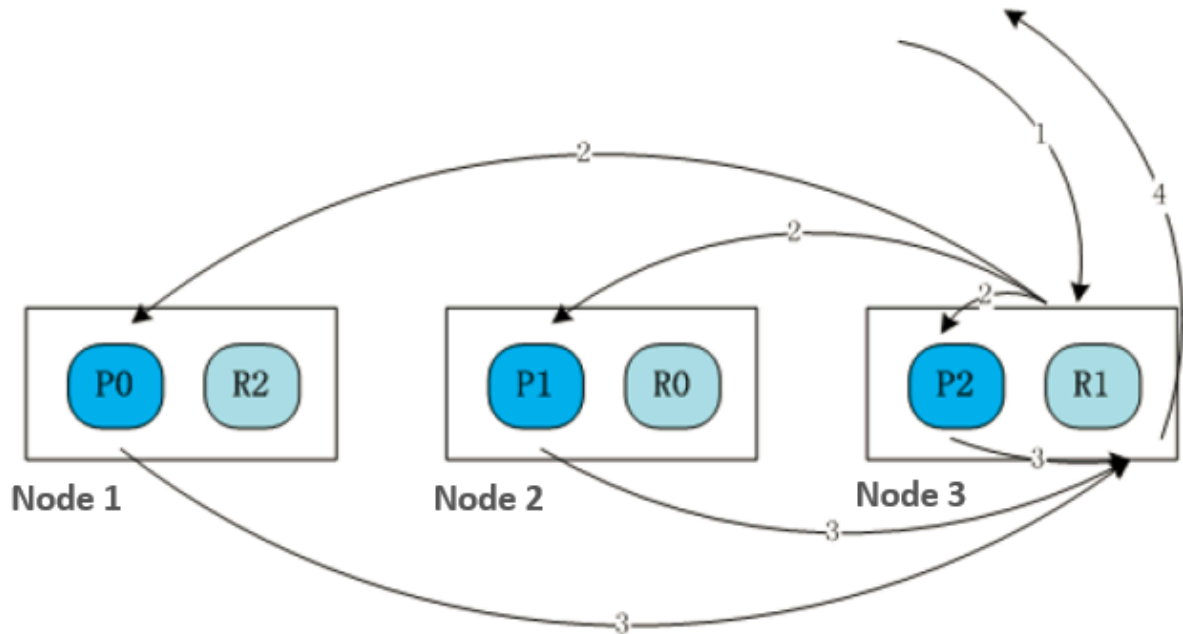
For the process of starting Elasticsearch, see the workflow chart in [Typical practice of Elasticsearch](#).

The following topics describe the workflows of Elasticsearch on MaxCompute features.

1.13.2.2 Distributed retrieval workflow

The following figure shows the distributed retrieval workflow:

Figure 1-13: Distributed retrieval workflow



Each cluster consists of three nodes, as shown in the preceding figure. The index has three shards: P0, P1, and P2, which are distributed across three nodes. The shards work in 1:1 backup mode, so there are three replicas: R0, R1, and R2. The retrieval process is as follows:

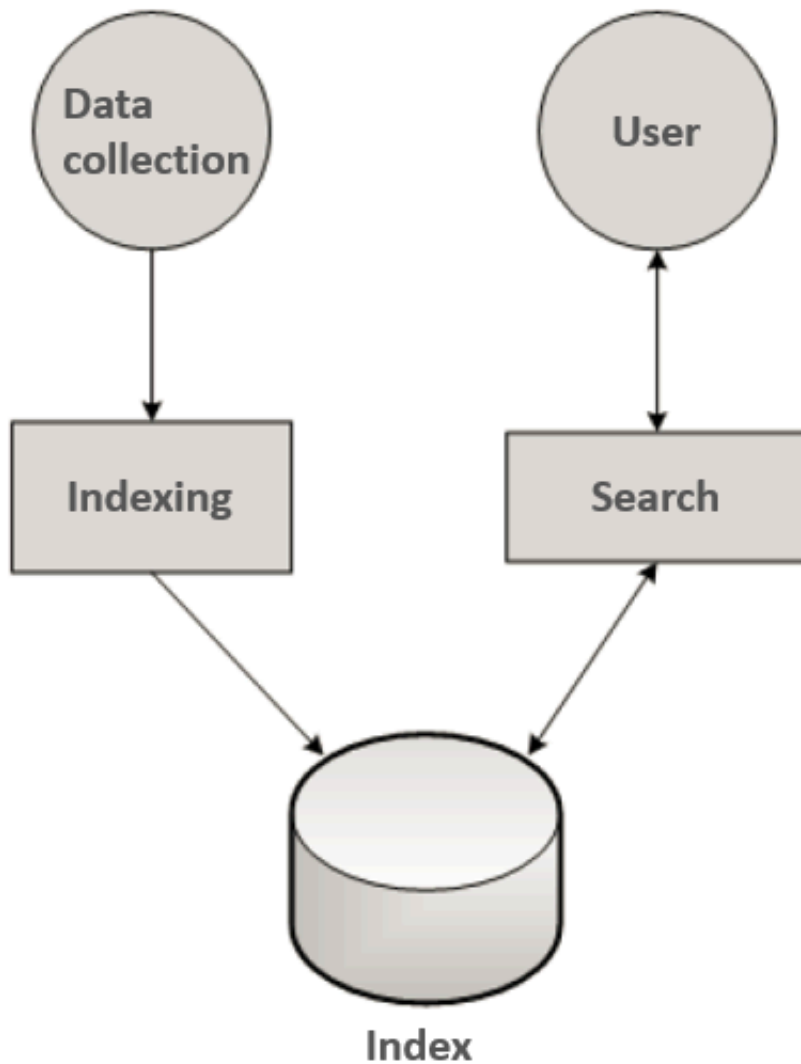
1. A user sends a retrieval request to node 3.
2. After receiving the request, node 3 sends a retrieval request (2) to P0, P1, and P2 based on the recorded index shard information.
3. The nodes where P0, P1, and P2 are located search for the requested information in the specified shards. Each node sends a retrieval result message (3) to node 3.
4. Node 3 combines the retrieval results from the other nodes, and returns a result to the user in an acknowledgment message (4).

Because multiple nodes perform data retrieval at the same time, the retrieval speed is improved. The performance of distributed retrieval increases with the number of nodes.

1.13.2.3 Full-text retrieval process

The following figure shows the full-text retrieval process.

Figure 1-14: Full-text retrieval process



The process is as follows:

1. The data collection module collects structured and unstructured data, converts the data into the field + value format, and submits the data to the index module.
2. The index module receives the data in the field + value format, performs segmentation and creates inverted indexes based on the predefined indexing method, and saves the indexes. The field type, indexing method, and segmentation rule are configured on the retrieval management page.

3. The search module receives and processes user requests. It parses the requests to obtain indexes, fields, and query statements. Then, the search module finds matching records from the inverted indexes.
4. The search module returns data that satisfies the user requirements, such as the sorting rule and quantity.

1.13.2.4 Authentication process

The authentication process is as follows:

1. You try to log on to the Elasticsearch on MaxCompute retrieval management or O&M platform. You are redirected to the authentication module. If you pass the authentication, you can access the platform. Otherwise, you are denied access to the platform.
2. The administrator can use the MaxCompute client to add Elasticsearch users and configure permissions for the users.
3. When you try to access the index library through an API, the system implements authentication. You can search for or operate data in the index library only after passing the authentication.

1.13.3 Quick start

Before you start an Elasticsearch service, determine the following issues:

- **Node planning:** Determine the number of nodes required for each role in the Elasticsearch cluster. An Elasticsearch cluster has two roles by default, `master` and `data`. Each role is deployed on three nodes. You can add nodes to the cluster at any time.
- **Resource planning:** Determine the CPU, memory, and disk space resources required for each node. The resources configured for each node cannot be changed later. 8 GB memory and 20 GB disk space are allocated to each node by default.



Note:

Note that only 50% of the memory allocated to a data node is used for the JVM heap.

- **Elasticsearch configuration:** Determine the running configurations of Elasticsearch nodes, such as the queue size of bulk requests, and support for cross-domain HTTP requests.

After you determine the preceding issues, you can start your Elasticsearch service in the MaxCompute client.

The following example shows how to quickly start a small Elasticsearch cluster based on the default configuration. The name of the Elasticsearch service is `es_first_cluster`.

1. Download a [MaxCompute client](#) that supports Elasticsearch. Configure the AccessKey, project, and endpoint.



Note:

Note that the download link here provides a non-standard version of the MaxCompute client.

2. Start `odpscmd` and run the following command:

```
server create es_first_cluster type elasticsearch_mdu;
```

Wait for several minutes. If OK is returned, the Elasticsearch service is started successfully. Then, create an Elasticsearch user.

3. In `odpscmd`, run the following command to create an Elasticsearch user with the permission of `all_access`:

```
server execute es_first_cluster create user admin with password 123456|all_access;
```

If OK is returned, the Elasticsearch user is created.

4. Access the Elasticsearch service. Assuming that the service is started in a MaxCompute project named `prj1`, run the following command in `odpscmd` to view the Elasticsearch cluster information:

```
curl -u admin:123456 http://search.aliyun.com:9200/prj1.es_first_cluster
```



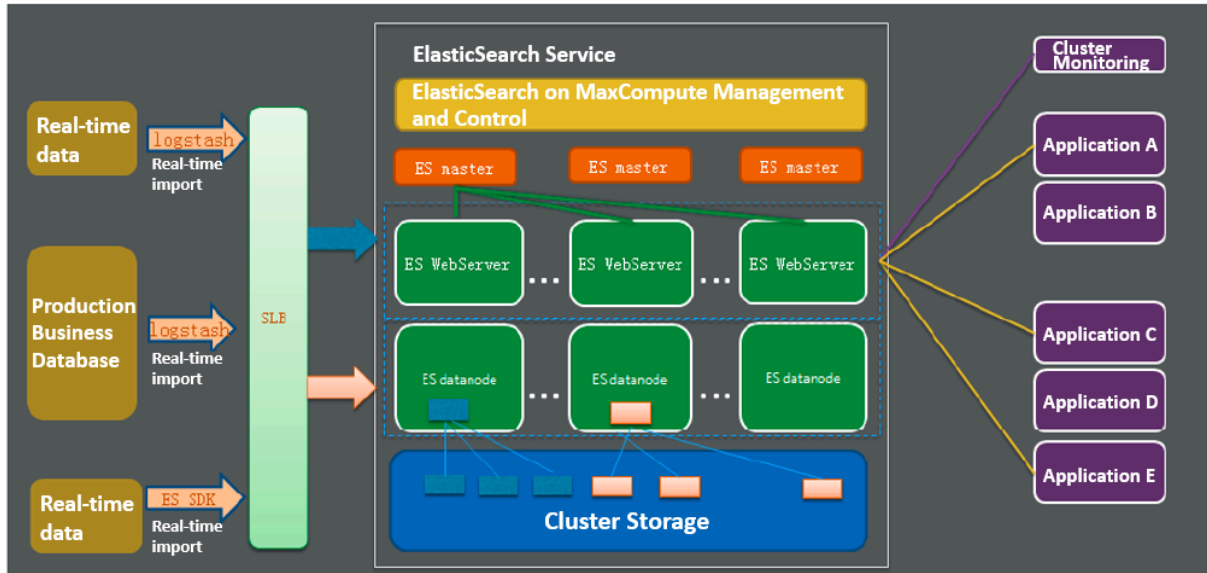
Note:

You can run the `server delete es_first_cluster` command to delete an Elasticsearch service. Note that this command deletes data irrevocably.

1.13.4 Support for Elasticsearch applications

1.13.4.1 ElasticSearch typical practice

Figure 1-15: Typical practice



Elasticsearch on MaxCompute allows you to start a set of Elasticsearch services on MaxCompute clusters by submitting a job. The project does not modify the native Elasticsearch code. Elasticsearch on MaxCompute works in the same way as the native Elasticsearch cluster.

1.13.4.2 Elasticsearch on MaxCompute support for VPC

Alibaba Cloud Elasticsearch on MaxCompute is an enterprise-class full-text retrieval system for retrieving massive amounts of data. To comply with data isolation and security requirements, Elasticsearch on MaxCompute provides support for Virtual Private Cloud (VPC) networks so that you can apply access policies at VPC level. (Elasticsearch VPC limits).

Elasticsearch on MaxCompute supports VPC networks in the following model:

- Classic networks, VPC, and the Internet are isolated from each other. Users can access only the endpoints and virtual IP addresses (VIPs) on their networks.
- Projects without a whitelist of VPC IDs and IP addresses are accessible for users from valid domains over the three types of networks. A domain is valid only if its access request is acknowledged.

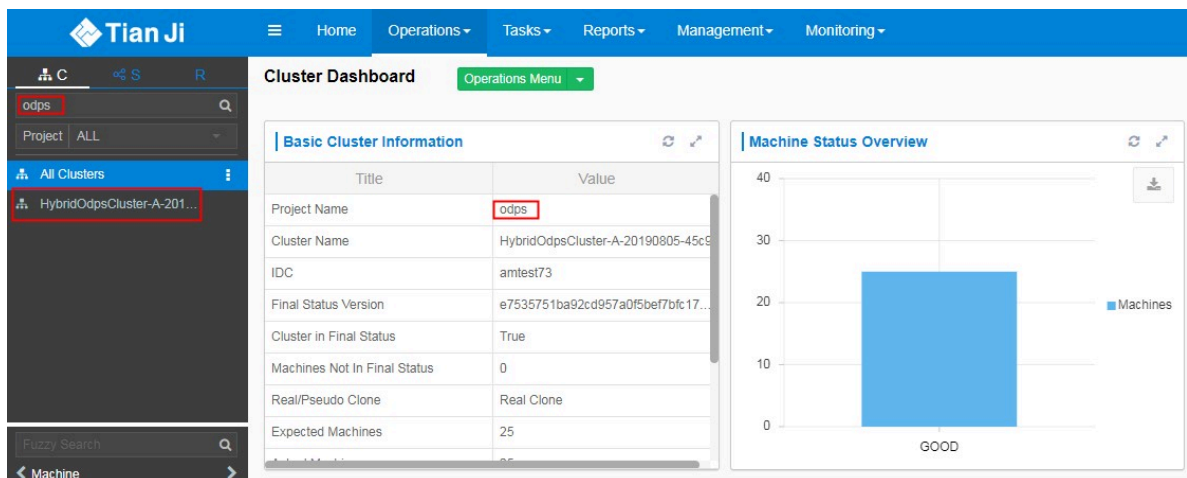
- When an ElasticSearch service instance is started in a MaxCompute project, they share the same VPCLIST, which is a whitelist of VPCs.
- Starting an ElasticSearch service instance occupies all resources by default. You must scale up the MaxCompute instance or scale down the ElasticSearch service instance if you start more ElasticSearch service instances.

Here is a specific use example. Starting one ElasticSearch service instance for each project is taken as the default practice when a MaxCompute VPC is deployed. You can start your Elasticsearch instances for your projects, apply for domain names and VIPs, and perform VPC health check in the Elasticsearch frontend.

1.13.5 Special notes

1.13.5.1 Find the Elasticsearch service domain name

1. Log on to the Apsara Infrastructure Management Framework console. Choose **Operations > Cluster Operations** from the top navigation bar. In the left-side navigation pane, click the **C** tab and search for ODPS.



2. Navigate to the MaxCompute cluster resource usage page.
3. Find the Elasticsearch service domain name.

service	serverole	app	n...	ty...	S...	...	result
odps-service-computer	odps-service-computer.Com...	com...	odps	ots	done	{ "e...	{ "instance_name": "odps", "db_name": "TIANJi-A-A2B4", "db_user": "1067309636274922", "do...
odps-service-console	odps-service-console.Cupid...	cupi...	odps...	dns	done	{ "d...	{ "ip": "10.36.103.41", "domain": "jobview.cn-hangzhou-env6-d01.odps.aliyun-inc.com", "dns...
odps-service-console	odps-service-console.LogVie...	log_v...	odps...	dns	done	{ "d...	{ "ip": "10.42.36.0.208", "domain": "logview.cn-hangzhou-env6-d01.odps.aliyun-inc.com", "dns": "lo...
odps-service-console	odps-service-console.WebC...	web...	odps...	dns	done	{ "d...	{ "ip": "10.36.103.155", "domain": "webconsole.cn-hangzhou-env6-d01.odps.aliyun-inc.com", "dns...
odps-service-console	odps-service-console.WebC...	web...	odps...	dns	done	{ "d...	{ "ip": "10.42.36.0.210", "domain": "webconsole.cn-hangzhou-env6-d01.odps.aliyun-inc.com", "dns...
odps-service-es	odps-service-es.ElasticSearc...	elasti...	odps...	dns	done	{ "d...	{ "ip": "10.36.103.65", "domain": "elasticsearch.cn-hangzhou-env6-d01.odps.aliyun.com", "dns": "e...
odps-service-frontend	odps-service-frontend.Fronte...	front...	odps...	dns	done	{ "d...	{ "ip": "10.36.102.181", "domain": "service.cn-hangzhou-env6-d01.odps.aliyun.com", "dns": "s...
odps-service-frontend	odps-service-frontend.Tunnel...	turn...	odps...	dns	done	{ "d...	{ "ip": "10.36.102.177", "domain": "dt.cn-hangzhou-env6-d01.odps.aliyun.com", "dns": "dt.cn...

1.13.5.2 Import table data from MaxCompute to Elasticsearch

Before you can use Elasticsearch on MaxCompute to search for MaxCompute table data, you must import table data from MaxCompute to Elasticsearch clusters.

MaxCompute MapReduce is designed to carry out the task of exporting data from MaxCompute to Elasticsearch. You can import data to Elasticsearch through simple configurations.

By utilizing the distributed dispatching capability of MaxCompute, you can easily control the concurrency. Besides, you can add the MapReduce job to the scheduled tasks on D2.

The data import process is as follows:

1. Download the JAR package of the MapReduce job.
2. Run the following command to add the JAR package to the MaxCompute resource files in the MaxCompute console:

```
add jar /PATH/T0/elasticsearch_output-1.0.0.jar
```

3. Create configuration file `es_mr.conf` in the following format for the MapReduce job:

```
<configuration>
  <property>
    <name>key1</name>
    <value>value1</value>
  </property>
  <property>
    <name>key2</name>
    <value>value2</value>
  </property>
</configuration>
```

4. Submit the MapReduce job in the MaxCompute console. Example:

```
jar -conf es_mr.conf -classpath /PATH/T0/elasticsearch_output-1.0.0.jar
    -resources elasticsearch_output-1.0.0.jar -Dworker_num=5
com.aliyun.odps.export.elasticsearch.mr.EsOutputJob <TABLE_NAME> [
PARTITION_SPEC];
-- Five workers run concurrently to export data from <TABLE_NAME> [
PARTITION_SPEC] to the Elasticsearch cluster.
```

The following table describes the parameters in the `es_mr.conf` file.

Table 1-51: Parameters

Parameter	Example	Required	Default value	Description
es.resource	my_index/ my_type	Yes	-	Index and type in the target Elasticsearch for imported data.
es.nodes	-	Yes	-	Elasticsearch endpoint.
es.nodes.client.only	true	No	false	Send data to client-only nodes only.
es.col.field.mapping	odps_col1: es_field1, odps_col3: es_field2	Yes	-	Mapping between the MaxCompute columns to be imported and the Elasticsearch fields.
es.batch.size.bytes	1 MB	No	1 MB	Batch data transfer size.
es.batch.size.entries	1000	No	1000	Number of data entries transferred each time.
es.net.http.auth.user	admin	No	-	Elasticsearch username.
es.net.http.auth.pass	123456	No	-	Elasticsearch password.
es.mapping.routing	field_routing	No	-	Routing field name, in the <CONSTANT> format for a constant.
es.mapping.id	field_id	No	-	id field of a document.

1.14 Non-structured data access and processing (integrated computing scenarios)

1.14.1 Overview

MaxCompute SQL cannot directly process external data (such as non-structured data from OSS). Data must be imported to MaxCompute tables through relevant tools before computation. The MaxCompute team introduces the non-structured data processing framework to the MaxCompute system architecture to resolve this problem.

You can execute a simple DDL statement to create an external table in MaxCompute and associate MaxCompute tables with external data sources. This table can then act as an interface between MaxCompute and external data sources. You can fully use the computing capabilities of MaxCompute SQL to process data in the external table.

MaxCompute can process the following data sources by creating external tables:

- Internal data sources: OSS, Table Store, AnalyticDB, RDS, HDFS (Alibaba Cloud), and TDDL.
- External data sources: HDFS (open-source), MongoDB, and Hbase.

The following sections illustrate various data sources.

1.14.2 Internal data sources

1.14.2.1 OSS data source

1.14.2.1.1 Preface

As the core computing component of the Alibaba Cloud big data platform, MaxCompute provides powerful computing capabilities. It can schedule large amounts of nodes for parallel computing, and effectively manage failover and retry mechanisms in distributed computing. MaxCompute SQL implements different data processing logics through simple semantics. It is widely used within and outside Alibaba Group. It allows interoperability among different data sources and is significant for the integrated data ecology of Alibaba Cloud.

The following examples show how to access and process OSS unstructured data in MaxCompute.

1.14.2.1.2 Use the built-in extractor to read OSS data

1.14.2.1.2.1 Overview

You can use the MaxCompute built-in extractor to easily read OSS data in the specified format. You only need to create an external table as the source table for data query. For example, a CSV file is stored in OSS. The endpoint is `oss-cn-shanghai-internal.aliyuncs.com`, the bucket is `oss-odps-test`, and the file path is `/demo/SampleData/CSV/AmbulanceData/vehicle.csv`. The following topics provide operation examples.

1.14.2.1.2.2 Create an external table

Run the following command to create an external table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS ambulance_data_csv_external
(
  vehicleId bigint,
  recordId bigint,
  patientId bigint,
  calls bigint,
  locationLatitude double,
  locationLongitude double,
  recordTime string,
  direction string
)
STORED BY 'com.aliyun.odps.CsvStorageHandler'
LOCATION 'oss://<your-id>:<your-secret-key>@oss-cn-shanghai-internal.aliyuncs.com/oss-odps-test/Demo/SampleData/CSV/AmbulanceData/';
```



Note:

- **com.aliyun.odps.CsvStorageHandler** is a built-in **StorageHandler** for processing CSV files. It defines how to read and write CSV files. You only need to specify this name. The relevant logic is implemented by the system.
- **LOCATION** specifies an OSS directory. The system reads all files in the directory by default.
- An external table records only the corresponding OSS directory. When the table is deleted, OSS data stored in the directory specified by **LOCATION** is not deleted.

1.14.2.1.2.3 Query an external table

After an external table is created, you can use it in the same way you use a MaxCompute table.

The content of `/demo/SampleData/CSV/AmbulanceData/vehicle.csv` is as follows:

```
1,1,51,1,46.81006,-92.08174,9/14/2017 0:00,S
1,2,13,1,46.81006,-92.08174,9/14/2017 0:00,NE
1,3,48,1,46.81006,-92.08174,9/14/2017 0:00,NE
1,4,30,1,46.81006,-92.08174,9/14/2017 0:00,W
1,5,47,1,46.81006,-92.08174,9/14/2017 0:00,S
1,6,9,1,46.81006,-92.08174,9/14/2017 0:00,S
1,7,53,1,46.81006,-92.08174,9/14/2017 0:00,N
1,8,63,1,46.81006,-92.08174,9/14/2017 0:00,SW
1,9,4,1,46.81006,-92.08174,9/14/2017 0:00,NE
```

```
1,10,31,1,46.81006,-92.08174,9/14/2017 0:00,N
```

Run the following command to submit a job, which calls a built-in CSV extractor to read data from OSS:

```
SELECT recordId, patientId, direction
FROM ambulance_data_csv_external
WHERE patientId > 25;
```

Command output:

```
+-----+-----+-----+
| recordId | patientId | direction |
+-----+-----+-----+
| 1 | 51 | S |
| 3 | 48 | NE |
| 4 | 30 | W |
| 5 | 47 | S |
| 7 | 53 | N |
| 8 | 63 | SW |
| 10 | 31 | N |
+-----+-----+-----+
```



Note:

The system provides built-in `CsvStorageHandler`, `TsvStorageHandler`, and `TextStorageHandler`.

1.14.2.1.3 Custom extractors

1.14.2.1.3.1 Overview

If OSS data is in a complicated format that cannot be processed by the built-in extractor, you must customize an extractor to read data from OSS files. For example, a text file is stored in OSS. The file is not in the CSV format and the columns of records are separated by vertical bars (|). The file path is `/demo/SampleData/CustomTxt/AmbulanceData/vehicle.csv`. The following topics provide operation examples.

1.14.2.1.3.2 Define StorageHandler

You can customize the data parsing logic. `StorageHandler` is the unified entrance of your custom logic. You can specify the types of custom extractors and outputters. `StorageHandler` provides only a simple definition. For example, you can implement `SpeicalTextStorageHandler`:

```
package com.aliyun.odps.udf.example.text;
public class SpeicalTextStorageHandler extends OdpsStorageHandler {
    @Override
```



```

public Class<? extends Extractor> getExtractorClass() {
    return TextExtractor.class;
}
@Override
public Class<? extends Outputter> getOutputterClass() {
    return TextOutputter.class;
}
}

```

**Note:**

Note that `TextStorageHandler` that is built in MaxCompute can process the data format in the preceding example (text separated by vertical bars (|)). This example is provided only to show you how to use the SDK to customize a `StorageHandler` (especially extractor) for processing uncommonly structured data.

1.14.2.1.3.3 Define an extractor

In the following example, `TextExtractor` is used to extract records from a text file, where the delimiter is imported as a parameter. `TextExtractor` can be used for all text files of the similar format.

```

/**
 * Text extractor that extract schematized records from formatted
 * plain-
 * text(csv, tsv etc.)
 */
public class TextExtractor extends Extractor {
    private InputStreamSet inputs;
    private String columnDelimiter;
    private DataAttributes attributes;
    private BufferedReader currentReader;
    private boolean firstRead = true;
    public TextExtractor() {
        // Default to ",", this can be overwritten if a specific delimiter is
        // provided (via DataAttributes)
        this.columnDelimiter = ",";
    }
    // No particular usage for execution context in this example
    @Override
    public void setup(ExecutionContext ctx, InputStreamSet inputs,
        DataAttributes attributes) {
        this.inputs = inputs;
        -- inputs specifies an InputStreamSet. An InputStream is returned each
        time next() is called. This InputStream can read all the content of
        an OSS file.
        this.attributes = attributes;
        // Check if "delimiter" attribute is supplied via SQL query
        String columnDelimiter = this.attributes.getValueByKey("delimiter");
        -- The delimiter can be used as a parameter in DDL statements.
        if ( columnDelimiter != null)
        {
            this.columnDelimiter = columnDelimiter;
        }
        // note: more properties can be initied from attributes if needed
    }
    @Override

```

```

public Record extract() throws IOException {
    String line = readNextLine();
    if (line == null) {
        return null;
        -- If NULL is returned, all records in the table have been read.
    }
    return textLineToRecord(line);
    -- textLineToRecord splits a row into multiple columns using the
    delimiter. For the implementation process, see Complete TextExtractor
    implementation.
    -- extractor() returns a record that is extracted from OSS data.
}
@Override
public void close(){
    // no-op
}
}

```

1.14.2.1.3.4 Compile and package code

You can compile and package Java code, and run the following command to upload it to MaxCompute. The procedure is the same as that for a normal Java UDF.

```
add jar odps-udf-example.jar;
```

1.14.2.1.3.5 Create an external table

After you upload a JAR package, you need to run the following command to create an external table. This command is similar to the one that you run before using a built-in extractor. The difference is that this command uses a custom `StorageHandler`.

```

CREATE EXTERNAL TABLE IF NOT EXISTS ambulance_data_txt_external
(
    vehicleId int,
    recordId int,
    patientId int,
    calls int,
    locationLatitute double,
    locationLongtitue double,
    recordTime string,
    direction string
)
STORED BY 'com.aliyun.odps.udf.example.text.SpeicalTextStorageHandler'
-- STORED BY specifies the class name of a custom StorageHandler.
WITH SERDEPROPERTIES('delimiter'='|')
-- SERDEPROPERITES can be used to specify parameters. These parameters
are transferred to an extractor through DataAttributes.
LOCATION 'oss://<*>your-id*>:<*>your-secret-key*>@oss-cn-shanghai
-internal.aliyuncs.com/oss-odps-test/Demo/SampleData/CustomTxt/
AmbulanceData/'
USING 'odps-udf-example.jar';

```

```
-- Specify the JAR package where the class definition is located.
```

1.14.2.1.3.6 Query an external table

The content of `/demo/SampleData/CustomTxt/AmbulanceData/vehicle.csv` is as follows:

```
1|1|51|1|46.81006|-92.08174|9/14/2017 0:00|S
1|2|13|1|46.81006|-92.08174|9/14/2017 0:00|NE
1|3|48|1|46.81006|-92.08174|9/14/2017 0:00|NE
1|4|30|1|46.81006|-92.08174|9/14/2017 0:00|W
1|5|47|1|46.81006|-92.08174|9/14/2017 0:00|S
1|6|9|1|46.81006|-92.08174|9/14/2017 0:00|S
1|7|53|1|46.81006|-92.08174|9/14/2017 0:00|N
1|8|63|1|46.81006|-92.08174|9/14/2017 0:00|SW
1|9|4|1|46.81006|-92.08174|9/14/2017 0:00|NE
1|10|31|1|46.81006|-92.08174|9/14/2017 0:00|N
```

Run the following command to submit a job, which calls a custom extractor to read data from OSS:

```
SELECT recordId, patientId, direction
FROM ambulance_data_txt_external
WHERE patientId > 25;
```

Command output:

```
+-----+-----+-----+
| recordId | patientId | direction |
+-----+-----+-----+
| 1 | 51 | S |
| 3 | 48 | NE |
| 4 | 30 | W |
| 5 | 47 | S |
| 7 | 53 | N |
| 8 | 63 | SW |
| 10 | 31 | N |
+-----+-----+-----+
```

1.14.2.1.4 Advanced usage

1.14.2.1.4.1 Use a custom extractor to read external unstructured data

The preceding topic describes how to use built-in and custom extractors to process text files such as .csv files that are stored in OSS. This topic describes how to use UDF extractors to process non-text files in OSS.

The following example shows how to process audio files (.wav files) in OSS.

1. Customize the SpeechSentenceSnrExtractor main logic. Use the SETUP API to read parameters, initialize the parameters, and import the audio processing model (by using the resource function).

```
public SpeechSentenceSnrExtractor(){
    this.utteranceLabels = new HashMap<String, UtteranceLabel>();
}
@Override
public void setup(ExecutionContext ctx, InputStreamSet inputs,
    DataAttributes attributes){
    this.inputs = inputs;
    this.attributes = attributes;
    this.mlfFileName =
        this.attributes.getValueByKey(MLF_FILE_ATTRIBUTE_KEY);
    String sampleRateInKHzStr =
        this.attributes.getValueByKey(SPEECH_SAMPLE_RATE_KEY);
    this.sampleRateInKHz = Double.parseDouble(sampleRateInKHzStr);
    try {
        // read the speech model file from resource and load the model into
        // memory
        BufferedInputStream inputStream =
            ctx.readResourceFileAsStream(mlfFileName);
        loadMlfLabelsFromResource(inputStream);
        inputStream.close();
    } catch (IOException e) {
        throw new RuntimeException("reading model from mlf failed with
            exception
            " + e.getMessage());
    }
}
@Override
public Record extract() throws IOException {
    SourceInputStream inputStream = inputs.next();
    if (inputStream == null){
        return null;
    }
    // process one wav file to extract one output record [snr, id]
    String fileName = inputStream.getFileName();
    fileName = fileName.substring(fileName.lastIndexOf('/') + 1);
    logger.info("Processing wav file " + fileName);
    // infer id from speech file name
    String id = fileName.substring(0, fileName.lastIndexOf('.'));
    // read speech file into memory buffer
    long fileSize = inputStream.getFileSize();
    byte[] buffer = new byte[(int)fileSize];
    int readSize = inputStream.readToEnd(buffer);
    inputStream.close();
    // compute the avg sentence snr from speech file
    double snr = computeSnr(id, buffer, readSize);
    // construct output record [snr, id]
    Column[] outputColumns = this.attributes.getRecordColumns();
    ArrayRecord record = new ArrayRecord(outputColumns);
    record.setDouble(0, snr);
    record.setString(1, id);
    return record;
}
private void loadMlfLabelsFromResource(BufferedInputStream
    fileInputStream)
    throws IOException {
    // loading MLF label from resource, skipped here
}
```

```
// compute the snr of the speech sentence, assuming the input buffer
contains the entire content of a wav file
private double computeSnr(String id, byte[] buffer, int
    validBufferLen){
// computing the snr value for the wav file (supplied as byte buffer
array), skipped here
}
```

**Note:**

The Extractor() API implements the reading and processing logic of audio files. It calculates the signal-to-noise ratio (SNR) of the read data based on the audio processing model and writes the result to a record in [snr, id] format.

2. Run the following commands to create an external table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS speech_sentence_snr_external
(
    sentence_snr double,
    id string
)
STORED BY 'com.aliyun.odps.udf.example.speech.SpeechStorageHandler'
WITH SERDEPROPERTIES (
    'mlfFileName'='sm_random_5_utterance.text.label' ,
    'speechSampleRateInKHz' = '16'
)
LOCATION 'oss://<your-id>:<your-secret-key>@oss-cn-shanghai-
internal.aliyuncs.com/oss-odps-test/dev/SpeechSentenceTest/'
USING 'odps-udf-example.jar,sm_random_5_utterance.text.label';
```

3. Run the following commands to read data from OSS:

```
SELECT sentence_snr, id
FROM speech_sentence_snr_external
WHERE sentence_snr > 10.0;
```

4. The command output is as follows:

```
-----
| sentence_snr | id |
-----
| 34.4703 | J310209090013_H02_K03_042 |
-----
| 31.3905 | tsh148_seg_2_3013_3_6_48_80bd359827e24dd7_0 |
-----
| 35.4774 | tsh148_seg_3013_1_31_11_9d7c87aef9f3e559_0 |
-----
| 16.0462 | tsh148_seg_3013_2_29_49_f4cb0990a6b4060c_0 |
-----
| 14.5568 | tsh_148_3013_5_13_47_3d5008d792408f81_0 |
-----
```

**Note:**

By using the UDF extractor, you can run SQL statements to process multiple audio files in OSS in a distributed manner. Similarly, you can use the large-scale

computation capabilities of MaxCompute to process unstructured data such as images and videos.

1.14.2.1.5 Data partitions

1.14.2.1.5.1 Overview

In the preceding topic, `LOCATION` is used to specify an OSS directory, which is associated with an external table. MaxCompute reads all data in the directory, including all files in the subdirectories. When there is a large volume of data in the directory, a full-text scan will cause extra I/O operations and processing time. There are two solutions:

- **Reducing the data volume:** You need to properly plan data storage addresses. Create multiple external tables for data from different parts, with the `LOCATION` of each external table pointing to a subset of data.
- **Partitioning data:** The external table, like an internal table, supports partitioning. You can create partitions to facilitate data management.

The following topics describe the partition feature of external tables.

1.14.2.1.5.2 Standard organization method and path format of partition data in OSS

Unlike the data in internal tables, data stored in external storage (such as OSS) cannot be managed in MaxCompute. If you need to use the partitioned table feature of MaxCompute, make sure that the data file paths in OSS are in the following format:

```
partitionKey1=value1\partitionKey2=value2\...
```

Example:

1. Your daily log files are stored in OSS, and you want to access some of the data from MaxCompute on a daily basis. If the log files are in the CVS format or a similar custom format, you can execute the following statement to create a partitioned external table:

```
CREATE EXTERNAL TABLE log_table_external (  
  click STRING,  
  ip STRING,  
  url STRING,  
)  
PARTITIONED BY (  
  year STRING,
```

```
month STRING,
day STRING
)
STORED BY 'com.aliyun.odps.CsvStorageHandler'
LOCATION 'oss://<ak_id>:<ak_key>@oss-cn-shanghai-internal.aliyuncs.
com/oss-odps-test/log_data/';
```

**Note:**

In the preceding example, the **PARTITIONED BY** clause is used to specify a partitioned external table. The partition keys are year, month, and day.

2. For the partitions to take effect, you must specify the OSS storage directory in the format shown in the preceding example. An example of a valid directory layout is as follows:

```
osscmd ls oss://oss-odps-test/log_data/
2017-09-14 08:03:35 128MB Standard oss://oss-odps-
test/log_data/year=2017/month=06/day=01/logfile
2017-09-14 08:04:12 127MB Standard oss://oss-odps-
test/log_data/year=2017/month=06/day=01/logfile. 1
2017-09-14 08:05:02 118MB Standard oss://oss-odps-
test/log_data/year=2017/month=06/day=02/logfile
2017-09-14 08:06:45 123MB Standard oss://oss-odps-
test/log_data/year=2017/month=07/day=10/logfile
2017-09-14 08:07:11 115MB Standard oss://oss-odps-
test/log_data/year=2017/month=08/day=08/logfile
...
```

**Note:**

If you have uploaded offline data to OSS by using `osscmd` or other OSS tools, you can define the data path format. To ensure the partitioned external table feature operates normally, we recommend that the path where data is stored to is in the format specified in the previous example.

3. Then, you can execute the **ALTER TABLE ADD PARTITION** statement to import the partition information to MaxCompute. An example of the DDL statement is as follows:

```
ALTER TABLE log_table_external ADD PARTITION (year = '2017', month =
'06', day = '01')
ALTER TABLE log_table_external ADD PARTITION (year = '2017', month =
'06', day = '02')
ALTER TABLE log_table_external ADD PARTITION (year = '2017', month =
'07', day = '10')
ALTER TABLE log_table_external ADD PARTITION (year = '2017', month =
'08', day = '08')
```

...

4. When the data is ready and the partition information has been imported to MaxCompute, you can execute SQL statements to operate the partitions in the external table in OSS.

- Execute the following statement to count the number of unique IP addresses in the log dated June 1, 2017:

```
SELECT count(distinct(ip)) FROM log_table_external
WHERE year = '2017' AND month = '06' AND day = '01';
```



Note:

In the `log_table_external` directory that corresponds to an external table, only files in the `log_data/year=2017/month=06/day=01` subdirectory (logfile and To prevents unnecessary I/O operations, a full scan of the `log_data/` directory is not performed.

- Similarly, you can execute the following statement to analyze data for the second half of 2017:

```
SELECT count(distinct(ip)) FROM log_table_external
WHERE year = '2017' AND month > '06';
```



Note:

In this case, only the logs for the second half of 2017 stored in OSS are accessed.

1.14.2.1.5.3 Custom path of partition data in OSS

If you have historical data stored in OSS paths that are not in the `partitionKey1=value1\partitionKey2=value2\...` format, you can still access the data by using the MaxCompute partition feature. MaxCompute provides a way to import partitions through a custom path.

Example:

1. The data path only contains partition values (without partition keys). An example of the path layout is as follows:

```
osscmd ls oss://oss-odps-test/log_data_customized/
2017-09-14 08:03:35 128MB Standard oss://oss-odps-
test/log_data_customized/2017/06/01/logfile
2017-09-14 08:04:12 127MB Standard oss://oss-odps-
test/log_data_customized/2017/06/01/logfile. 1
2017-09-14 08:05:02 118MB Standard oss://oss-odps-
```



```
test/log_data_customized/2017/06/02/logfile
2017-09-14 08:06:45 123MB Standard oss://oss-odps-
test/log_data_customized/2017/07/10/logfile
2017-09-14 08:07:11 115MB Standard oss://oss-odps-
test/log_data_customized/2017/08/08/logfile
...
```

2. You can run the following statement to bind subdirectories to different partitions:

```
ALTER TABLE log_table_external ADD PARTITION (year = '2017', month =
'06', day = '01')
LOCATION 'oss://<ak_id>:<ak_key>@oss-cn-shanghai-internal.aliyuncs.
com/oss-odps-test/log_data_customized/2017/06/01/';
```



Note:

The ADD PARTITION and LOCATION clauses are specified in the preceding example to bind the partitions to data paths. Even if the data storage path is not in the partitionKey1=value1\partitionKey2=value2\... format, you can still access the partition data in the subdirectory.

1.14.2.1.5.4 Access fully-customized non-partitioned data subsets

In certain situations, you might need to access a file subset in an OSS path, but files in this subset do not have any obvious regularity in terms of directory layout. The unstructured data processing framework of MaxCompute is able to handle this situation, but will not be discussed in this topic.

If you require advanced operations such as this, contact the MaxCompute technical team for support.

1.14.2.1.6 Output OSS data

1.14.2.1.6.1 Create an external table

To write data to OSS, you need to run the CREATE EXTERNAL TABLE statement to create an external table first. The process is the same as that of reading data from OSS. After the external table is created, you can run MaxCompute SQL statements such as INSERT INTO and INSERT OVERWRITE to write data to OSS. In the following example, the built-in TsvStorageHandler is used.

```
DROP TABLE IF EXISTS tpch_lineitem_tsv_external;
CREATE EXTERNAL TABLE IF NOT EXISTS tpch_lineitem_tsv_external
(
  orderkey BIGINT,
  supkey BIGINT,
  discount DOUBLE,
  tax DOUBLE,
```

```
shipdate STRING,  
linestatus STRING,  
shipmode STRING,  
comment STRING  
)  
STORED BY 'com.aliyun.odps.TsvStorageHandler'  
LOCATION 'oss://<AK_id>:<AK_secret>@oss-cn-hangzhou-zmf.aliyuncs.com/  
oss-odps-test/tsv_output_folder/';
```

**Note:**

The preceding DDL statement creates an external table named `tpch_lineitem_tsv_external`, and associates two external data dimensions with this external table.

- **Data storage medium:** `LOCATION` associates an OSS address with the external table. This address will be used to read or write data from or to the external table.
- **Data storage format:** `StorageHandler` is used to define the data access mode. In this example, MaxCompute built-in `com.aliyun.odps.TsvStorageHandler` is used to read or write data from or to TSV files. You can also use the MaxCompute SDK to define `StorageHandlers`.

1.14.2.1.6.2 Write data to a TSV text file by using an INSERT statement on an external table

After you associate a file in OSS with an external table, you can run a standard SQL `INSERT OVERWRITE/INSERT INTO` statement on the external table to write data to the OSS file. The source data can be either data stored in a MaxCompute internal table or external data that is imported to MaxCompute through an external table.

**Note:**

- **MaxCompute internal table:** You can run an `INSERT` statement on an external table to write data from a MaxCompute internal table to an external storage medium.
- **External data imported to MaxCompute through an external table:** You can import external data to MaxCompute through an external table, use the data for computations, and then export the results to an external address or storage medium. For example, import Table Store data to MaxCompute and then export the data to OSS.

The following example assumes that you have a MaxCompute internal table named `tpch_lineitem` and want to export some of the data to OSS in the TSV format. After you create an external table, run the following `INSERT OVERWRITE` statements to export data:

```
INSERT OVERWRITE TABLE tpch_lineitem_tsv_external
SELECT l_orderkey, l_supkey, l_discount, l_tax, l_shipdate,
l_linestatus,
l_shipmode, l_comment
FROM tpch_lineitem
WHERE l_discount = 0.07 and l_tax = 0.01;
```

The preceding example selects eight columns from the rows in `tpch_lineitem` table that satisfy the conditions `l_discount = 0.07` and `l_tax = 0.01` and writes it to `tpch_lineitem_tsv_external` in OSS in the TSV format. After this operation is complete, you can view the corresponding TSV data file in OSS.

**Notice:**

Data exported from MaxCompute to OSS is stored in a special file structure.

- When you run `INSERT INTO/OVERWRITE` statements on an OSS address, all data is exported to the `.odps` folder at the specified `LOCATION`.
- The `.meta` file in the `.odps` folder is an extra macro data file written by MaxCompute to record valid data in the current folder. Typically, if the `INSERT` operation is successful, all the data in the current folder is valid. You are only required to parse the macro data if a job fails.
- If a job fails or is terminated, perform the `INSERT OVERWRITE` operation again until it is complete. This prevents parsing of the `.meta` file.
- If you need to parse the `.meta` file, contact Alibaba Cloud technical team for support.

The number of files that are generated during MaxCompute built-in TSV/CSV processing is equal to the number of concurrent SQL stages. You can use the flexible semantics and configurations of MaxCompute to limit the number of generated files. In the preceding example, if you need to force a TSV file to be generated, you can append `DISTRIBUTE BY l_discount` to the `INSERT OVERWRITE` operation. Then, a reduce stage with only one reducer is added at last so that only one TSV file is output.

1.14.2.1.6.3 Write data to an unstructured file by using an INSERT statement on an external table

MaxCompute also provides Outputter APIs for data output. You can use the APIs to write user data to a custom unstructured data file through OutputStream. Further details are not covered in this topic.

If you have this requirement, contact the MaxCompute technical team for support.

1.14.2.1.6.4 Migrate data between different storage media with MaxCompute

External tables act as an interface between MaxCompute and external storage media. External tables can be used to read or write data from or to various external storage media such as OSS and Table Store. Based on the external table feature, various data computing and storage links can be established. For example,

1. MaxCompute reads the OSS data associated with External Table A, and performs complicated computations. MaxCompute then outputs the results to the OSS address associated with External Table B.
2. MaxCompute reads the Table Store data associated with External Table A, and performs complicated computations. MaxCompute then outputs the results to the OSS address associated with External Table B.

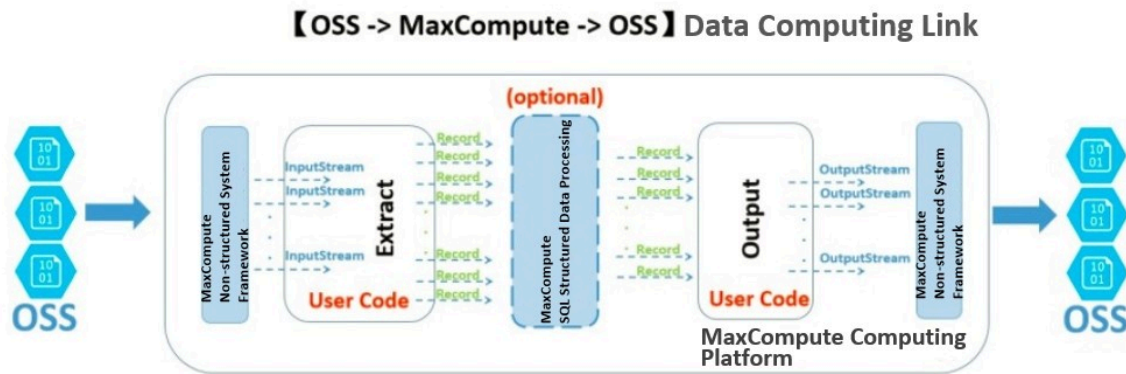


Note:

The preceding examples and data sources are scenarios with MaxCompute tables. The only difference is that the SELECT statement originates from an external table instead of a MaxCompute table.

Example:

By using MaxCompute as a central computing platform, you can import data from an OSS instance, and export that data to a different OSS instance (in a different location, or a different OSS account), as shown in the following figure.



From a data flow and processing logic standpoint, the unstructured data processing framework can be considered as a coupled data ingress and egress at both ends of the MaxCompute platform.

1. The external data (from an OSS instance) is converted based on the unstructured framework, and provided to the UDF API in the form of a Java InputStream class. The UDF extract logic is only required to read, parse, transform, and compute the data from the InputStream class, and return the data in the Record format used by MaxCompute.
2. Part of the returned records are used in the SQL logical operations on MaxCompute. These operations utilize the powerful SQL computation engine built into MaxCompute, and may generate new records.
3. The computed records are transferred to the UDF Output logic for further computation. Finally, the required information is extracted from the records, output through OutputStream, and written to the OSS instance.



Notice:

You can perform any combination of the preceding steps based on your needs.

1.14.2.2 Table Store data source

1.14.2.2.1 Preface

As the core computing component of the Alibaba Cloud big data platform, MaxCompute meets most distributed computing requirements within and outside Alibaba Group. As the entry of distributed data processing, MaxCompute SQL provides powerful support for quick processing and storage of large volumes (exabytes) of offline data. With the continuous expansion of big data business, many new data usage scenarios emerge. To adapt to the new scenarios,

the MaxCompute computing framework is constantly evolving. Its powerful computation capabilities, originally designed to process internal data in special formats, have expanded to process external data sources in various formats. This topic describes in detail how to import data from Table Store to MaxCompute, implementing seamless interoperability between data sources.

Compared with traditional databases, NoSQL KV Store supports flexible schema, scalability and real-time response for applications such as online service.

Alibaba Cloud Table Store is a large-scale NoSQL data storage service based on the Apsara system. It supports storage and real-time access of massive KV data. Table Store is widely used by all business units in Alibaba Group and the Alibaba Cloud ecosystem. In particular, Table Store features, such as row-level real-time update and override writing, are a supplement to the append-only operation of MaxCompute tables. As a storage-oriented service, Table Store does not provide sufficient computing capabilities to process large amounts of data concurrently. This makes it important to enable data interoperability between MaxCompute and Table Store.

The following examples show how to access and process Table Store data in MaxCompute.

1.14.2.2.2 MaxCompute reads and computes data in Table Store

1.14.2.2.2.1 Prerequisites and assumptions

This document assumes that you have basic knowledge of Table Store operations. If you are not familiar with Table Store or KV tables, we recommend that you first familiarize yourself with the basic concepts of Table Store (such as primary keys, partition keys, and attribute columns) before reading this topic.

1.14.2.2.2.2 Create an external table

External tables can act as interfaces between MaxCompute and Table Store: You can run a DDL statement (CREATE EXTERNAL TABLE) to import a table description in Table Store to the MaxCompute meta system. Then, you can process data in the Table Store table in the same way you process data in a MaxCompute table.

Example:

```
DROP TABLE IF EXISTS ots_table_external;  
CREATE EXTERNAL TABLE IF NOT EXISTS ots_table_external
```

```
(
odps_orderkey bigint,
odps_orderdate string,
odps_custkey bigint,
odps_orderstatus string,
odps_totalprice double
)
STORED BY 'com.aliyun.odps.TableStoreStorageHandler'
-- com.aliyun.odps.TableStoreStorageHandler is a MaxCompute built-
in StorageHandler for processing Table Store data. It defines the
interaction between MaxCompute and Table Store. The relevant logic is
implemented by MaxCompute.
WITH SERDEPROPERTIES (
-- SERDEPROPERTIES is an API that provides parameter options. Two
options must be specified for TableStoreStorageHandler: tablestore.
columns.mapping and tablestore.table.name.
'tablestore.columns.mapping'=':o_orderkey, :o_orderdate, o_custkey,
o_orderstatus,o_totalprice',
-- tablestore.columns.mapping: This option is required. It describes
the columns of Table Store tables that are accessed by MaxCompute,
including primary key and attribute columns. Column names starting
with a colon (:) are primary key columns in Table Store tables. In
this example, :o_orderkey and :o_orderdate are primary key columns.
The other column names specified are attribute columns. Table Store
supports up to four primary keys of the bigint or string type. The
first primary key is the partition key. When you specify a mapping,
you must provide all primary key columns of the specified Table Store
table. You do not have to specify all the attribute columns, only
those accessed by MaxCompute.
'tablestore.table.name'='ots_tpch_orders'
-- tablestore.table.name: This option is required. It describes the
names of Table Store tables that are accessed by MaxCompute. If you
specify an invalid (nonexistent) Table Store table name, an error is
returned and MaxCompute does not create a Table Store table with this
name.
)
LOCATION 'tablestore://<your AK id>:<your AK secret key>@odps-ots-
dev.cn-
hangzhou.ots.aliyuncs.com';
-- The LOCATION clause specifies the Table Store information,
including the instance name and endpoint.
```

**Note:**

The preceding example maps a Table Store table to a MaxCompute external table. The subsequent operations on the Table Store table can be performed through the external table.

1.14.2.2.2.3 Access Table Store data through an external table

After you follow the preceding example to create an external table, Table Store data is imported to MaxCompute. Then, you can access Table Store data by using MaxCompute SQL statements.

Example:

```
SELECT odps_orderkey, odps_orderdate, SUM(odps_totalprice) AS  
totalprice  
FROM ots_table_external  
WHERE odps_orderkey > 5000 AND odps_orderdate >20170725 AND odps_order  
date <20170910  
GROUP BY odps_orderkey, odps_orderdate  
HAVING totalprice> 2000;
```

**Note:**

This example uses common MaxCompute SQL statements. Table Store access details are processed internally by MaxCompute.

If you need to use a copy of data for multiple computations, you can import the data from Table Store to a MaxCompute table (internal). This is more efficient than reading the data from Table Store every time.

Example:

```
CREATE TABLE internal_orders AS  
SELECT odps_orderkey, odps_orderdate, odps_custkey, odps_totalprice  
FROM ots_table_external  
WHERE odps_orderkey > 5000 ;
```

**Note:**

internal_orders is a common MaxCompute table that has all the features of a MaxCompute internal table. These features include efficient compressed column storage and complete meta. This table is stored in MaxCompute, so it can be accessed faster than an external table in Table Store. This feature is particularly suitable for hot data that is used for multiple computations.

1.14.2.2.3 Write data from MaxCompute to Table Store

Data interaction between MaxCompute and Table Store includes importing data from Table Store to MaxCompute for batch processing and exporting the data processing results from MaxCompute to Table Store. Table Store features such as real-time update and single-line overwrite allow you to quickly upload offline computing results to online applications. You can use MaxCompute SQL INSERT OVERWRITE statements to export data to Table Store.

**Note:**

MaxCompute does not create external tables in Table Store. Before exporting data to a table in Table Store, make sure that the table has already been created in Table Store. Otherwise, an error is reported.

The following example assumes that you have created an external table named `ots_table_external` in MaxCompute. There is a table named `ots_tpch_orders` in Table Store. There is an internal table named `internal_orders` in MaxCompute. You want to process data in `internal_orders` before writing it to Table Store. For this purpose, you can run the following **INSERT OVERWRITE TABLE** statement on the external table.

```
INSERT OVERWRITE TABLE ots_table_external
SELECT odps_orderkey, odps_orderdate, odps_custkey, CONCAT(odps_custk
ey,
'SHIPPED'), CEIL(odps_totalprice)
FROM internal_orders;
```



Note:

Table Store is a NoSQL storage medium that stores KV data. Data output from MaxCompute affects only the rows that contain the corresponding primary keys in the Table Store table. In addition, only the attribute columns in the external table was created are updated. The columns that are not included in the external table are not modified.

1.14.2.3 AnalyticDB data source

1.14.2.3.1 Overview

AnalyticDB updates or processes data. If both the data processed by AnalyticDB and the data in MaxCompute are used for computation, the data from AnalyticDB must be synchronized with the data from MaxCompute. To accomplish this, you can create an external table to access the AnalyticDB data.

The following example shows how MaxCompute accesses and processes AnalyticDB data.

1.14.2.3.2 Write data to AnalyticDB

1.14.2.3.2.1 Create an external table

Run the following command to create an external table:

```
set odps.sql.hive.compatible=true;
drop table if exists ads_table_external;
```

```
CREATE EXTERNAL TABLE if not exists ads_table_external
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
  c_bigint bigint ,
  c_double double ,
  c_float float ,
  --c_time datetime ,
  c_date date ,
  c_timestamp datetime ,
  c_string string
)
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'
location 'jdbc:mysql:host:port/databasename? useSSL=false&user=${user}
&password=${password}&table=${tablename}'
TBLPROPERTIES(
  'mcfed.mapreduce.jdbc.input.orderby'='c_int'
)
;
```

**Note:**

The preceding command is for reference only.

1.14.2.3.2.2 Write and query data

After an external table is created, you can use it in the same way you would use a MaxCompute table. You can execute the **INSERT OVERWRITE | INTO** and **SELECT** statements to write data and query whether the write operation is successful respectively. For more information about the statements, see *DML statements* in *MaxCompute SQL*.

1.14.2.3.3 Read data from AnalyticDB

Run the following commands to read data from AnalyticDB:

```
set odps.sql.hive.compatible=true;
drop table if exists ads_read_external;
CREATE EXTERNAL TABLE if not exists ads_read_external
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
  c_bigint bigint ,
  c_double double ,
  c_float float ,
  --c_time datetime ,
  c_date date ,
  c_timestamp datetime ,
  c_string string
)
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'
location 'jdbc:mysql:host:port/databasename? useSSL=false&user=${user}
&password=${password}&table=${tablename}'
```

```
TBLPROPERTIES(  
  'mcfed.mapreduce.jdbc.input.orderby'='c_int'  
)  
;  
-- Create an external table.  
select * from ads_read;  
-- Query and read data.
```

**Note:**

The preceding commands are for reference only.

1.14.2.4 RDS data source

1.14.2.4.1 Overview

RDS updates or processes data. If both the data processed by RDS and the data in MaxCompute is used in computation, the data in RDS must be synchronized to MaxCompute. In this case, you can access the data in RDS by creating an external table.

The following examples show how MaxCompute accesses and processes RDS data.

**Note:**

When you create an external table, the corresponding table may not exist in RDS. However, when you perform the SELECT or INSERT operation on external tables, you must create the corresponding tables in RDS first.

1.14.2.4.2 Write data to RDS

1.14.2.4.2.1 Create an external table

Run the following command to create an external table:

```
set odps.sql.hive.compatible=true;  
drop table if exists rds_table_external;  
CREATE EXTERNAL TABLE if not exists rds_table_external  
(  
  id bigint,  
  name string,  
  age tinyint  
)  
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'  
location 'jdbc:mysql:host:port/databasename? useSSL=false&user=${user}  
&password=${password}&table=${tablename}'  
TBLPROPERTIES(  
  'mcfed.mapreduce.jdbc.input.orderby'='c_int'  
)
```

;

**Note:**

The preceding command is for reference only.

1.14.2.4.2.2 Write and query data

After an external table is created, you can use it in the same way you use a MaxCompute table. You can execute the **INSERT OVERWRITE | INTO** and **SELECT** statements respectively to write data and query whether the write operation is successful. For more information about the **INSERT OVERWRITE | INTO** and **SELECT** statements, see *DML statements in MaxCompute SQL*.

1.14.2.4.3 Read data from RDS

Run the following commands to read data from RDS:

```

set odps.sql.hive.compatible=true;
drop table if exists rds_read_external;
CREATE EXTERNAL TABLE if not exists rds_read_external
(
  id int,
  name string,
  age int
)
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'
location 'jdbc:mysql:host:port/databasename? useSSL=false&user=${user}
&password=${password}&table=${tablename}'
TBLPROPERTIES(
  'mcfed.mapreduce.jdbc.input.orderby'='c_int'
)
;
-- Create an external table.
select * from rds_read;
-- Query and read data.

```

**Note:**

The preceding commands are for reference only.

1.14.2.5 HDFS data source (Alibaba Cloud)

1.14.2.5.1 Overview

Alibaba Cloud Hadoop Distributed File System (HDFS) is a distributed file system designed for Alibaba Cloud computing resources such as ECS and Container Service.

HDFS allows you to manage and access data in the same way as its open-source counterpart. HDFS features such as unlimited capacity, performance scale-out,

single namespace, multi-tenancy, high reliability, and high availability, can be used without the need to modify existing big data analysis applications.

MaxCompute can interact with HDFS to jointly compute external tables.

HDFS supports multiple file formats, such as text file, sequence file, RC file, Parquet, and AVRO. The following example use text file to show how MaxCompute accesses and processes HDFS data.

1.14.2.5.2 Data processing for common tables

1.14.2.5.2.1 Write data to HDFS

1.14.2.5.2.1.1 Create an external table

Run the following command to create an external table:

```
set odps.sql.hive.compatible=true;
drop table if exists textfile_external;
CREATE external TABLE if not exists textfile_external
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
  c_bigint bigint ,
  c_double double ,
  c_float float ,
  c_date date ,
  c_timestamp datetime ,
  c_string string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim'=',')
STORED AS textfile
location "mcfed:dfs://host:port/user/textfile"
-- host must be set as MountPointId.
-- /user/textfile is the file path. Replace it with the actual file
path.
TBLPROPERTIES(
  "mcfed.fs.dfs.impl"="com.alibaba.dfs.DistributedFileSystem"
);
```



Note:

The preceding command is for reference only.

1.14.2.5.2.1.2 Write and query data

After an external table is created, you can use it in the same way you use a MaxCompute table. You can execute the INSERT OVERWRITE | INTO and SELECT statements respectively to write data and query whether the write operation

is successful. For more information about the **INSERT OVERWRITE | INTO** and **SELECT** statements, see *DML statements in MaxCompute SQL*.

1.14.2.5.2 Read data from HDFS

Run the following command to read data from HDFS after you upload the textfile file to HDFS:

```
set odps.sql.hive.compatible=true;
drop table if exists textfile_external_read;
CREATE external TABLE if not exists textfile_external_read
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
  c_bigint bigint ,
  c_double double ,
  c_float float ,
  c_date date ,
  c_timestamp datetime ,
  c_string string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim'=',')
STORED AS textfile
location "mcfed:dfs://host:port/user/textfile"
-- host must be set as MountPointId.
-- /user/textfile is the file path. Replace it with the actual file
path.
TBLPROPERTIES(
  "mcfed.fs.dfs.impl"="com.alibaba.dfs.DistributedFileSystem"
);
-- Create an external table.
select * from textfile_external_read;
select count(*) from textfile_external_read;
select a.c_int,a.c_boolean,a.c_string,b.value from textfile_e
xternal_read a join dfstest b on a.c_int=b.id;
-- Query and read data.
```



Note:

The preceding command is for reference only.

1.14.2.5.3 Data processing for partitioned tables

Run the following commands to create an external table and process its data:

```
set odps.sql.hive.compatible=true;
drop table if exists textfile_partition;
CREATE external TABLE if not exists textfile_partition
(
  id string,
  name string
)
partitioned by (date string)
```

```

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
,
WITH SERDEPROPERTIES ('field.delim'=',')
STORED AS textfile
location "mcfed:dfs://host:port/user/partition/textfile/"
-- host must be set as MountPointId.
-- user/partition/textfile/ is the file path. Replace it with the
actual file path.
TBLPROPERTIES(
    "mcfed.fs.dfs.impl"="com.alibaba.dfs.DistributedFileSystem"
);
-- Create an external table.
alter table textfile_partition add partition (date='20190218');
alter table textfile_partition add partition (date='20190219');
-- Add partitions.
insert into table textfile_partition partition(date='20190218')select
'1','cd' from (select count(*) from textfile_partition)a;
insert into table textfile_partition partition(date='20190219')select
'2','gh' from (select count(*) from textfile_partition)a;
-- Write data to HDFS.
select * from textfile_partition;
select count(*) from textfile_partition;
select a.id,a.name,b.value from textfile_partition a join dfstest b on
a.id=b.id;
-- Query and read data.

```

**Note:**

The preceding commands are for reference only.

1.14.2.6 TDDL data source

1.14.2.6.1 Overview

By encapsulating MySQL, TDDL provides features such as data partitioning, read/write splitting, and failover. In most cases, TDDL can be directly used to access MySQL databases. TDDL also provides Corona connection mode. Corona is a MySQL proxy that follows the standard MySQL protocol and can use JDBC to establish a connection.

MaxCompute can access MySQL databases of TDDL. Built-in StorageHandlers encapsulate native APIs provided by Hadoop such as org, Apache, Hadoop, MapReduce, lib, and db. MySQL JDBC is used for underlying data communication.

The following topics describe how MaxCompute accesses and processes TDDL data.

**Notice:**

Among create, read, update, and delete (CRUD) operations, only the following operations are supported:

- MaxCompute reads data from the external table created for a MySQL database.

- MaxCompute writes data to the external table in the append mode.

1.14.2.6.2 Prerequisites

Because many features are disabled in the MaxCompute 2.0 by default, you must manually configure the following settings:

```
set odps.sql.hive.compatible=true;
-- You must configure this item for all DDL and DML statements to be
used on TDDL external tables.
```

```
set odps.sql.udf.java.retain.legacy=false;
-- You must configure this item for all DDL and DML statements to be
used on TDDL external tables.
```

```
set odps.sql.jdbc.splits.num=3;
-- Set the number of splits that MaxCompute reads from the MySQL
database. Maximum value: 256. Default value: 1. You must configure
this item for the SELECT operation on TDDL external tables.
```

```
set odps.sql.jdbc.reducer.num=3;
-- Set the number of concurrent instances that MaxCompute writes to
the MySQL database. Maximum value: 256. Default value: 64. If the
number of concurrent instances in the generated execution plan is
smaller than this value, no changes are made. You must configure this
item for the INSERT operation on TDDL external tables.
```

```
set odps.sql.hive.compatible=true;
-- Use an open-source community API to obtain and parse MySQL data
types. You must configure this item for all DDL and DML statements to
be used on TDDL external tables.
```

```
set odps.sql.type.system.odps2=true;
-- You must configure this item if new data types TINYINT, SMALLINT
, INT, FLOAT, VARCHAR, TIMESTAMP, and BINARY are involved in SQL
operations such as CREATE, SELECT, and INSERT.
```

1.14.2.6.3 Create a TDDL external table

1.14.2.6.3.1 Syntax

External tables can act as interfaces between MaxCompute and databases. The method used to process MySQL unstructured data in TDDL is similar to the method to access and process OSS unstructured data. First, you must execute the **CREATE EXTERNAL TABLE** statement to create an external table. The syntax is as follows:

```
-- Remember to add the corresponding SET statement.
DROP TABLE [IF EXISTS] <external_table_name>;
CREATE EXTERNAL TABLE [IF NOT EXISTS] <external_table_name>
(<column schemas>)
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'
location 'jdbc:mysql://path_format'
TBLPROPERTIES(
```



```
); ...
```

Description:

- **column schema:** supports the following data types.

Table 1-52: Data type mapping

MySQL type	MaxCompute type
TINYINT (unsigned)	TINYINT
SMALLINT (unsigned)	SMALLINT
INT (unsigned)	INT
BIGINT (unsigned)	BIGINT
BOOLEAN	BOOLEAN
FLOAT	FLOAT
DOUBLE	DOUBLE
VARCHAR	VARCHAR
TEXT	STRING
DATE	DATE
DATETIME	DATETIME
DECIMAL	DECIMAL (x, y) (The default precision is (10, 0). An error is returned when overflow occurs.)

**Notice:**

Because unsigned data types are not supported in MaxCompute, loss of precision may occur if unsigned types are specified.

- `setproject odps.sql.udf.strict.mode=true;` (strict mode, which is the default mode).

- **When reading external tables:** MaxCompute can read the data if unsigned data is converted to signed data without loss of precision. A `RuntimeException` ("value out of range") error is reported if loss of precision occurs during data type conversion.
- **When writing external tables:** MaxCompute does not check data types. You can specify the SQL mode to let MySQL produce desired data check actions.

For more information about SQL mode settings and data check actions, see [Server SQL Modes](#).

- `setproject odps.sql.udf.strict.mode=false;` (non-strict mode)

■ When reading external tables: MaxCompute can read unsigned data that has been converted to signed data without loss of precision. NULL is obtained if loss of precision occurs during data type conversion.

■ When writing external tables: MaxCompute does not check data types. You can specify the SQL mode to let MySQL produce desired data check actions.

- **STORED BY:** Only built-in StorageHandlers are supported. TDDL table field types must be within the range of supported data types in column schema.
- **LOCATION:** Three LOCATION formats are supported.

1. Access a MySQL database through a JDBC connection string.

```
jdbc:mysql://<user>:<password>@<host>/<databaseName>? useSSL=false&
table=<tableName>
```

user and password are the username and password of the JDBC connection string. host is the network address of the MySQL database. databaseName is the name of the MySQL database. tableName is the name of the MySQL table corresponding to the external table.

2. Access the MySQL database of TDDL through Corona.

```
jdbc:mysql://<user>:<password>@<host>/<databaseName>? useSSL=false&
table=<tableName>
```

3. Access the MySQL database of TDDL through an application name.

```
jdbc:mysql://dummy_host? table=<tableName>
```

tableName is the name of the MySQL table corresponding to the external table. You must specify `odps.federation.jdbc.tddl.appname` in the TBLPROPERTIES clause.



Notice:

In the first location format, MaxCompute interacts with the database through JDBC. You must enter your username and password as plaintext data, which makes this location format less secure than others. Although usernames and passwords will be hidden when LogView or DESC EXTENDED TABLE is used in

MaxCompute, we recommend that you use a separate DDL statement to create an external table before using the external table.

For example, a project member with higher permissions can create an external table in MaxCompute. Other project members can then directly use the external table. This prevents project members with lower permissions from using the plaintext username and password, and prevents the plaintext password from being contained in SQL scripts.

- **TBLPROPERTIES:** includes the following items.
 - **odps.federation.jdbc.condition:** specifies the filter when MaxCompute reads data from a MySQL database. The difference between `odps.federation.jdbc.condition` and `select * from text_test_jdbc_write_external where condition`:

Suppose the MySQL table contains 100 rows of data and you want to filter the data such that you obtain 10 rows. When you execute `odps.federation.jdbc.condition`, the MySQL table is filtered and MaxCompute only reads 10 rows from the external table. When you execute `select * from text_test_jdbc_write_external where condition`, MaxCompute reads 100 rows from the MySQL table, and then obtains 10 rows.

- **odps.federation.jdbc.colmapping:** specified column name mapping. Example:

```
-- mysql schema: mysqlId int
-- MaxCompute create table
CREATE EXTERNAL TABLE if not exists table_external
(
  odpsId1 int,
  odpsId2 int
)
STORED BY ...
location ...
TBLPROPERTIES('odps.federation.jdbc.colmapping'='odpsId1:mysqlId
, odpsId2:mysqlId');
```

- **odps.federation.jdbc.insert.type:** specifies the insertion type when data is written into the MySQL database. The following data insertion types are supported: `simpleInsert`, `insertOnDuplicateKeyUpdate`, and `replaceInto`. By default, the insertion type is `simpleInsert` if this parameter is not specified.

The INSERT statement executed in MaxCompute is parsed into the following SQL statements to update the database:

```
insert into sqlTable xxx values xxx;
```

```
insert into sqlTable xxx values xxx on duplicate key update col1
=values(col1), col2=values(col2);
replace into sqlTable xxx values xxx;
```

- **odps.federation.jdbc.tddl.app.access.key**: the AccessKey ID for the authorized application.
- **odps.federation.jdbc.tddl.app.secret.key**: the AccessKey Secret for the authorized application.
- **odps.federation.jdbc.tddl.appname**: the application name of TDDL. Note that if you specify this value, MaxCompute uses the application name to access the MySQL database in TDDL SDK mode.

1.14.2.6.3.2 Example

The following example shows how to use the application name to access the MySQL database of TDDL. In this example, the application name is ODPS_TDDL_TEST_APP and the table name is odps_federation_localrun_write.

Example:

```
-- Remember to add the corresponding SET statement.
drop table if exists text_test_jdbc_external;
CREATE EXTERNAL TABLE if not exists text_test_jdbc_external
(
  colmapping tinyint, --c_tinyint tinyint,
  c_smallint smallint,
  c_int int,
  c_bigint bigint,
  c_utinyint tinyint,
  c_usmallint smallint,
  c_uint int,
  c_ubigint bigint,
  c_boolean tinyint,
  --c_float float, -- in tddl, not recommend float and double type as
it may lost precision
  --c_double double,
  c_string string,
  c_datetime datetime,
  c_decimal decimal
)
STORED BY 'com.aliyun.odps.jdbc.JdbcStorageHandler'
location 'jdbc:mysql://dummy_host? table=odps_federation_loca
lrun_write'
TBLPROPERTIES(
'odps.federation.jdbc.insert.type'='simpleInsert',
'odps.federation.jdbc.condition'='c_boolean = 1 and c_int is not null
and c_utinyint=127',
'odps.federation.jdbc.colmapping'='colmapping:c_tinyint',
'odps.federation.jdbc.tddl.appname'='ODPS_TDDL_TEST_APP',
'odps.federation.jdbc.tddl.app.access.key'='your tddl app access key',
```

```
'odps.federation.jdbc.tddl.app.secret.key'='your tddl app secret key
');
```

1.14.2.6.4 Read data from an external table

For complex operations such as GROUP JOIN, we recommend that you import data from external table to MaxCompute tables before performing operations. This improves the efficiency of data computation. The following example shows how to import data from an associated MySQL external table to MaxCompute.

Create a MaxCompute table

Example:

```
CREATE TABLE if not exists text_test_jdbc_max_compute
(
  c_tinyint tinyint,
  c_smallint smallint,
  c_int int,
  c_bigint bigint,
  c_tinyint tinyint,
  c_smallint smallint,
  c_uint int,
  c_ubigint bigint,
  c_boolean tinyint,
  --c_float float,
  --c_double double,
  c_string string,
  c_datetime datetime,
  c_decimal decimal
);
```

Import data to a MaxCompute table

Example:

```
-- Remember to add the SET statement.
insert OVERWRITE TABLE text_test_jdbc_odps select * from text_test_
jdbc_read_external;
```

Relationship between creating an external table and importing data to a MaxCompute table

When you create an external table, only a data channel is established between MaxCompute and MySQL. MaxCompute does not store any MySQL data. If external table data is lost from the MySQL database, it will not be available in MaxCompute.

When data is imported to a MaxCompute table, the data is actually stored in the MySQL database. If imported data is lost from the MySQL database, it can be retrieved from the MaxCompute table.

1.14.2.6.5 Write data to an external table in the append mode

The column names and data types of the external table must be consistent with those of the database to ensure that the correct data is written to the external table. For more information about data check actions when loss of precision occurs during data type conversion, see the column schema parameter of [Syntax](#).

An example of the command used is as follows:

```
-- Remember to add the SET statement.  
insert INTO TABLE text_test_jdbc_external select * from text_test_  
jdbc_max_compute;
```



Note:

For MySQL external tables, `insert INTO mysql-external-table` uses the same syntax as `insert OVERWRITE mysql-external-table`. No matter which statement is executed, data is appended to the table and you can use `ODPS.federation.jdbc.insert` type to specify the data insertion type. For more information, see the `TBLPROPERTIES` parameter in [Syntax](#). However, the preceding syntax notes are not applicable to MaxCompute tables.

1.14.3 External data sources

1.14.3.1 HDFS data source (open-source)

1.14.3.1.1 Overview

HDFS is the most widely used storage service in the open-source community. Most customers use HDFS at the underlying layer of their self-developed big data systems.

MaxCompute uses external tables to access HDFS data to facilitate data migration, interact with self-developed customer systems, and reduce the efforts and costs of customers.

HDFS supports multiple file formats, such as text file, sequence file, RC file, Parquet, and AVRO. The following example use text file to show how MaxCompute accesses and processes HDFS data.

1.14.3.1.2 Write data to HDFS

1.14.3.1.2.1 Create an external table

Run the following command to create an external table after the testfile script has been compiled:

```
set odps.sql.hive.compatible=true;
drop table if exists textfiletest;
CREATE EXTERNAL TABLE if not exists textfiletest
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
  c_bigint bigint ,
  c_double double ,
  c_float float ,
  --c_time datetime ,
  c_date date ,
  c_timestamp datetime ,
  c_string string
)
STORED as TEXTFILE
location 'hdfs://host:port/user/wbyy/';
-- File path /user/wbyy/ is for reference only. Replace it with the
path to actually be accessed.
```



Note:

The preceding command is for reference only.

1.14.3.1.2.2 Write and query data

After an external table is created, you can use it in the same way you would use a MaxCompute table. You can execute the INSERT OVERWRITE | INTO and SELECT statements to write data and check whether the write operation is successful respectively. For more information about the statements, see *DML statements* in *MaxCompute SQL*.

1.14.3.1.3 Read data from HDFS

Run the following command to read data from HDFS after you compile the testfile script:

```
set odps.sql.hive.compatible=true;
drop table if exists testfile_read;
CREATE EXTERNAL TABLE if not exists testfile_read
(
  c_int int ,
  c_tinyint tinyint ,
  c_boolean boolean ,
  c_smallint smallint ,
```

```
c_biging bigint ,
c_double double ,
c_float float ,
--c_time datetime ,
c_date date ,
c_timestamp datetime ,
c_string string
)
STORED as TEXTFILE
location 'hdfs://host:port/user/wbyy/';
-- File path /user/wbyy/ is for reference only. Replace it with the
path to actually be accessed.
-- Create an external table.
select * from testfile_read;
-- Query and read data.
```

**Note:**

The preceding command is for reference only.

1.14.3.2 MongoDB data source

1.14.3.2.1 Overview

ApsaraDB for MongoDB is a stable, reliable, and auto-scaling database service that is fully compatible with MongoDB protocols. MongoDB offers a full range of database solutions, such as disaster recovery, backup, restoration, monitoring, and alerting.

MaxCompute can interact with MongoDB for joint computation after you create external tables.

The following examples show how MaxCompute accesses and processes MongoDB data.

1.14.3.2.2 Prerequisites

You must first deploy MongoDB before creating an external table and processing MongoDB data.

1. Run the following command to enable the MongoDB service:

```
bin/mongod --dbpath=./db
```

2. Run the following command to start the MongoDB client:

```
bin/mongo --host=${host}
```

3. Run the following command to create a database:

```
use mongodb
```

4. Run the following command to create a username and password:

```
db.createUser({user: '${user}', pwd: '${password}', roles: [{role: 'readWrite', db: 'mongodb'}]})
```

5. Run the following command to check whether the operation is successful. A response of 1 indicates a successful operation.

```
db.auth('${user}', '${password}')
```

1.14.3.2.3 Write data to MongoDB

1.14.3.2.3.1 Create an external table

Run the following command to create a collection in MongoDB:

```
db.createCollection("${tablename}", { capped : true, autoIndexId : true, size : 6142800, max : 10000 } )  
-- The values of the size and max parameters are for reference only.  
Replace them with the values to actually be used.
```

After the collection has been created, run the following command to create an external table:

```
set odps.sql.hive.compatible=true;  
drop table if exists mongo_table_external;  
CREATE external TABLE if not exists mongo_table_external  
(  
    id string,  
    name string  
)  
STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'  
location "mcfed:mongodb://${user}:${password}@host:port/mongodb.${tablename}"  
TBLPROPERTIES(  
    "mcfed.mongo.input.split_size"="2",  
    -- input.split_size value is for reference only. Replace them with the values to actually be used.  
    "mcfed.location"="mongodb://${user}:${password}@host:port/mongodb.${tablename}",  
    "mcfed.mongo.input.uri"="mongodb://${user}:${password}@host:port/mongodb.${tablename}",  
    "mcfed.mongo.output.uri"="mongodb://${user}:${password}@host:port/mongodb.${tablename}"
```

);

**Note:**

The preceding commands are for reference only.

1.14.3.2.3.2 Write and query data

After an external table is created, you can use it in the same way that you would use a MaxCompute table. You can execute the **INSERT OVERWRITE | INTO** and **SELECT** statements to write data and check whether the write operation is successful respectively. For more information about the statements, see *DML statements in MaxCompute SQL*.

1.14.3.2.4 Read data from MongoDB

Run the following command to read data from MongoDB after a row of data has been inserted into a created collection:

```

set odps.sql.hive.compatible=true;
drop table if exists mongo_read_external;
CREATE external TABLE if not mongo_read_external
(
    id string,
    name string
)
STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'
location "mcfed:mongodb://${user}:${password}@host:port/mongodb.${
tablename}"
TBLPROPERTIES(
    "mcfed.mongo.input.split_size"="2",
    -- The value of the input.split_size parameter is for reference only.
    Replace it with the value to actually be used.
    "mcfed.location"="mongodb://${user}:${password}@host:port/
mongodb.${tablename}",
    "mcfed.mongo.input.uri"="mongodb://${user}:${password}@host:
port/mongodb.${tablename}"
);
-- Create an external table.
select * from mongo_external;
-- Query and read data.

```

**Note:**

The preceding command is for reference only.

1.14.3.3 HBase data source

1.14.3.3.1 Overview

ApsaraDB for HBase is a distributed database based on Hadoop. It can store PBs of data and be used in scenarios requiring high-throughput random read/writes.

MaxCompute can interact with HBase for joint computation after you create external tables.

The following examples show how MaxCompute accesses and processes HBase data .

1.14.3.3.2 Write data to HBase

1.14.3.3.2.1 Create an external table

Run the following command to create an external table:

```
set odps.sql.hive.compatible=true;
drop table if exists hbase_table_external;
CREATE EXTERNAL TABLE if not exists hbase_table_external
(
  id string,
  cfa string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ('mcfed.hbase.table.name'='${table.name}', 'mcfed.
hbase.columns.mapping'=':key,cf:a')
-- cf is the column family in the HBase table.
location 'hbase://host:port'
TBLPROPERTIES('hbase.table.name'='${table.name}', 'hbase.columns.
mapping'=':key,cf:a', 'mcfed.zookeeper.session.timeout'='30', 'mcfed
.hbase.client.retries.number'='1', "mcfed.hbase.zookeeper.quorum"="${
host}", "mcfed.hbase.zookeeper.property.clientPort"="${port}");
-- The values of the zookeeper.session.timeout and hbase.client.
retries.number parameters are for reference only. Replace them with
the values to actually be used.
```



Note:

The preceding command is for reference only.

1.14.3.3.2.2 Write and query data

After an external table is created, you can use it in the same way that you would use a MaxCompute table. You can execute the INSERT OVERWRITE | INTO and SELECT statements to write data and check whether the write operation is successful respectively. For more information about the statements, see *DML statements in MaxCompute SQL*.

1.14.3.3.3 Read data from HBase

Run the following commands to read data from HBase after you have created a table in the HBase client and inserted data into it:

```
set odps.sql.hive.compatible=true;
drop table if exists hbase_read_external;
```

```
CREATE EXTERNAL TABLE if not exists hbase_read_external
(
  id string,
  name string,
  a string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ('mcfed.hbase.table.name'='${table.name}','mcfed.
hbase.columns.mapping'=':key,f1:name,f1:a')
-- f1 is the column family in the HBase table.
location 'hbase://host:port'
TBLPROPERTIES('hbase.table.name'='${table.name}','hbase.columns.
mapping'=':key,f1:name,f1:a', 'mcfed.zookeeper.session.timeout'='30
', 'mcfed.hbase.client.retries.number'='1', "mcfed.hbase.zookeeper.
quorum"="${host}", "mcfed.hbase.zookeeper.property.clientPort"="${port
}");
-- The values of the zookeeper.session.timeout and hbase.client.
retries.number parameters are for reference only. Replace them with
the values to actually be used.
-- Create an external table.
select * from hbase_read_external;
select count(*) from hbase_read_external;
select a.id,a.name from hbase_read_external a join hbase_test b on a.
id=b.id;
-- Query and read data.
```

**Note:**

The preceding commands are for reference only.

1.15 Unstructured data access and processing (inside MaxCompute)

1.15.1 Overview

MaxCompute has the following problems when processing unstructured data:

MaxCompute stores data as volumes and must export generated unstructured data to an external system for processing.

To alleviate these problems, MaxCompute uses external tables to enable connections between MaxCompute and various data types. MaxCompute uses external tables to read and write data volumes as well as process unstructured data from external sources such as OSS.

The following topics describe how MaxCompute accesses and processes volume unstructured data through external tables.

1.15.2 Create a volume external table

1.15.2.1 Syntax

You must execute the **CREATE EXTERNAL TABLE** statement to create an external table.

```
DROP TABLE [IF EXISTS] <external_table_name>;
CREATE EXTERNAL TABLE [IF NOT EXISTS] <external_table_name>
(<column schemas>)
[PARTITIONED BY (partition column schemas)]
STORED BY '<StorageHandler>'
[WITH SERDEPROPERTIES (
    'name'='value'
)]
LOCATION 'volume://...'
[USING '<Resourcename>']
;
```

Description:

- **STORED BY:** Two built-in `StorageHandlers` `com.aliyun.odps.CsvStorageHandler` and `com.aliyun.odps.TsvStorageHandler` are supported. They can be used to read and write CSV files where the column delimiter is a comma and the row delimiter is `\n` or TSV files where the column delimiter is `\t` and the row delimiter is `\n`. If the built-in `StorageHandlers` cannot be used for some reason, you can build a custom `StorageHandler`.
- **WITH SERDEPROPERTIES:** specifies table attributes such as delimiters for a custom `StorageHandler`.
- **LOCATION:** the location format of the table.

Format:

```
volume://[project_name]/volume_name/partition_value
```

Example:

```
volume://test_project/volume_data/20190102
```

`project_name` is optional. If `project_name` is not specified, the current project is used to obtain volume data after the DML SQL statement is executed. In the

preceding example, if the current project is myproject, you can use the following location format:

```
volume:///volume_data/20190102
```



Notice:

- The location of a non-partitioned table must point to a volume partition, instead of the volume itself.
- The location of a partitioned table must point to the volume itself.
- The volume path cannot contain an equal sign (=) and does not support the default standard partition path `ds=2017071` that is used when a partition is created. The partition path must be customized. The custom partition path can be any path supported by the volume. For example, if the partition path is 20190102, the path combined with volume path can be `volume:///test_project/volume_data/20190102`.

- **USING:** specifies the StorageHandler resource. To use a custom StorageHandler, you must first export the custom StorageHandler as a JAR package and then add it to MaxCompute as a JAR resource.

1.15.2.2 Use the built-in StorageHandler to create an external table

You can use the built-in StorageHandler to create a partitioned or non-partitioned table.

Create a non-partitioned table

Example:

```
DROP TABLE IF EXISTS volume_ext;
CREATE EXTERNAL TABLE volume_ext
(
  key string,
  value string
)
STORED BY 'com.aliyun.odps.CsvStorageHandler'--The built-in StorageHandler.
LOCATION 'volume:///test_project/volume_data/20190102'
;
```

Create a partitioned table

Example:

```
DROP TABLE IF EXISTS volume_ext_pt;
```

```
CREATE EXTERNAL TABLE volume_ext_pt
(
  key string,
  value string
)
PARTITIONED BY (ds string)
STORED BY 'com.aliyun.odps.CsvStorageHandler'--The built-in StorageHandler.
LOCATION 'volume://test_project/volume_data'
;
ALTER TABLE volume_ext_pt DROP IF EXISTS PARTITION (ds="20190102");
ALTER TABLE volume_ext_pt ADD PARTITION (ds="20190102") LOCATION "
volume://test_project/volume_data/20190102";
```

1.15.2.3 Use a custom StorageHandler to create a table

When the built-in StorageHandlers are unable to meet the requirements of your business, you can customize a StorageHandler through Java and specify some attributes of the Volume external table which you want to process data through.

The following example shows how to use a custom StorageHandler to create an external table.

Assume that the data type is TEXT and the column delimiter is "|". You can perform the following steps to create an external table:

1. Use MaxCompute Studio or MaxCompute Eclipse development plug-in to customize various Java classes.
2. Export the JAR package. In this example, the package name is odps-volume-example.jar.
3. Run the following command to add the JAR package to MaxCompute as a resource:

```
add jar odps-volume-example.jar -f;
```

4. Run the following commands to create an external table:

```
DROP TABLE IF EXISTS volume_ext;
CREATE EXTERNAL TABLE volume_ext
(
  key string,
  value string
)
STORED BY 'com.aliyun.odps.udf.example.text.TextStorageHandler'
WITH SERDEPROPERTIES (
  'delimiter'='|'
)
LOCATION 'volume://myproject/volume_data/20190102'
USING 'odps-volume-example.jar'
```

;

**Note:**

After the external table is created, you can operate the volume data through the external table.

1.15.3 Access a volume external table

Volume external tables can be accessed in the same way that you would access a MaxCompute table. Example:

```
select key,value from volume_ext_pt where ds="20190102";
```

1.16 Security solution

1.16.1 Target users

This User Guide is intended for all owners and administrators of MaxCompute projects, and users interested in the MaxCompute multi-tenant data security system. The MaxCompute multi-tenant data security system includes:

- User authentication
- User and authorization management of projects
- Cross-project resource sharing
- Project protection

1.16.2 Quick start

Add a user and grant permissions to the user

Scenario: Jack is the administrator of the prj1 project. A new team member Alice, who already has an Alibaba Cloud account (alice@aliyun.com), applies to join the project. Alice applies for the following permissions: viewing table lists, submitting jobs, and creating tables.

The admin performs the following operations to add Alice to the project:

```
use prj1
add user aliyun$alice@aliyun.com;
-- Add a user.
grant List, CreateTable, CreateInstance on project prj1 to user aliyun
$alice@aliyun.com
```



```
-- Grant permissions to the user.
```

Add a user and grant permissions to the user using an ACL

Scenario: Jack is the administrator of prj1. Three new members Alice, Bob, and Charlie join in as data reviewers. They require the following permissions: viewing table lists, submitting jobs, and reading the table userprofile.

The project administrator can use object-based ACL authorization in this scenario.

The operations are as follows:

```
use prj1
add user aliyun$alice@aliyun.com
-- Add a user.
add user aliyun$bob@aliyun.com
add user aliyun$charlie@aliyun.com
create role tableviewer
-- Create a role.
grant List, CreateInstance on project prj1 to role tableviewer; --
Grant permissions to the role
-- Grant permissions to the role.
grant Describe, Select on table userprofile to role tableviewer
grant tableviewer to aliyun$alice@aliyun.com
-- Grant the tableviewer role to a user.
grant tableviewer to aliyun$bob@aliyun.com
grant tableviewer to aliyun$charlie@aliyun.com
```

Package and share resources

Scenario: Jack is the administrator of prj1. John is the administrator of prj2. Due to business requirements, Jack wants to share some resources of prj1 (such as datamining.jar and sampletable) to John's prj2. A user in prj2 (Bob) requires access to these resources. The prj2 administrator can configure an ACL or policy to automatically authorize prj2 users to access these resources, without the intervention of Jack.

The operations are as follows:

1. Prj1 administrator Jack creates a resource package in prj1.

```
use prj1
create package datamining
-- Create a package.
add resource datamining.jar to package datamining
-- Add resources to the package.
add table sampletable to package datamining
-- Add the table to the package.
allow project prj2 to install package datamining
```

```
-- Share the package to prj2.
```

2. Prj2 administrator Bob installs the package in prj2.

```
use prj2
install package prj1.datamining
-- Install the package.
describe package prj1.datamining
-- View the resource list of the package.
```

3. Configure automatic authorization for Bob on the package.

```
use prj2
grant Read on package prj1.datamining to user aliyun$bob@aliyun.com
-- Use an ACL to allow Bob to use the package.
```



Note:

For more information about cross-project resource sharing, see [Cross-project resource sharing](#).

Configure project protection

Scenario: Jack is the administrator of project prj1. This project contains sensitive data such as user IDs and shopping records. The project also stores many data mining algorithms to which the organization holds intellectual property rights. Jack wants to protect the sensitive data and algorithms in the project. He wants the data to be accessible only to users in the project. The data must not be able to flow out of the project.

The operations are as follows:

```
use prj1
set ProjectProtection=true
-- Enable project protection.
```

When project protection is enabled, data in the project can flow only within the project. Data cannot flow out. In some cases, for example, a user (Alice) requires to export data tables for business purposes. This operation is approved by the project administrator. MaxCompute provides two methods to export data from a protected project.

Method 1: Create an exception policy. For more information, see [Data export methods when project protection is enabled](#).

1. Create a policy file. Create a policy file named `/tmp/exception_policy.txt`. It only allows Alice to export t1 from prj1 using a SQL task. The policy is defined as follows:

```
{
  "Version": "1",
  "Statement": [{
    "Effect": "Allow",
    "Principal": "ALIYUN$alice@aliyun.com",
    "Action": ["odps:Describe", "odps:Select"],
    "Resource": "acs:odps:*:projects/prj1/tables/t1",
    "Condition": {
      "StringEquals": { "odps:TaskType": "SQL" }
    }
  }]
}
```

2. Configure the exception policy.

```
use prj1
-- Enable project protection and configure an exception policy.
set ProjectProtection=true with exception /tmp/exception_policy.txt
```

**Note:**

When you configure the exception policy, ensure that the principal cannot update the data resources or recreate an object with the same name (using `DROP TABLE` and `CREATE TABLE`). This prevents data leakage due to time-of-check to time-of-use (TOC2TOU).

Method 2: Configure trusted projects. Configure prj2 as a trusted project of prj1 to enable data flow from prj1 to prj2. For more information, see [Data export methods when project protection is enabled](#).

```
use prj1
      add trustedproject prj2
```

**Note:**

In MaxCompute, package-based resource sharing and project protection are mutually independent mechanisms that take effect at the same time, but their functions are mutually restrictive.

In MaxCompute, resource sharing has a higher priority than project protection. This means, if an object in a protected project is shared with other projects through the package mechanism, cross-project access to this object is not subject to the project protection rules.

1.16.3 User authentication

The main purpose of user authentication is to verify the identity of a request sender. Authentication typically includes:

- Verifying the true identity of a message sender
- Checking whether the message was tampered with before it is received.

1.16.4 Project user and authorization management

1.16.4.1 Overview

Projects are the foundation of the MaxCompute multi-tenant system and the basic units of data management and computing. When you create a project, you are automatically the project owner. All objects in the project, such as tables, instances, resources, and UDFs, belong to you. Objects in the project can only be accessed by the owner and users that are authorized by the owner.

This topic describes users, roles, and authorization management of projects. For example, Alice is the owner of test_project, and another user from Alice's project team requests to access the resources in test_project. Alice can use the methods described in this topic to perform user and authorization management. If a user that wants access to Alice's project is not from her project team, Alice can implement cross-project sharing. For more information, see [Cross-project resource sharing](#).

1.16.4.2 User management

Add a user

If Alice (the project owner) decides to authorize another user, she must add the user to this project. Only users in a project can be authorized.

Run the following command to add a user:

```
add user <full_username>  
-- Add a user to a project.
```

Remove a user

When a user leaves the project team, Alice needs to remove the user from the project. After a user is removed from the project, the user no longer has any access permissions on project resources.

Run the following command to remove a user from a project:

```
remove user <full_username>  
-- Remove a user from a project.
```



Note:

- After a user is removed, the user no longer has any access permissions on project resources.
- Before you remove a user who has been assigned a role, you must first revoke the role. For information about roles, see [Role management](#).
- After a user is removed, the ACL authorization related to the user is retained. However, the policy authorization at the role level is revoked, and the policy authorization at the project level is retained. If the user is added to the project again, the previous ACL authorization of the user is re-activated.
- MaxCompute does not support complete removal of a user and the relevant authorization data.

1.16.4.3 Role management

A role is a collection of access permissions. A role can be used to assign the same permissions to a group of users. Role-based authorization can greatly simplify the authorization process and reduce authorization management costs. When granting permissions to users, you should consider using role-based authorization.

An admin role is automatically created when a project is created. This role is granted permissions to access all objects of the project, manage users and roles, and authorize users and roles. Compared with the project owner, the admin role cannot assign another user with the admin role, configure security rules for a project, or change the authentication model of the project. Permissions of the admin role cannot be modified.

The role management commands are as follows:

```
create role <rolename>  
-- Create a role.  
drop role <rolename>  
-- Delete a role.  
grant <rolename> to <username>  
-- Assign a role to a user.  
revoke <rolename> from <username>
```

-- Revoke the role of a user.

**Note:**

When you delete a role, MaxCompute checks whether there are users assigned with this role. If the role is assigned to users, the role fails to be deleted. To delete the role, you must revoke this role from all users.

1.16.4.4 ACL authorization actions

Authorization usually involves three elements: subject, object, and action. In MaxCompute, a subject is the user, there are various types of objects in a project, and actions are performed on objects. Different types of objects support different actions.

MaxCompute projects support the following object types and actions:

Table 1-53: Object types and actions

Object	Action	Description
Project	Read	Check the information about the project itself (not including any objects of the project), such as CreateTime.
Project	Write	Update the information of the project itself (not including any objects of the project), such as Comments.
Project	List	View a list of all types of objects in the project.
Project	CreateTable	Create a table in the project.
Project	CreateInstance	Create an instance in the project.
Project	CreateFunction	Create a function in the project.
Project	CreateResource	Create resource in the project.
Project	CreateJob	Create a job in the project.
Project	CreateVolume	Create a volume in the project.
Project	All	All the permissions above.
Table	Describe	Read the metadata of the table.
Table	Select	Read the information of the table.

Object	Action	Description
Table	Alter	Alter the metadata of the table; add or drop partitions.
Table	Update	Override or add data to the table.
Table	Drop	Drop the table.
Table	All	All the preceding permissions.
Function	Read	Read and execute permissions.
Function	Write	Update.
Function	Delete	Delete.
Function	All	All the preceding permissions.
Resource, instance, job, volume	Read	Read permissions.
Resource, instance, job, volume	Write	Update permissions.
Resource, instance, job, volume	Delete	Delete permissions.
Resource, instance, job, volume	All	All the preceding permissions.

**Note:**

In the preceding permissions, the CreateTable action of project objects, as well as the Select, Alter, Update, and Drop actions of table objects, must be used together with the CreateInstance action of project objects. Before using the preceding permissions to complete actions, you must assign the CreateInstance permission.

After adding users or creating roles, these users or roles should be authorized. The ACL authorization mechanism of MaxCompute is object-based. Authorization data (the access control list, or ACL) is considered as a sub-resource of an object). Therefore, ACL authorization can be performed only when the objects exist. When the objects are deleted, authorized permission data is automatically deleted.

The ACL of MaxCompute supports authorization using commands like SQL92-defined GRANT/REVOKE commands. Use the corresponding authorization commands to authorize existing project objects or revoke their authorization.

Command syntax:

```
grant actions on object to subject
revoke actions on object from subject
actions ::= action_item1, action_item2, ...
object ::= project project_name | table schema_name | instance
inst_name | function func_name | resource res_name
subject ::= user full_username | role role_name
```



Note:

The ACL authorization commands of MaxCompute do not support the [WITH GRANT OPTION] parameter. That is, when user A authorizes user B to access an object, user B cannot authorize user C to access the same object. Therefore, all authorization actions must be completed by users with at least one of the following identities:

- Project owner
- Users with the admin role in the project
- Object creators in the project

ACL authorization example:

Scenario: Users `alice@aliyun.com` and `bob@aliyun.com` are new members of `test_project`. In `test_project`, they must submit jobs, create data tables, and view existing objects of the project. The administrator then takes the following authorization actions:

```
use test_project
-- Open a project.
security
add user aliyun$alice@aliyun.com
-- Add a user.
add user aliyun$bob@aliyun.com
-- Add a user.
create role worker
-- Create a role.
grant worker TO aliyun$alice@aliyun.com
-- Assign a role.
grant worker TO aliyun$bob@aliyun.com
-- Assign a role.
grant CreateInstance, CreateResource, CreateFunction, CreateTable,
List ON PROJECT test_project TO ROLE worke r
```



```
-- Authorize a role.
```

1.16.4.5 View permissions

MaxCompute allows you to view permissions in different dimensions. For example, you can view permissions of a specified user, permissions of a specified role, or the authorization list of a specified object.

View the permissions of a user

```
show grants
-- View the access permissions of the current user.
show grants for <username>
-- View the access permissions of a specified user. Only project
  owners and administrators can view the access permissions of a
  specified user.
```

View the permissions of a role

```
describe role <rolename>
-- View the access permissions granted to a specified role.
```

View the authorization list of an object

```
show acl for <objectName> [on type <objectType>]
-- View the authorization list of a specified object.
```



Note:

If [on type <objectType>] is not specified, the default type is table.

MaxCompute uses characters A, C, D, and G to indicate the permissions of users or roles. The characters are described as follows:

- **A: allow.** Access is allowed.
- **D: deny.** Access is denied.
- **C: condition.** This is a conditional authorization. This character appears only in the policy authorization system. For more information, see [Condition block structure](#).
- **G: grant.** You can grant permissions to this object.

Example:

```
odps@test_project> show grants for aliyun$odpctest1@aliyun.com
[roles]
dev
Authorization Type: ACL
[role/dev]
A projects/test_project/tables/t1: Select [user/odpctest1@aliyun.com]
A projects/test_project: CreateTable | CreateInstance | CreateFunction
  | List
A projects/test_project/tables/t1: Describe | Select
```

```
Authorization Type: Policy  
[role/dev]  
AC projects/test_project/tables/test_*: Describe  
DC projects/test_project/tables/alifinance_*: Select [user/odptest1@  
aliyun.com]  
A projects/test_project: Create* | List  
AC projects/test_project/tables/alipay_*: Describe | Select  
Authorization Type: ObjectCreator  
AG projects/test_project/tables/t6: All  
AG projects/test_project/tables/t7: All
```

1.16.5 Cross-project resource sharing

1.16.5.1 Overview

You are the owner or administrator (admin role) of a project, and someone requests to access resources of your project. If the applicant is a member of your project team, we recommend that you use the user and authorization management features for your project. For more information, see [User and authorization management of projects](#). If the applicant is not a member of your project team, you can use the package-based resource sharing feature described in this topic.

A package is used to share data and resources across projects. It can be used to implement cross-project user authorization. The following scenario describes a problem that can only be resolved effectively with the package mechanism.

Members of the Alifinance project need to access Alipay project data. The Alipay project administrator adds Alifinance project users to the Alipay project, and then grants the new users common permissions. For security concerns, the Alipay project administrator does not want to authorize every user of the Alifinance project team. A mechanism is required to allow the Alifinance project administrator to control access to the authorized objects.

By using the package feature, the Alipay project administrator can package the objects that the Alifinance team needs to access, and then allow the package to be installed in the Alifinance project. After installing the package, the Alifinance project administrator can decide whether to grant permissions on the package to users in the Alifinance project.

A package involves two subjects: package creator and package user. The package creator provides resources. The package creator packages the resources to be shared and the corresponding access permissions, and provides the package receiver with the permissions to install and use the package. The package user

consumes the resources. After installing the package published by the package creator, the package user can directly access the resources.

The following topics describe the operations that can be performed by a package creator and package user.

1.16.5.2 Package usage

1.16.5.2.1 Operations for package creators

Create a package

Run the following commands to create a package:

```
create package <pkgname>
```

Delete a package

Run the following commands to drop a package:

```
delete package <pkgname>
```

Add a resource to be shared to the package

Run the following commands to add a resource to the package:

```
add project_object to package package_name [with privileges <
privileges>]
remove project_object from package package_name
project_object ::= table table_name | instance inst_name | function
func_name | resource res_name
privileges ::= action_item1, action_item2, ...
```



Note:

- The types of supported objects exclude projects, so you cannot use a package to create objects in other projects.
- In addition to the objects, the operation permissions on the objects are also added to the package. When not passed [with privileges Privileges] When you specify an action permission, the default is read-only, that is, read/describe/select. An object (resource) and its permissions are considered as a whole. You can delete resources in a package. The permissions are revoked when resources are deleted.

Allow other projects to use a package

Run the following commands to allow other projects to use a package:

```
allow project <prjname> to install package <pkgname> [using label <number>]
```

Revoke the permission for other projects to use a package

Run the following commands to revoke another project's permission to use package:

```
disallow project <prjname> to install package <pkgname>
```

View the list of packages already created and installed

Run the following commands to view the list of packages already created and installed:

```
show packages
```

View details of a package

Run the following commands to view details of a package:

```
describe package <pkgname>
```

1.16.5.2.2 Operations for package users

The installed package is a type of independent object in MaxCompute. To access resources in a package (other projects' resources shared with you), you must have the permission to read the package. If you do not have read permissions, submit an application to the project owner or admin for the permissions. The project owner or admin can grant the permissions by using ACL authorization or policy authorization.

For example, the following ACL authorization rule allows user `odps_test@aliyun.com` to access resources in a package:

```
use prj2 security
install package prj1.testpkg
grant read on package prj1.testpackage to user aliyun$odps_test@aliyun.com
```

The following policy authorization rule allows any user in `prj2` to access resources in a package:

```
use prj2
```

```
install package prj1.testpkg  
put policy /tmp/policy.txt
```

The contents of /tmp/policy.txt are as follow:

```
{  
  "Version": "1", "Statement": [{  
    "Effect": "Allow",  
    "Principal": "*",  
    "Action": "odps:Read", "Resource": "acs:odps:*:projects/prj2/packages/  
prj1.testpkg"  
  }]  
}
```

Install a package

Run the following commands to install a package:

```
install package <pkgname>;
```



Note:

**The pkgName of a package to be installed must be in the format of
<projectName>.<packageName>.**

Uninstall package

Run the following commands to uninstall a package:

```
uninstall package <pkgname>;
```



Note:

**The pkgName of a package to be uninstalled must be in the format of
<projectName>.<packageName>.**

View packages

Run the following commands to view packages:

```
show packages  
-- View the list of packages already created and installed.  
describe package <pkgname>  
-- View details of a package.
```

1.16.6 Project protection

1.16.6.1 Overview

Some enterprises (such as financial institutions and military enterprises) have high data security requirements. For example, their employees can only perform their

jobs in the workplace, and are not allowed to take work materials out of the office . All USB ports on office computers are disabled. These measures aim to prevent leakage of sensitive data.

For example, you are a MaxCompute project administrator in charge of a project with sensitive data. The data must not be shared to other projects. You are required to perform the following configurations to prohibit all operations that could result in data outflow.

1.16.6.2 Data protection

MaxCompute provides a project protection mechanism that prohibits operations that introduce data leakage risks. You can simply configure your project as follows to enable project protection:

```
set security.ProjectProtection=true
-- Enable project protection. This rule allows inbound data flows, but
prohibits outbound data flows.
```

After project protection is enabled, the data flow of the project is controlled. Data can flow in, but cannot flow out.

Project protection is disabled by default (ProjectProtection = false). If you have access permissions on multiple projects, you can use any cross-project data access to migrate data between projects. If a project stores highly-sensitive data, the administrator must configure a project protection mechanism.

1.16.6.3 Data export methods when project protection is enabled

After you enable project protection, you may soon encounter this situation: A user (Alice) submits a request to export the data of a table from the project. It is verified that this table contains no sensitive data. MaxCompute provides two methods to export data after project protection is enabled.

Configure an exception policy

The project owner can run the following command to attach an exception policy when enabling project protection:

```
SET ProjectProtection=true WITH EXCEPTION <policyFile>
```



Note:

This policy mechanism is different from the policy-based authorization mechanism (though the command syntax is the same). This policy describes exceptions of the project protection mechanism. All access requests matching the policy are not subject to the project protection rules.

Example:

The following policy allows the user `Alice@aliyun.com` to export data out of the `alipay` project when performing the `SELECT` operation on the `alipay.table_test` table in a SQL task:

```
{
  "Version": "1", "Statement": [{
    "Effect": "Allow", "Principal": "ALIYUN$Alice@aliyun.com",
    "Action": ["odps:Select"],
    "Resource": "acs:odps:*:projects/alipay/tables/table_test",
    "Condition": {
      "StringEquals": { "odps:TaskType": ["DT", "SQL"]
    }
  }
}]
}
```



Note:

- The preceding exception policy does not grant any permissions. If Alice does not have the `SELECT` permission on `alipay.table_test`, the preceding exception policy does not allow Alice to export data. Project protection specifies data flow control, not access control. Data flow control is effective only when a user can access the target data.
- Data leakage due to TOC2TOU (also known as the race condition problem) arises in the following situation:
 1. [TOC stage] User A submits an application to the project owner to export table `t1`. The project owner verifies that `t1` does not contain sensitive data. The project owner configures an exception policy, which allows user A to export `t1`.
 2. A malicious user changes the content of `t1` by writing sensitive data to it.
 3. [TOU stage] User A exports `t1`. The `t1` exported by the user is not the same `t1` that was authorized by the project owner.

Suggestions on TOC2TOU prevention: For a table that a user applies to export, the project owner must make sure that no other user (including admins) can update the table or create a table with the same name (using `DROP TABLE` and

CREATE TABLE). In the preceding TOC2TOU scenario, we recommend that the project owner create a snapshot of t1 in step 1. Then, create an exception policy for the user to use this snapshot. Do not grant the admin role to any users.

Configure a trusted project

If the current project is protected, and the target project is a trusted project of the current project, data flows to the target project are not subject to the project protection rules. If each project in a group is mutually configured as trusted projects, the group is considered a trusted project group. Data can flow freely within the group, but cannot flow out.

Run the following command to manage trusted projects:

```
list trustedprojects
-- Show all trusted projects of the current project.
add trustedproject <projectname>
-- Add a trusted project of the current project.
remove trustedproject <projectname>
-- Remove a trusted project of the current project.
```

1.16.6.4 Resource sharing and data protection

In MaxCompute, package-based resource sharing and project protection are mutually independent mechanisms that take effect at the same time, but their functions are mutually restrictive.

In MaxCompute, resource sharing takes precedence over project protection. If a data object is shared to users in other projects through resource sharing, the project protection rules will not apply to this data object.

To prevent data outflow from the project, after you enable project protection (ProjectProtection=true), you must verify the following points:

- **Make sure that no trusted projects are added. If one is added, evaluate possible risks.**
- **Make sure that no exception policies are configured. If one is configured, evaluate possible risks, especially risks due to TOC2TOU.**
- **Check whether package data sharing is not in use. If package data sharing is in use, make sure that the package contains no sensitive data.**

1.16.7 Project security configuration

MaxCompute is a multi-tenant data processing platform. Different tenants may have different data security requirements. MaxCompute provides project-level security configuration to satisfy data security requirements of different tenants. Project owners can customize their external accounts and authentication models as required.

MaxCompute supports multiple orthogonal authorization mechanisms, such as ACL-based authorization, policy-based authorization, and implicit authorization (for example, an object creator is automatically authorized to access the object). However, not all users require these security mechanisms. You can configure an authentication model that best suits your business demands and usage habits.

```
show SecurityConfiguration
-- View the security configuration of the project.
set security.CheckPermissionUsingACL=true/false
-- Enable or disable ACL-based authorization. Default value: true.
set security.CheckPermissionUsingPolicy=true/false
-- Enable or disable policy-based authorization. Default value: true.
set security.ObjectCreatorHasAccessPermission=true/false
-- Allow or disallow an object creator to be granted the object access
  permission by default. Default value: true.
set security.ObjectCreatorHasGrantPermission=true/false
-- Allow or disallow an object creator to be granted the authorization
  permission by default. Default value: true.
set security.LabelSecurity=true/false
-- Enable or disable the label security policy.
set security.ProjectProtection=true/false
-- Enable or disable project protection to allow or prohibit data
  transfer from the project.
```

1.16.8 Authorization policies

1.16.8.1 Policy overview

Policy authorization is a principal-based authorization. Permission data authorized by policy (that is, access policy) is considered as a type of sub-resource of the authorization subject. Policy authorization can be performed only if the subject exists. When a subject is deleted, their authorization data is deleted automatically. Policy authorization uses an access policy language customized for MaxCompute to allow or deny subjects access to project objects.

Policy authorization is a new authorization mechanism mainly used to handle complicated authorization scenarios that ACL authorization struggles to deal with, such as:

- Authorize a group of objects, such as all functions and all tables starting with taobao, at a time.
- For authorizations with restrictive conditions, such as one that takes effect only in a specified period, one that takes effect only if the requester initiates the request from a specified IP addresses, or one that allows the user to use SQL only (disallowing other tasks) to access a table.

The command format of policy authorization is as follows:

```
GET POLICY;
-- Read the project policy.
PUT POLICY <policyFile>;
-- Set (overwrite) the project policy.
GET POLICY ON ROLE <roleName>;
-- Read the policy of a role in the project.
PUT POLICY <policyFile> ON ROLE <roleName>;
-- Set (overwrite) the policy of a role in the project.
```



Note:

MaxCompute currently supports project policies and role policies. A project policy applies to all users of the project, while a role policy applies only to users to whom the role is assigned. You must specify a principal (user) for project policies, but you cannot specify a principal for role policies, because the role will be assigned to users.

An example of project policy authorization is as follows:

Scenario: Authorized user `alice@aliyun.com` can only submit a request before 23:59:59 2017-11-11 from an IP address in the subnet 10.32.180.0-23, and can only perform the `CreateInstance`, `CreateTable`, and `List` operations in `test_project`. No tables in `test_project` can be dropped.

The policy is as follows:

- ```
{
 "Version": "1", "Statement": [{
 "Effect": "Allow", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": ["odps:CreateTable", "odps:CreateInstance", "odps:List"],
 "Resource": "acs:odps:*:projects/test_project",
 "Condition": { "DateLessThan": {
 "acs:CurrentTime": "2017-11-11T23:59:59Z"
 } },
 "IpAddress": { "acs:SourceIp": "10.32.180.0/23"
 }
 }],
 },
 {
 "Effect": "Deny", "Principal": "ALIYUN$alice@aliyun.com", "Action": "odps:Drop", "Resource": "acs:odps:*:projects/test_project/tables/*"
 }
]
```

```

]]
 }
 ``json

```

- ``json
 

```

{
 "Version": "1", "Statement": [{
 "Effect": "Allow", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": ["odps:CreateTable", "odps:CreateInstance", "odps:List"],
 "Resource": "acs:odps:*:projects/test_project",
 "Condition": { "DateLessThan": {
 "acs:CurrentTime": "2017-11-11T23:59:59Z"
 }
 },
 "IpAddress": { "acs:SourceIp": "10.32.180.0/23"
 }
 },
],
 {
 "Effect": "Deny", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": "odps:Drop",
 "Resource": "acs:odps:*:projects/test_project/tables/*"
 }
]
}

```
- {
 

```

{
 "Version": "1", "Statement": [{
 "Effect": "Allow", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": ["odps:CreateTable", "odps:CreateInstance", "odps:List"],
 "Resource": "acs:odps:*:projects/test_project",
 "Condition": { "DateLessThan": {
 "acs:CurrentTime": "2017-11-11T23:59:59Z"
 }
 },
 "IpAddress": { "acs:SourceIp": "10.32.180.0/23"
 }
 },
],
 {
 "Effect": "Deny", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": "odps:Drop",
 "Resource": "acs:odps:*:projects/test_project/tables/*"
 }
]
}

```

**Note:**

- **Currently, only role policies and project policies are supported, and user policies are not.**
- **Every policy only supports one policy file. Since put policies override existing policies, you must follow the sequence below to modify a policy:**
  - 1. Get Policy.**
  - 2. Merge policy statements.**
  - 3. Put policies.**

### 1.16.8.2 Policy-related terms

**Permission is a basic concept of access control. When a requester wants to take an action on a resource, the action may be allowed or denied, based on the permission settings. A statement refers to the formal description of a single permission, and policy refers to a set of statements.**

**An access policy comprises the following access control elements: principal, action, resource, access restriction, and effect. These elements are briefly described below.**

#### Principal

**A principal of an object is a user or group to which permissions are assigned in an access policy. For example, the access policy allows Michael to perform the CreateObject action on the resource SampleBucket before December 31, 2017. Michael is the principal of the object.**

#### Action

**An action is an activity that the principal has permission to perform. For example, the access policy allows Michael to perform the CreateObject action on the resource SampleBucket before December 31, 2017. Therefore, CreateObject is an action of the access policy.**

#### Resource

**Resource is the object a principal requests access to. For example, the access policy allows Michael to perform the CreateObject action on the resource SampleBucket before December 31, 2017. SampleBucket is a resource of the access policy.**

#### Access restriction

**Access restriction is the prerequisite for the permission to take effect. For example, the access policy allows Michael to perform the CreateObject action on resource SampleBucket before December 31, 2017. The access restriction is before December 31, 2017.**

#### Effect

**Authorization effect has two options: Allow (action) or deny (action). In general, deny actions are generally more efficient and are checked first during permission checks.**

**Notice:**

The deny action and revoking permission are completely different concepts. The latter usually revokes permissions for both allow action and deny action. For example, a traditional database supports the revoke and revoke deny actions.

### 1.16.8.3 Access policy structure

#### 1.16.8.3.1 Overview

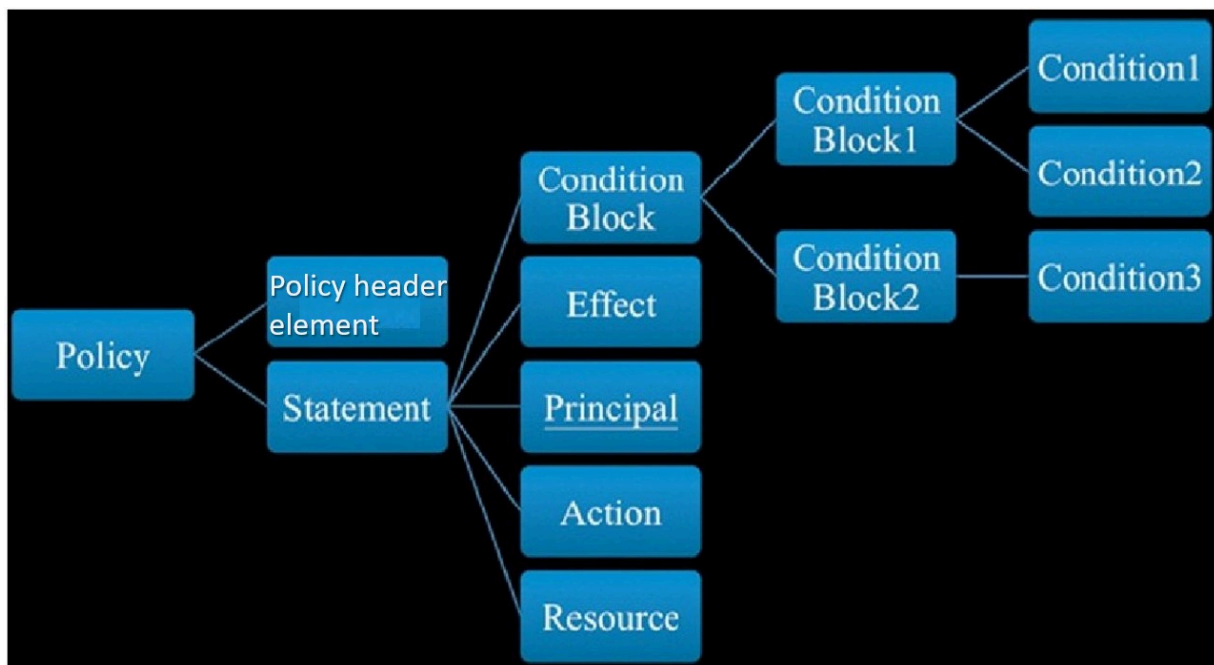
The following figure shows the structure of an access policy. A policy consists of the following parts:

- An optional policy header
- One or more statements

The policy header is optional and includes the policy version. The policy body is a set of statements.

The following figure shows the structure of a policy.

Figure 1-16: Policy structure



#### 1.16.8.3.2 Authorization statement structure

An authorization statement includes the following entries:

- **Effect:** indicates the permission type of this statement. The value can be either Allow or Deny.
- **Principal:** If a policy is bound to a user or role in the authorization process, such as the role policy of MaxCompute, you cannot appoint a principal. If a policy is bound to a project or objects of the project in the authorization process, such as the project policy of MaxCompute, you must specify a principal.
- **Action:** indicates the authorization operation. It can be one or more operation names, and supports the asterisk (\*) and question mark (?) wildcard characters. The asterisk (\*) matches any number of characters, and the question mark (?) matches a single character. For example, Action = \* indicates all operations.
- **Resource:** indicates the authorization object. It can be one or more object names, and supports the asterisk (\*) and question mark (?) wildcard characters. The asterisk (\*) matches any number of characters, and the question mark (?) matches a single character. For example, Resource = \* indicates all objects.
- **Condition block:** indicates the conditions that must be met for the permission described by this authorization statement to take effect. See the next topic for the structure of the condition block.

### 1.16.8.3.3 Conditional block structure

A condition block consists of one or more condition clauses. A condition clause consists of an action type, keyword, and condition value. The action types and keywords will be described in detail in the subsequent sections.

Whether a condition block is satisfied is determined as follows:

- A conditional keyword can correspond to one or more values. If the conditional keyword value is equal to one of the corresponding values, the condition is satisfied.
- A condition clause of a conditional operation type is satisfied if all conditional keywords in the clause are satisfied.
- A condition block is satisfied only if all of its condition clauses are satisfied.

### 1.16.8.3.4 Conditional action type

The following action types are supported: string, number, date, Boolean, and IP address. The methods supported by each conditional operation type are as follows:

**String:**

```
StringEquals
StringNotEquals
StringEqualsIgnoreCase
StringNotEqualsIgnoreCase
StringLike
StringNotLike
```

**Numeric:**

```
NumericEquals
NumericEquals
NumericLessThan
NumericLessThanEquals
NumericGreaterThan
NumericGreaterThanEquals
```

**Date and time:**

```
DateEquals
DateNotEquals
DateLessThan
DateLessThanEquals
DateGreaterThan
DateGreaterThanEquals
```

**Boolean:**

```
Bool
```

**IP address:**

```
IpAddress NotIpAddress
```

### 1.16.8.3.5 Conditional keywords

**MaxCompute supports the conditional keywords reserved by Alibaba Cloud Service (ACS ). The following table describes these conditional keywords.**

Table 1-54: Conditional keywords

| Conditiona<br>l keywords<br>reserved by<br>ACS | Type             | Description                                                                                                               |
|------------------------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------------|
| acs:<br>CurrentTime                            | Date and<br>time | The time when the Web server receives a request. It is based on the ISO 8601 standard, for example, 2017-11-11T23:59:59Z. |

| Conditional keywords reserved by ACS | Type       | Description                                                                  |
|--------------------------------------|------------|------------------------------------------------------------------------------|
| acs:SecureTransport                  | Boolean    | Whether the request is sent over a secure channel, such as an HTTPS channel. |
| acs:SourceIp                         | IP address | The IP address of the client that sent the request.                          |
| acs:UserAgent                        | String     | The UserAgent of the client that sent the request.                           |
| acs:Referer                          | String     | The HTTP referer that sent the request.                                      |

**Note:**

acs:SourceIp refers to the remote\_ip of the HTTP connection, not the (leftmost) client IP address in the x-forwarded-for HTTP header field. For example, if 10.230.205.105 is a LAN IP address, acs:SourceIp is the egress gateway IP address of this LAN. If the network egress uses a proxy server, acs:SourceIp is the IP address of the proxy server. If the request traverses across multiple proxy servers, acs:SourceIp is the IP address of the final proxy server. The value of acs:SourceIp may vary depending on the rules configured on the proxy server.

## 1.16.8.4 Access policy norm

### 1.16.8.4.1 Principal naming convention

The principal is the request sender. Currently, only an Alibaba Cloud account, domain account, or Taobao account is accepted as a principal. A cloud account can be represented by ID or DisplayName.

**Example:**

```
"Principal": "43274"
"Principal": "ALIYUN$bob@aliyun.com"
```



```
"Principal": ["ALIYUN$bob@aliyun.com", "ALIYUN$jack@aliyun.com", "TAOBAO$alice"]
```

## 1.16.8.4.2 Resource naming convention

The following naming conventions are used for MaxCompute resources.

```
acs:<service-name>:<namespace>:<relative-id>
```

The parameters are described as follows.

Table 1-55: Parameters

| Name         | Description                                                                                                                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| acs          | Retained resource header.                                                                                                                                                                                                                                                                                                   |
| service-name | The name of an open cloud service, such as MaxCompute, OSS, and TableStore.                                                                                                                                                                                                                                                 |
| namespace    | Naming space, used for resource isolation. If a cloud account ID is used for resource isolation, the value can be the cloud account ID. If this option is not supported, use the asterisk (*) wildcard character instead.                                                                                                   |
| relative-id  | Indicates the service-related resource. Its meaning depends on specific services. This part of the format description supports a tree structure similar to the file path. Using MaxCompute as an example, the format of relative-id is:<br><pre>projects/&lt;project_name&gt;/&lt;object_type&gt;/&lt;object_name&gt;</pre> |

Some MaxCompute resource naming examples.

Table 1-56: Naming examples

| Item                         | Description                                                 |
|------------------------------|-------------------------------------------------------------|
| *                            | All objects in the project.                                 |
| projects/prj1/tables/t1      | Table t1 of Project prj1.                                   |
| projects/prj1/instances/*    | All instances of Project prj1.                              |
| projects/prj1/tables/*       | All tables of Project prj1.                                 |
| projects/prj1/tables/taobao* | All tables of Project prj1 whose names start with "taobao". |

### 1.16.8.4.3 Action naming

Action naming conventions are as follows:

```
<service-name>:<action-name>
```

**Description:**

- **service-name:** name of an open cloud service, for example, maxcompute, oss, and table store.
- **action-name:** name of service-related action APIs.

The following table lists MaxCompute action naming examples.

Table 1-57: MaxCompute action naming examples

| Naming example   | Description                                             |
|------------------|---------------------------------------------------------|
| *                | All actions.                                            |
| odps:*           | All MaxCompute actions.                                 |
| odps:CreateTable | The CreateTable action of MaxCompute.                   |
| odps:Create*     | All MaxCompute actions whose names start with "Create". |

### 1.16.8.4.4 Condition keys naming

The naming format for condition keys retained by the open cloud service is:

```
acs:<condition-key>
```

**Description:**

**condition-key:** ACS reserves 5 types of condition keys, which are accessible for all open services. They are: acs:CurrentTime, acs:SecureTransport, acs:SourceIp, acs:UserAgent, acs:Referer.

The naming format for condition keys related to the specific service is:

```
<service-name>:<condition-key>
```

**Description:**

**Condition-key:** service-defined condition key.

### 1.16.8.4.5 Access policy example

**Policy example:**

```
{
 "Version": "1",
 "Statement": [{
 "Effect": "Allow",
 "Principal": "ALIYUN$alice@aliyun.com",
 "Action": ["odps:CreateTable", "odps:CreateInstance", "odps:List"],
 "Resource": "acs:odps:*:projects/prj1",
 "Condition": { "DateLessThan": {
 "acs:CurrentTime": "2017-11-11T23:59:59Z"
 }
 },
 "IpAddress": { "acs:SourceIp": "10.32.180.0/23"
}
}],
{
 "Effect": "Deny", "Principal": "ALIYUN$alice@aliyun.com",
 "Action": "odps:Drop",
 "Resource": "acs:odps:*:projects/prj1/tables/*"
}
}]
}
```



**Note:**

The authorized user (alice@aliyun.com) can only submit a request from subnet 10.32.180.0/23 before 2017-11-11T23:59:59Z. The user can only perform the CreateInstance, CreateTable, and List operations on the prj1 project. The user cannot delete tables from prj1.

### 1.16.8.5 Differences between policy authorization and ACL authorization

**ACL authorization:**

- Use ACL authorization to grant or revoke permissions when both the grantee (such as a user or role) and the object (such as a table) exist. Like the security feature of Oracle authorization, this avoids the security risk out of dropping and recreating an object with the same name.
- When dropping an object, all authorizations related to the object are automatically revoked.
- It only supports allow (whitelist) authorization, and does not support deny (blacklist) authorization.
- Use the classic Grant/Revoke commands for authorization. The command is simple and not prone to mistakes. Conditional authorization is not supported.

- This method is suitable for simple scenarios where condition or deny is not needed for authorization, and only the existing objects need to be authorized.

#### Policy authorization:

- Use policy authorization to grant or revoke permissions when the grantee or object is not available. The Object parameter supports wildcard "". For example, `projects/tbproj/tables/taobao` matches all tables whose names start with `taobao` in project `tbproj`. Like the features of MySQL authorization, policy authorization allows a non-existent object to be authorized, so the authorizer must consider the security risks of dropping and recreating an object with the same name.
- When dropping an object, the policy authorization related to this object is not deleted.
- Both allow (whitelist) authorization and deny (blacklist) authorization are supported. If allow and deny conflict, the deny action takes priority.
- Conditional authorization is supported. The authorizer can enforce 20 conditions on allow or deny authorization. For example, these conditions can be used to limit access to IP addresses within a subnet, and allow access before 23:59:59 on November 11, 2017.
- This method is suitable for relatively complicated scenarios where conditional authorization and deny action are needed and non-existent objects need to be authorized.
- Using policy authorization is more complicated than ACL authorization, but provides more flexibility.

### 1.16.8.6 Application limits

Table 1-58: Application limits

| Item                                 | Limit | Description                               |
|--------------------------------------|-------|-------------------------------------------|
| ACCESS_POLICY_SIZE_LIMIT             | 32 KB | The maximum size of AccessPolicy.         |
| USER_NUMBER_LIMIT_IN_ON<br>E_PROJECT | 1,000 | The maximum number of users in a project. |
| ROLE_NUMBER_LIMIT_IN_ON<br>E_PROJECT | 500   | The maximum number of roles in a project. |

| Item                                          | Limit | Description                                                        |
|-----------------------------------------------|-------|--------------------------------------------------------------------|
| ROLE_NAME_LENGTH_LIMIT                        | 64    | The maximum number of characters in a role name.                   |
| SECURITY_COMMENT_SIZE_LIMIT                   | 1 KB  | The maximum size of a security comment.                            |
| PACKAGE_NAME_LENGTH_LIMIT                     | 128   | The maximum number of characters in a package name.                |
| ALLOW_PROJECT_NUMBER_LIMIT_IN_ONE_PACKAGE     | 1024  | The maximum number of projects installed in a package.             |
| RESOURCE_NUMBER_LIMIT_IN_ONE_PACKAGE          | 256   | The maximum number of resources in a package.                      |
| PACKAGE_NUMBER_LIMIT_IN_ONE_PROJECT           | 512   | The maximum number of packages that can be created in a project.   |
| INSTALLED_PACKAGE_NUMBER_LIMIT_IN_ONE_PROJECT | 64    | The maximum number of packages that can be installed in a project. |

## 1.16.9 Collection of security statements

### 1.16.9.1 Project security configuration

#### Authentication

Table 1-59: Authorization configuration statements

| Statement                                       | Description                                                                                  |
|-------------------------------------------------|----------------------------------------------------------------------------------------------|
| show SecurityConfiguration                      | Displays the project security configuration.                                                 |
| set CheckPermissionUsingACL=true/false          | Enables or disables ACL-based authorization.                                                 |
| set CheckPermissionUsingPolicy=true/false       | Enables or disables policy-based authorization.                                              |
| set ObjectCreatorHasAccessPermission=true/false | Allows or disallows an object creator to be granted the object access permission by default. |

| Statement                                             | Description                                                                                                   |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <b>set ObjectCreatorHasGrantPermission=true/false</b> | <b>Allows or disallows an object creator to be granted the ACL-based authorization permission by default.</b> |

## Data protection

Table 1-60: Data protection statements

| Statement                                                         | Description                                      |
|-------------------------------------------------------------------|--------------------------------------------------|
| <b>set ProjectProtection=false</b>                                | <b>Disables project protection.</b>              |
| <b>set ProjectProtection=true [with exception &lt;policy&gt;]</b> | <b>Enables project protection.</b>               |
| <b>list TrustedProjects</b>                                       | <b>Displays the list of trusted projects.</b>    |
| <b>add TrustedProject &lt;projectName&gt;</b>                     | <b>Adds a project to trusted projects.</b>       |
| <b>remove trustedproject &lt;projectname&gt;;</b>                 | <b>Removes a project from trusted projects .</b> |

## 1.16.9.2 Project permission management

## User management

Table 1-61: User management statements

| Statement                            | Description                   |
|--------------------------------------|-------------------------------|
| <b>list users</b>                    | <b>Lists all added users.</b> |
| <b>add user &lt;username&gt;</b>     | <b>Adds a user.</b>           |
| <b>remove user &lt;username &gt;</b> | <b>Removes a user.</b>        |

## Role management

Table 1-62: Role management statements

| Statement                           | Description                     |
|-------------------------------------|---------------------------------|
| <b>list roles</b>                   | <b>Lists all created roles.</b> |
| <b>create role &lt;rolename&gt;</b> | <b>Creates a role.</b>          |
| <b>drop role &lt;rolename&gt;</b>   | <b>Deletes a role.</b>          |

| Statement                                            | Description                                |
|------------------------------------------------------|--------------------------------------------|
| <b>grant &lt;rolelist&gt; to &lt;username&gt;</b>    | <b>Revokes roles from a user.</b>          |
| <b>revoke &lt;rolelist&gt; from &lt;username&gt;</b> | <b>Grants one or more roles to a user.</b> |

## ACL-based authorization

Table 1-63: ACL-based authorization statements

| Statement                                                                                    | Description                             |
|----------------------------------------------------------------------------------------------|-----------------------------------------|
| <b>grant &lt;privList&gt; on &lt;objType&gt; &lt;objName&gt; to user &lt;username&gt;</b>    | <b>Authorizes a user.</b>               |
| <b>grant &lt;privList&gt; on &lt;objType&gt; &lt;objName&gt; to role &lt;rolename&gt;</b>    | <b>Authorizes a role.</b>               |
| <b>revoke &lt;privList&gt; on &lt;objType&gt; &lt;objName&gt; from user &lt;username&gt;</b> | <b>Revokes permissions from a user.</b> |
| <b>revoke &lt;privList&gt; on &lt;objType&gt; &lt;objName&gt; from role &lt;rolename&gt;</b> | <b>Revokes permissions from a role.</b> |

## Policy-based authorization

Table 1-64: Policy-based authorization statements

| Statement                                                     | Description                                           |
|---------------------------------------------------------------|-------------------------------------------------------|
| <b>get policy</b>                                             | <b>Displays policy settings at the project level.</b> |
| <b>put policy &lt;policyFile&gt;</b>                          | <b>Configures a policy at the project level.</b>      |
| <b>get policy on role &lt;roleName&gt;</b>                    | <b>Displays the policy settings of a role.</b>        |
| <b>put policy &lt;policyFile&gt; on role &lt;roleName&gt;</b> | <b>Configures a policy for a role.</b>                |

## Permission review

Table 1-65: Permission review statements

| Statement     | Description                                         |
|---------------|-----------------------------------------------------|
| <b>whoami</b> | <b>Displays information about the current user.</b> |

| Statement                                                              | Description                                                      |
|------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>show grants [for &lt;username&gt;] [on type &lt;objectType&gt;]</b> | Displays permissions and roles of a user .                       |
| <b>show acl for &lt;objectName&gt; [on type &lt;objectType&gt;]</b>    | Displays the authorization information of an object.             |
| <b>describe role &lt;roleName&gt;</b>                                  | Displays the authorization and assignment information of a role. |

### 1.16.9.3 Package-based resource sharing

#### Resource sharing

Table 1-66: Resource sharing statements

| Statement                                                                                         | Description                                         |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>create package &lt;pkgName&gt;</b>                                                             | Creates a package.                                  |
| <b>delete package &lt;pkgName&gt;</b>                                                             | Deletes a package.                                  |
| <b>add &lt;objType&gt; &lt;objName&gt; to package&lt;pkgName&gt; [with privileges privs]</b>      | Adds resources that need to be shared to a package. |
| <b>remove &lt;objType&gt; &lt;objName&gt; from package &lt;pkgName&gt;</b>                        | Removes shared resources from a package.            |
| <b>allow project &lt;prjName&gt; to install package &lt;pkgName&gt; [using label &lt;num&gt;]</b> | Allows a project to use a user package.             |
| <b>disallow project &lt;prjName&gt; to install package &lt;pkgName&gt;</b>                        | Disables a project from using a user package.       |

#### Resource use

Table 1-67: Resource use statements

| Statement                                | Description           |
|------------------------------------------|-----------------------|
| <b>install package &lt;pkgName&gt;</b>   | Installs a package.   |
| <b>uninstall package &lt;pkgName&gt;</b> | Uninstalls a package. |



## Package viewing

Table 1-68: Package viewing statements

| Statement                                | Description                                      |
|------------------------------------------|--------------------------------------------------|
| <b>show packages</b>                     | <b>Lists all created and installed packages.</b> |
| <b>describe package &lt; pkgName&gt;</b> | <b>Views details of a package.</b>               |

## 1.17 Frequently-used tools

### 1.17.1 MaxCompute console

#### 1.17.1.1 Usage notes

This topic provides usage notes for the MaxCompute client.

**Notice:**

- Do not rely on the output data of the client in any development or planning processes, as the data format may change. The client output format may not be forward compatible. The command syntax and behavior vary according to versions.
- The MaxCompute client is a Java program. It requires JRE to run. You need to download and install JRE 1.8 to use the MaxCompute client.

For more information about how to configure and use the client, see [Quick start](#).

#### 1.17.1.2 Install the client

This topic describes how to install the MaxCompute client.

1. Download the client package to your client computer.
2. Decompress the client package to a folder, where you can see the following folders:

```
bin/
conf/
lib/
plugins/
```

3. Edit the following parameters in the `odps_config.ini` file in the `conf` folder:

```
project_name=
access_id=*****
access_key=*****
```

```
end_point= <MaxCompute service address>
```

**Note:**

- Set `access_id` and `access_key` to the AccessKey ID and AccessKey Secret of your cloud account.
- If you frequently use a project, enter the project name after `project_name=`. Then, you do not need to run the `use project_name;` command each time you log on to the client.

4. After the modifications, run the `odps` file in the `bin` directory (`./bin/odpscmd` in a Linux system or `./bin/odpscmd.bat` in a Windows system). Then, you are ready to run SQL statements. An example is as follows:

```
create table tbl1(id bigint);
insert overwrite table tbl1 select count(*) from tbl1;
select 'welcome to MaxCompute!' from tbl1;
```

### 1.17.1.3 Configuration description

This topic describes some configurations and corresponding parameters of the MaxCompute client.

Help

Run the following command to view the help information about the client:

```
odps@ >./bin/odpscmd -h;
```

**Note:**

You can also enter `h;` or `help;` (case insensitive) in the interaction mode.

Startup parameters

Run the following command to specify a number of startup parameters:

```
Usage: odpscmd [OPTION]...
where options include:
 --help (-h)for help
 --project= use project
 --endpoint= set endpoint
 -u -p user name and password
 -k will skip beginning queries and start from specified position
 -r set retry times
 -f <"file_path;"> execute command in file
 -e <"command;[command;]..."> execute command, include sql command
 -C will display job counters
```

**Example:** (`-f` is used as an example)

1. Prepare a local script file named script.txt. The file is stored in D:/. Its contents are as follows:

```
DROP TABLE IF EXISTS test_table_mj;
CREATE TABLE test_table_mj (id string, name string);
DROP TABLE test_table_mj;
```

2. Run the following command:

```
odpscmd\bin>odpscmd -f d:/script.txt;
```

3. The command output is as follows:

```
ID = 20170528122432906gux77io3
Log view:
http://webconsole.odps.aliyun-inc.com:8080/logview/?h=http://
service-corp.odps.aliyun-inc.com/api&podps_public_dev&i2017052
8122432906gux77io3&tokenRnlsSzJoL242YW43dFFic1dmb1ZWZzFxQ1R
FPSxPRFBTX09CTzoxMDcwMDI1NjI3ODA1 NjI5LDE0MzM0MjA2NzMseyJTdGF0ZW
1lbnQiOlt7IkFjdGlvbiI6WyJvZHBzOlJlYWQiXSwiRWZmZWNOIjoiQWxs
ciLCJSZXNvdXJjZSI6WyJhY3M6b2RwczoQOnB yb2ply3RzL29kcHNfcHVibGljX2Rld
i9pbnN0YW5jZXMvMjAxNTA1MjgxMjI0MzI5MDZndXg3N2lvMyJdfV0sIlZlc
nNpb24iOiIxIn0=
OK
ID = 20170528122439318gcmkk6u1
Log view:
http://webconsole.odps.aliyun-inc.com:8080/logview/?h=http://
service-corp.odps.aliyun-inc.com/api&podps_public_dev&i2017052
8122439318gcmkk6u1&tokenSt0RXdlV0M5YjZET2I1MnJuUFkzWDN1aWp
zPSxPRFBTX09CTzoxMDcwMDI1NjI3ODA1NjI5LDE0MzM0MjA2ODAsyJTdGF
0ZW1lbnQiOlt7IkFjdGlvbiI6WyJvZHBzOlJlYWQiXSwiRWZmZWNOIjoiQWx
sb3ciLCJSZXNvdXJjZSI6WyJhY3M6b2RwczoQOnB yb2ply3RzL29kcHNfcHV
ibGljX2Rld i9pbnN0YW5jZXMvMjAxNTA1MjgxMjI0MzI5MDZndXg3N2lvMyJdfV0sIlZlc
nNpb24iOiIxIn0=
OK
ID = 20170528122440389g98cmlmf
Log view:
http://webconsole.odps.aliyun-inc.com:8080/logview/?h=http://
service-corp.odps.aliyun-inc.com/api&podps_public_dev&i2017052
8122440389g98cmlmf&tokenNWlwL0EvQThxUXhzcTRERDc5NFg0b2IxZ3Q
wPSxPRFBTX09CTzoxMDcwMDI1NjI3OD A1NjI5LDE0MzM0MjA2ODAsyJTdGF0
ZW1lbnQiOlt7IkFjdGlvbiI6WyJvZHBzOlJlYWQiXSwiRWZmZWNOIjoiQWxs
b3ciLCJSZXNvdXJjZSI6WyJhY3M6b2RwczoQOnB yb2ply3RzL29kcHNfcHV
ibGljX2Rld i9pbnN0YW5jZXMvMjAxNTA1MjgxMjI0MzI5MDZndXg3N2lvMyJdfV0sIlZlc
nNpb24iOiIxIn0=
OK
```

Interactive mode

- Directly run the client, and you will enter the interaction mode. The following information is displayed:

```
[admin: ~]$odpscmd
Aliyun ODPS Command Line Tool
Version 1.0
@Copyright 2012 Alibaba Cloud Computing Co., Ltd. All rights reserved
.
```

```
XXX@ XXX> INSERT OVERWRITE TABLE DUAL SELECT * FROM DUAL;
```

**Note:**

The first XXX indicates the identifier of MaxCompute, and the second XXX indicates the project to which you belong. Enter a command at the cursor (terminated by a semicolon), and press Enter to run it.

## Command output

The output of a SQL statement is in either HumanReadable (default) or MachineReadable format. If you use the -M parameter when running odpscmd, the output format is CSV.

**Note:**

This function currently applies only to SELECT and READ statements, and takes effect when reading data.

## Continue run

When you run odpscmd in -e or -f mode and want to start with an intermediate statement among a few statements, you can use the -k parameter. This parameter indicates that the execution starts from the specified statement while the preceding statements are skipped. If the parameter value is equal to or smaller than 0, execution starts from the first statement. A statement terminated by a semi-colon is considered a valid statement. At runtime, the process indicates the specific statement that is running successfully or fails.

For example, the `/tmp/dual.sql` file includes the following three SQL statements:

```
drop table dual;
create table dual (dummy string);
```

```
insert overwrite table dual select count(*) from dual;
```

**You can run the following command to ignore the first two statements:**

```
odpscmd-k 3 -f dual.sql
```

Obtain information about the current logon user

**Run the following command to obtain the cloud account of the current logon user and the endpoint that is used:**

```
whoami
```

**Example:**

```
odps@ hiveut>whoami; Name: odpstest@aliyun.com ID: 1090142773636588
End_Point: <MaxCompute service address> Project: lijunsecuritytest
```

Exit

**Command syntax:**

```
odps@ > quit;
```

**or**

```
q;
```

Configure a job priority

**Command syntax:**

```
Admin@ > ./bin/odpscmd --instance-priority=<PRIORITY>;
```

**Configuration file: odps\_config.ini**

```
instance_priority=<PRIORITY>
```



**Notice:**

- The value range of <PRIORITY> is 0–9, where 0 indicates the highest priority and 9 the lowest priority.
- The priority setting in the configuration file applies to all instances submitted in CLT.
- The priority value is 9 by default.

## DryRun mode

**In the DryRun mode, MaxCompute parses a SQL statement to check if the syntax is correct and generates an execution plan. MaxCompute does not submit a distributed job. Command syntax:**

```
./bin/odpscmd -y
```

## SQL reliability

**If an exception is returned during the execution of SQL statements such as INSERT or CREATE TABLE AS, the client tries to automatically recover data and metadata to the pre-execution state based on known information.**

- **Data that is overwritten by INSERT OVERWRITE during QUERY execution is recovered from the temporary backup directory to the original directory.**
- **Data that is generated by INSERT INTO during QUERY execution is removed.**
- **Tables created during QUERY execution and dynamically generated partition information are removed.**

**If MaxCompute fails to recover data, it returns a specific error code. The error code alerts you that further attempts can make some data unrecoverable. The error code is as follows:**

```
ODPS-
0110999: Critical! Internal error happened in commit operation and
rollback failed, possible breach of atomicity
```

## 1.17.2 Eclipse development plugin

### 1.17.2.1 Install Eclipse

**This topic describes how to install the Eclipse development plugin.**

#### Context

**MaxCompute provides the Eclipse development plugin to help you easily use the Java SDKs for MapReduce and UDFs in your development work. This plugin can simulate the running process of MapReduce or UDFs. It provides local debugging methods and simple template generation features.**



**Note:**

Compared with the local running mode of MapReduce, the Eclipse plugin does not support data synchronization with MaxCompute. You need to manually copy the data you need to the warehouse directory of the Eclipse plugin.

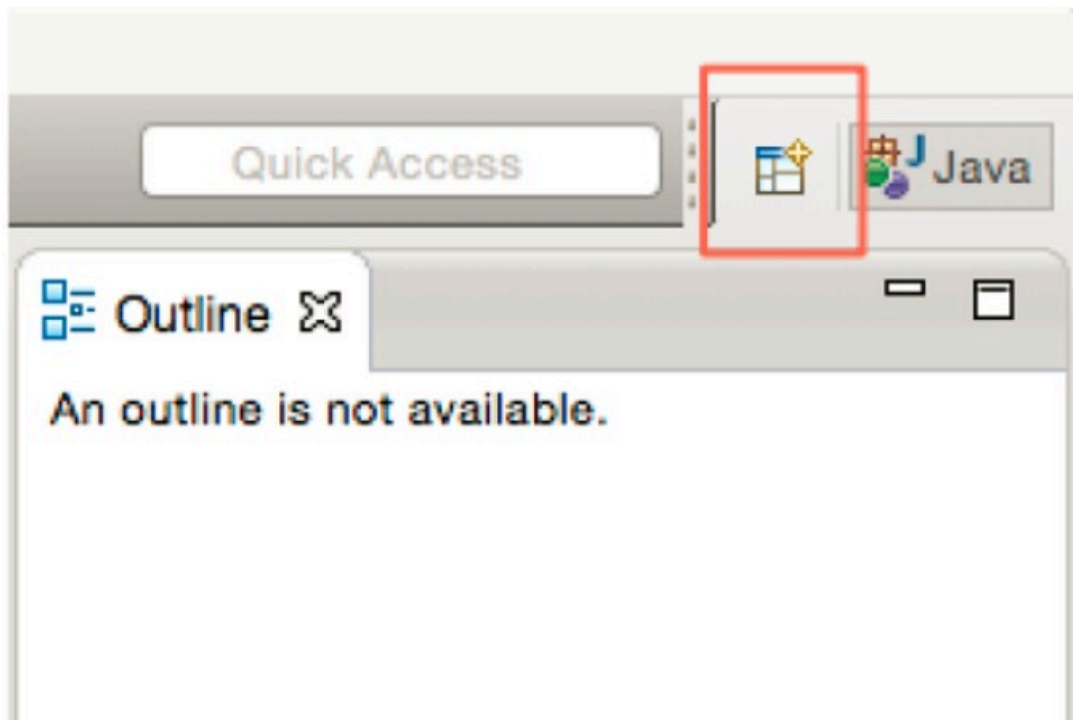
## Procedure

1. Decompress the Eclipse package. You will see the following JAR file.

`odps-eclipse-plugin-bundle-0.15.0.jar`

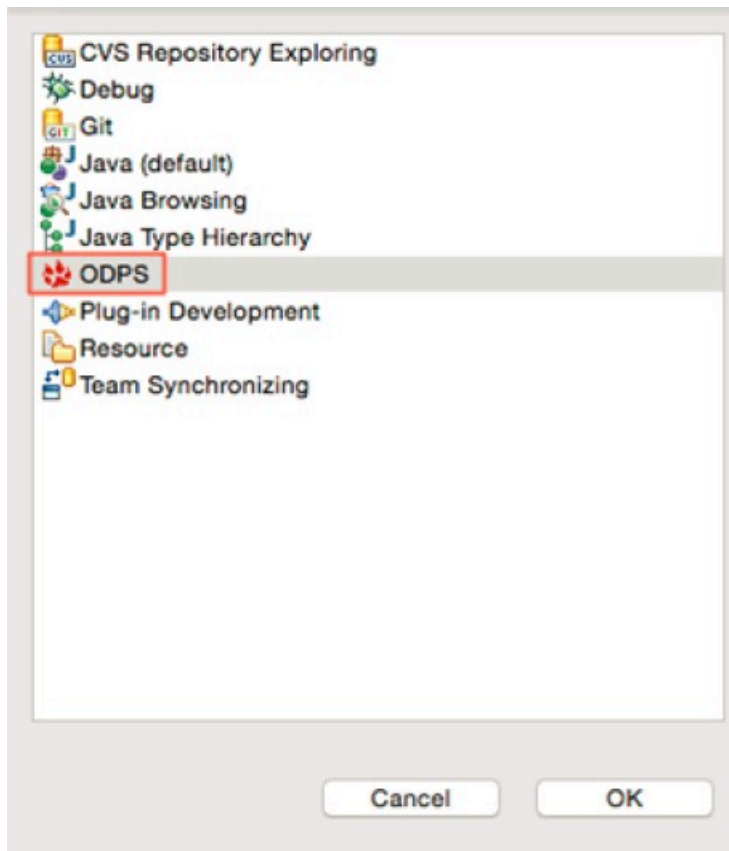
2. Place the plugin in the `plugins` subdirectory of the Eclipse installation directory.
3. Start Eclipse, and click Open Perspective in the upper-right corner, as shown in the following figure.

Figure 1-17: Eclipse installation 1



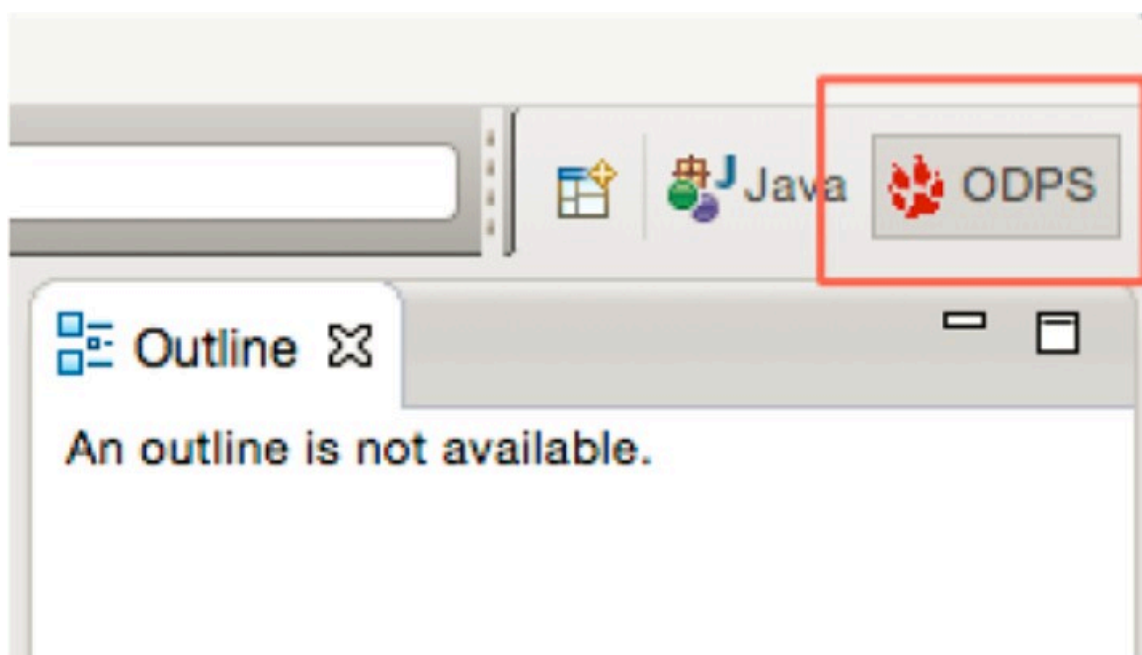
4. Click ODPS in the dialog box that appears, as shown in the following figure.

Figure 1-18: Eclipse installation 2



5. Click ODPS. A dialog box appears, as shown in the following figure.

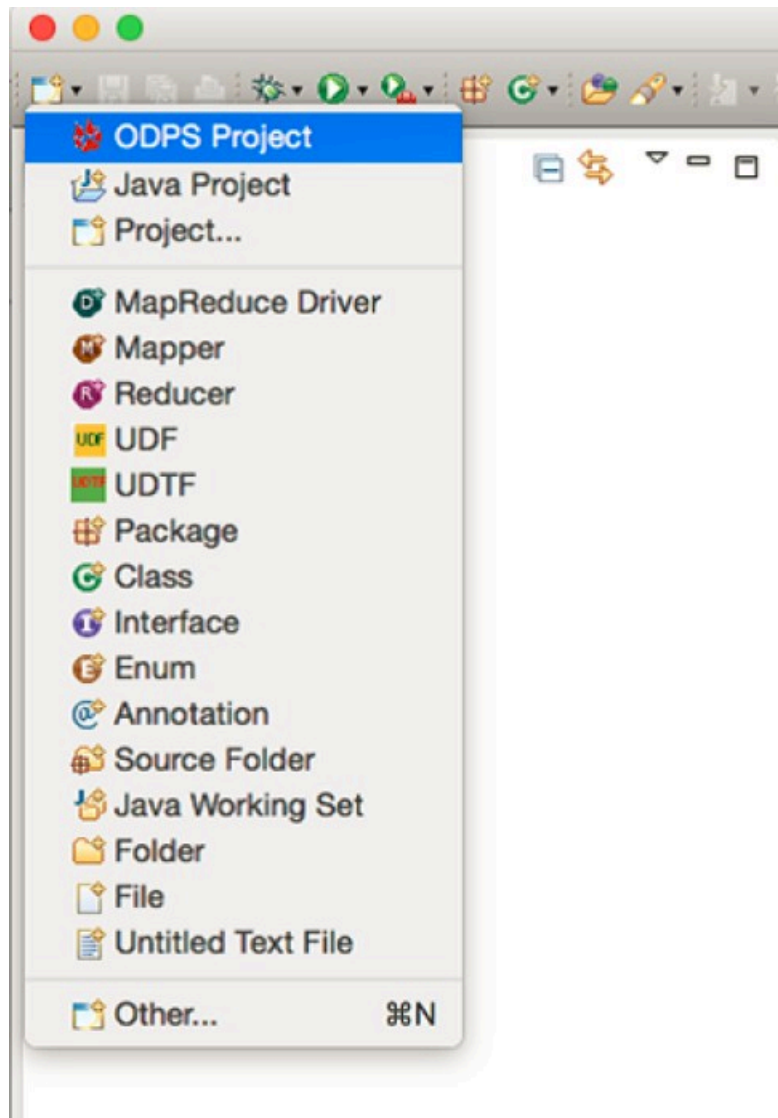
Figure 1-19: Eclipse installation 3





6. Click ODPS, and then click OK. An ODPS icon is displayed in the upper-right corner, indicating that the plugin has taken effect, as shown in the following figure.

Figure 1-20: Eclipse installation 4



## 1.17.2.2 Create a project

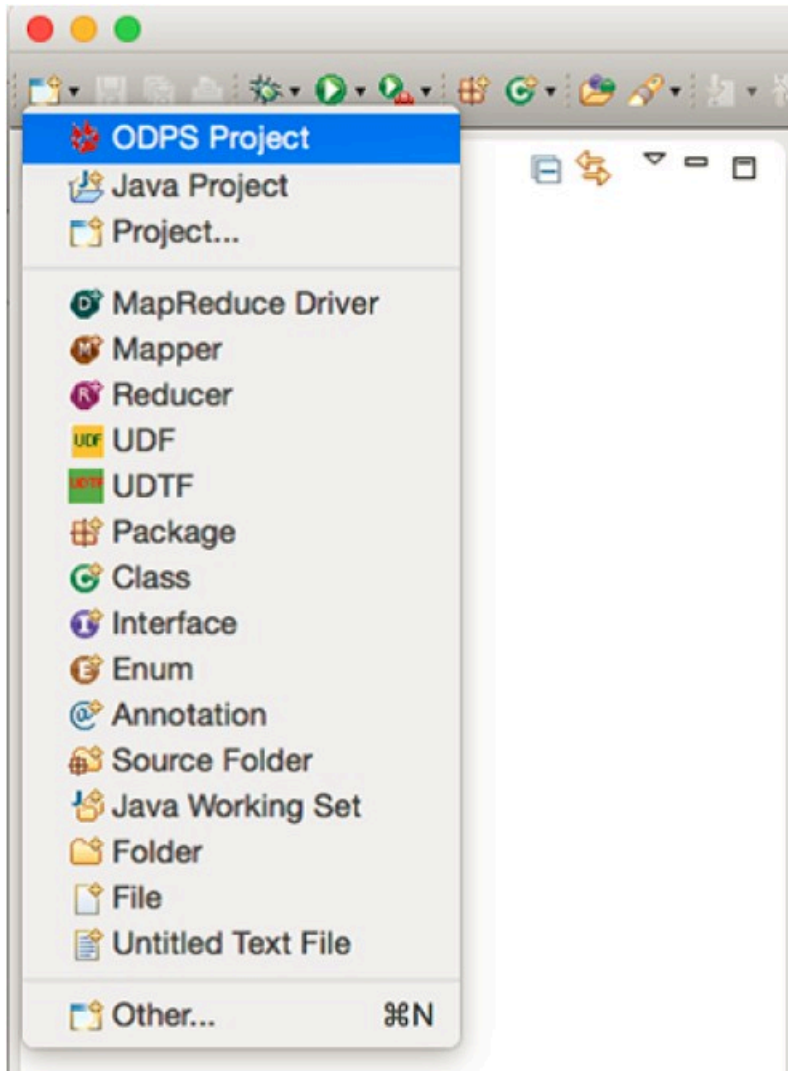
### 1.17.2.2.1 Method 1

This topic describes the first method to create a project in the Eclipse development plugin.

#### Procedure

1. Start Eclipse. Choose File > New > Project > ODPS > ODPS Project in the upper-left corner to create a project, as shown in the following figure.

Figure 1-21: Step 1

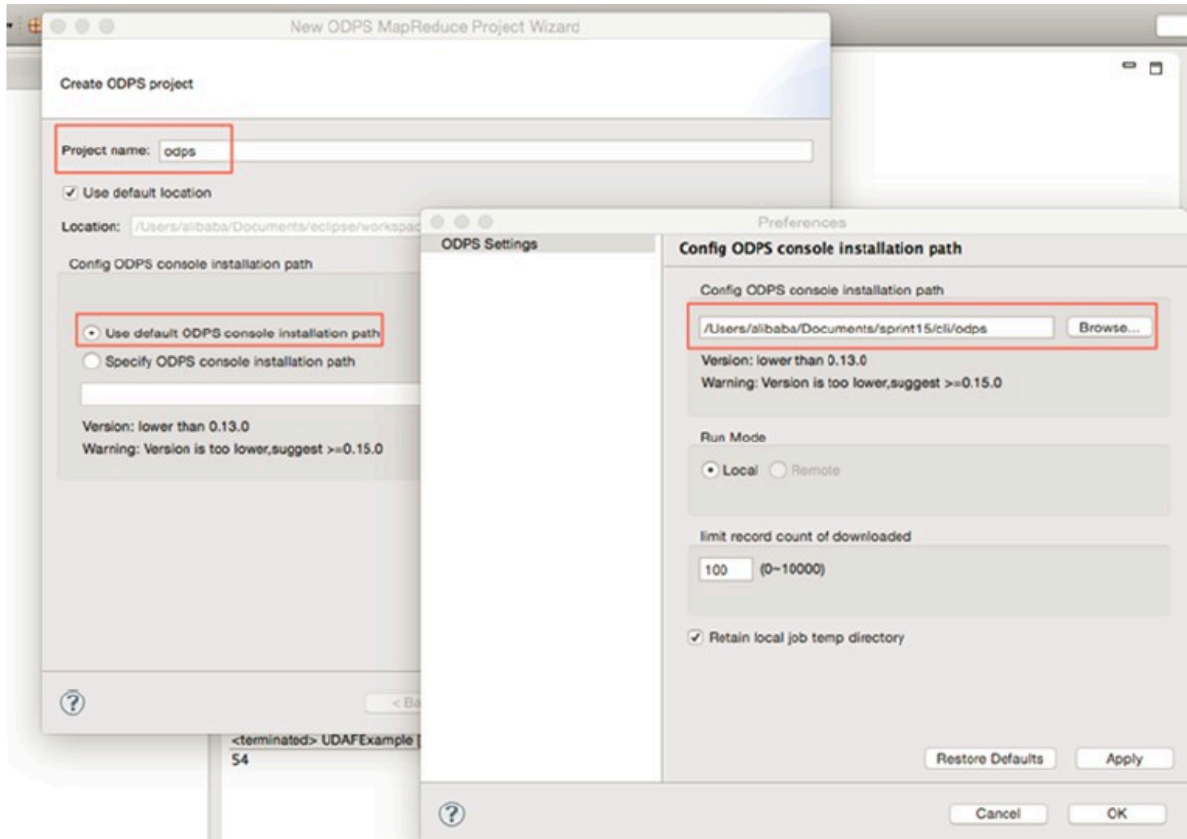


**Note:**

**In this example, the project name is ODPS.**

2. After the ODPS project is created, a dialog box is displayed, as shown in the following figure. Set Project name, select the path of the MaxCompute client, and then click Finish.

Figure 1-22: Step 2

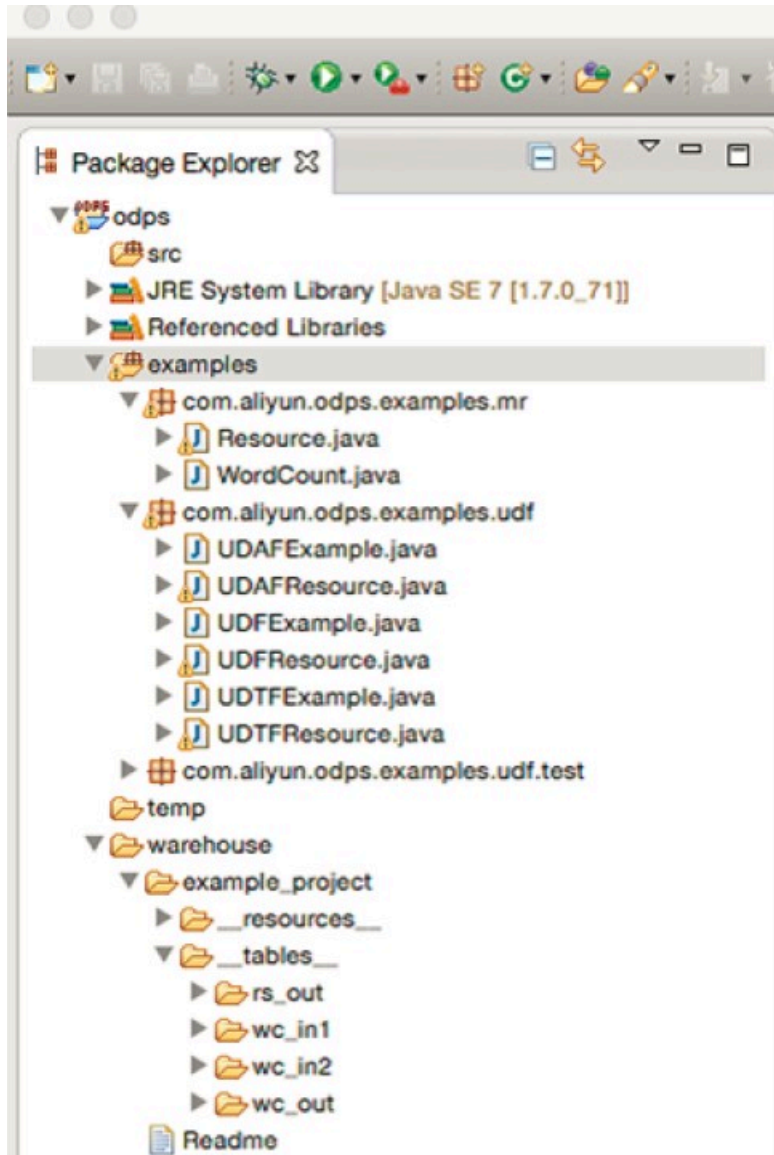


**Note:**

The client must be installed in advance.

3. After creating a project, you can see the directory structure on the left-side Package Explorer pane, as shown in the following figure.

Figure 1-23: Step 3



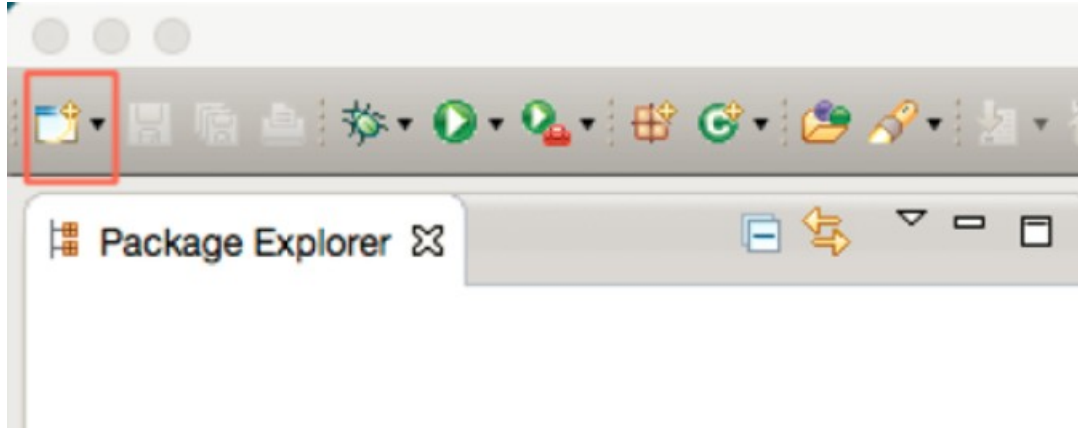
#### 1.17.2.2.2 Method 2

This topic describes the second method to create a project in the Eclipse development plugin.

#### Procedure

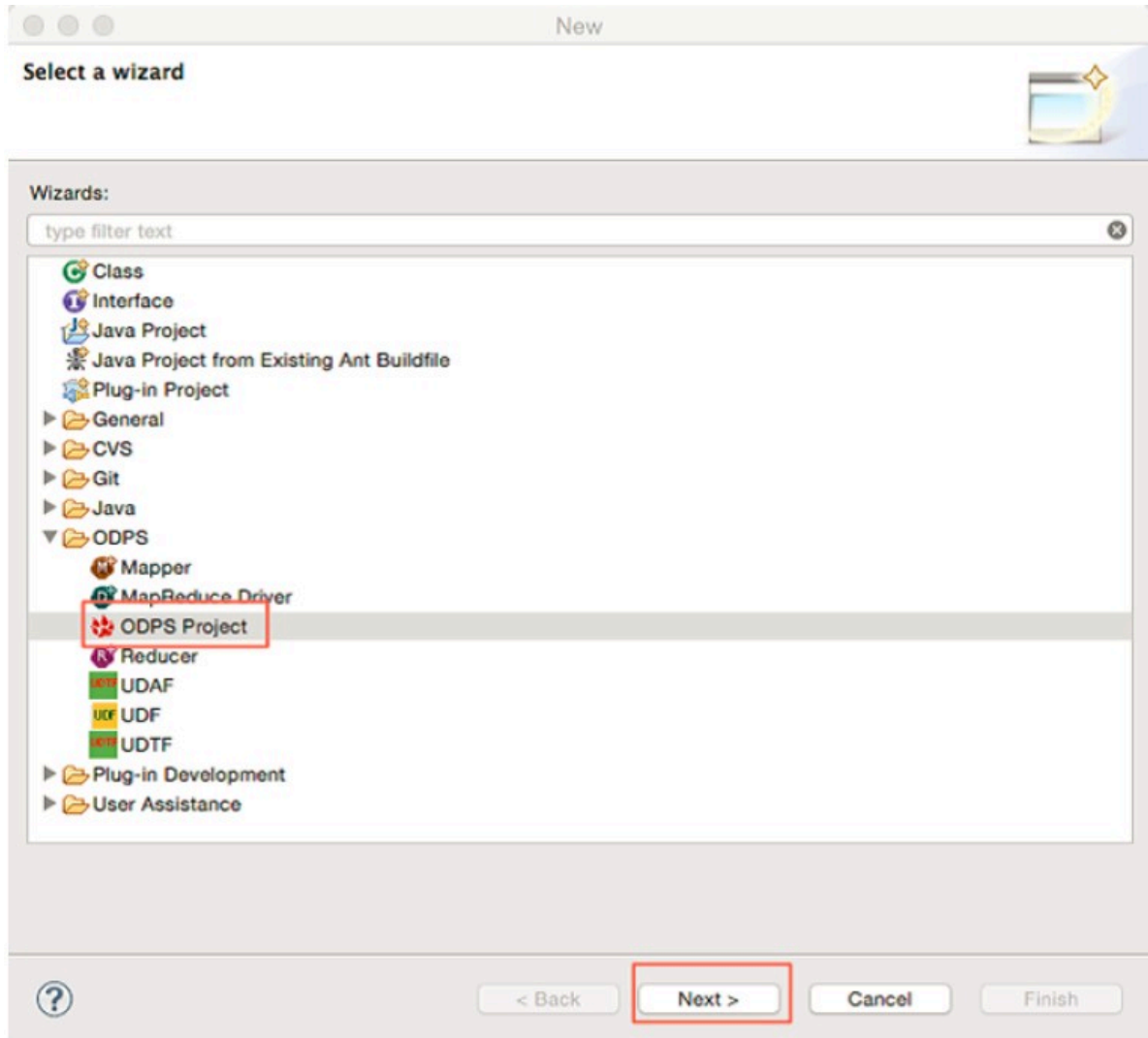
1. Start Eclipse. Click New in the upper-left corner, as shown in the following figure.

Figure 1-24: Step 1



2. In the dialog box that appears, select ODPS Project and click Next, as shown in the following figure.

Figure 1-25: Step 2



**Note:**

In this example, the project name is ODPS.

3. The subsequent steps are the same as those in method 1. After installing the Eclipse plugin, you can use it to compile MapReduce or UDF programs.



**Note:**

For a MapReduce running example, see [MapReduce running example](#). For a UDF development and running example, see [UDF development and running example](#).

## 1.17.2.3 MapReduce running example

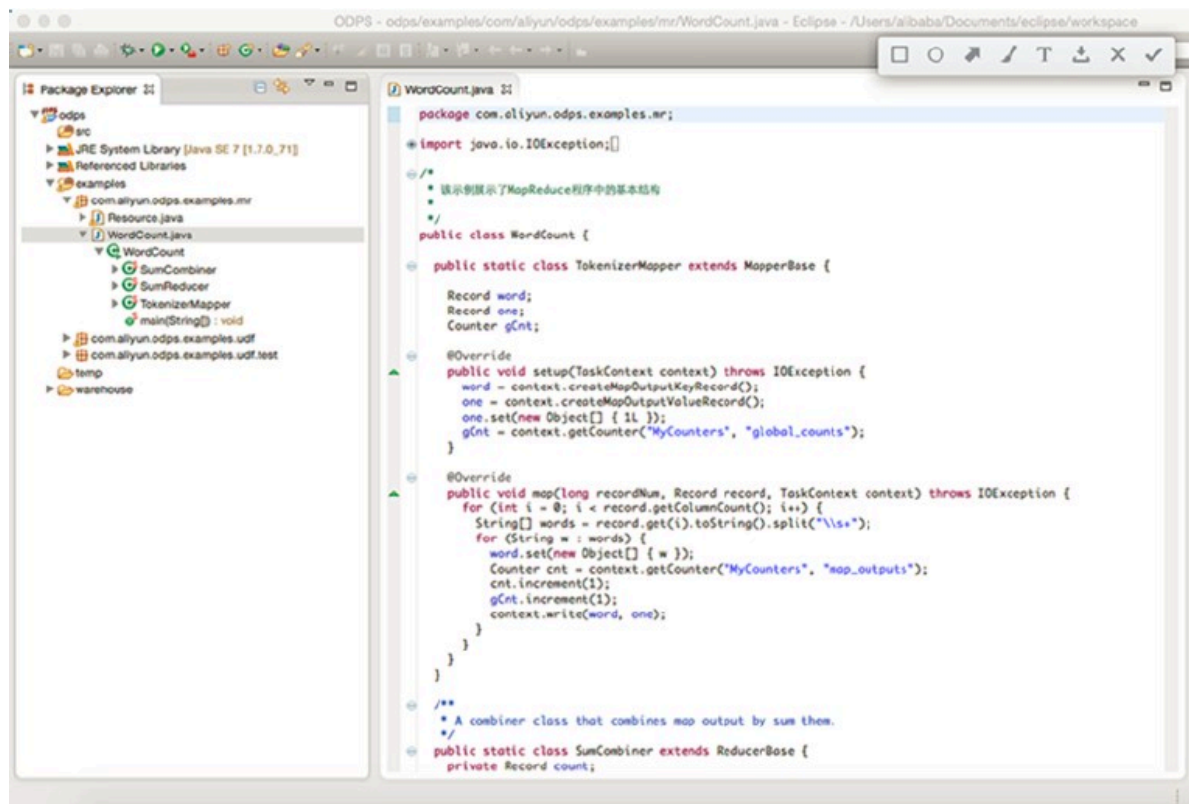
### 1.17.2.3.1 Quickly run a WordCount example

This topic describes how to use MapReduce to quickly run a WordCount example in the Eclipse plugin.

#### Procedure

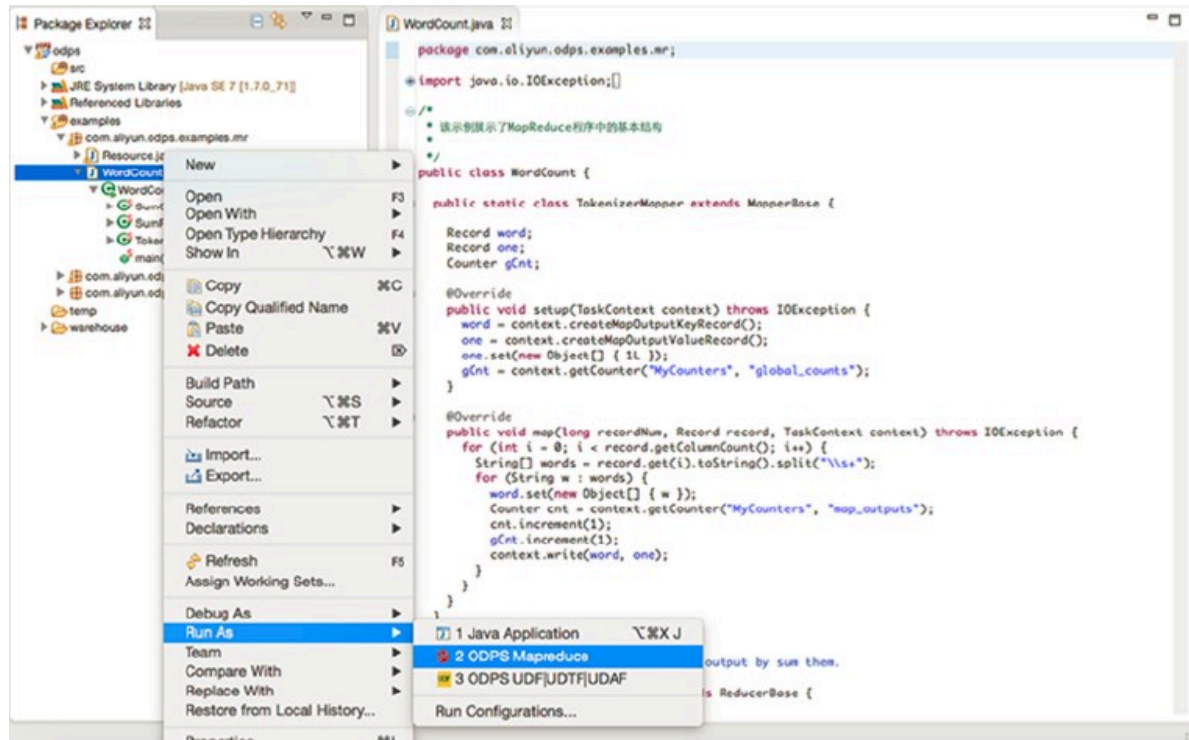
1. Select a WordCount example in MaxCompute, as shown in the following figure.

Figure 1-26: WordCount example



## 2. Right-click WordCount.java and choose Run AsODPS MapReduce from the shortcut menu, as shown in the following figure.

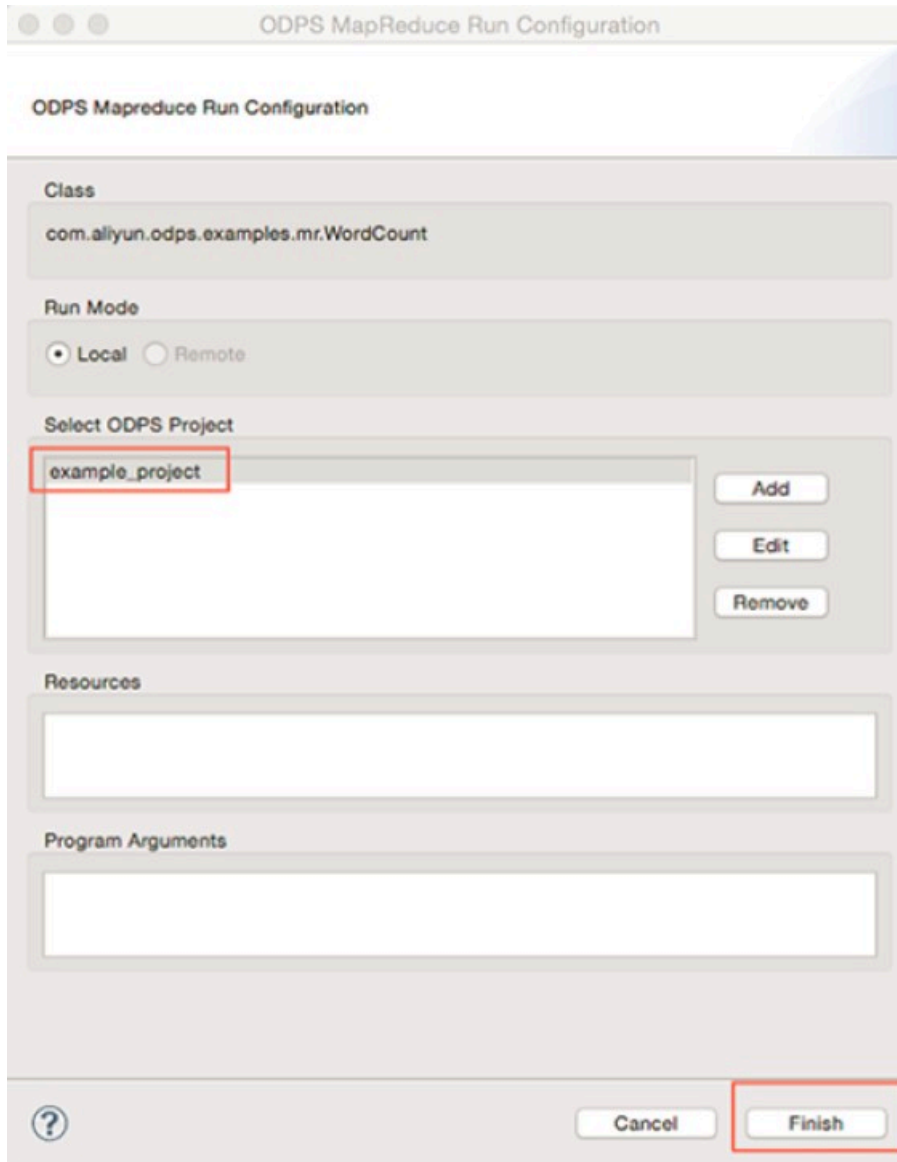
Figure 1-27: Run the WordCount example





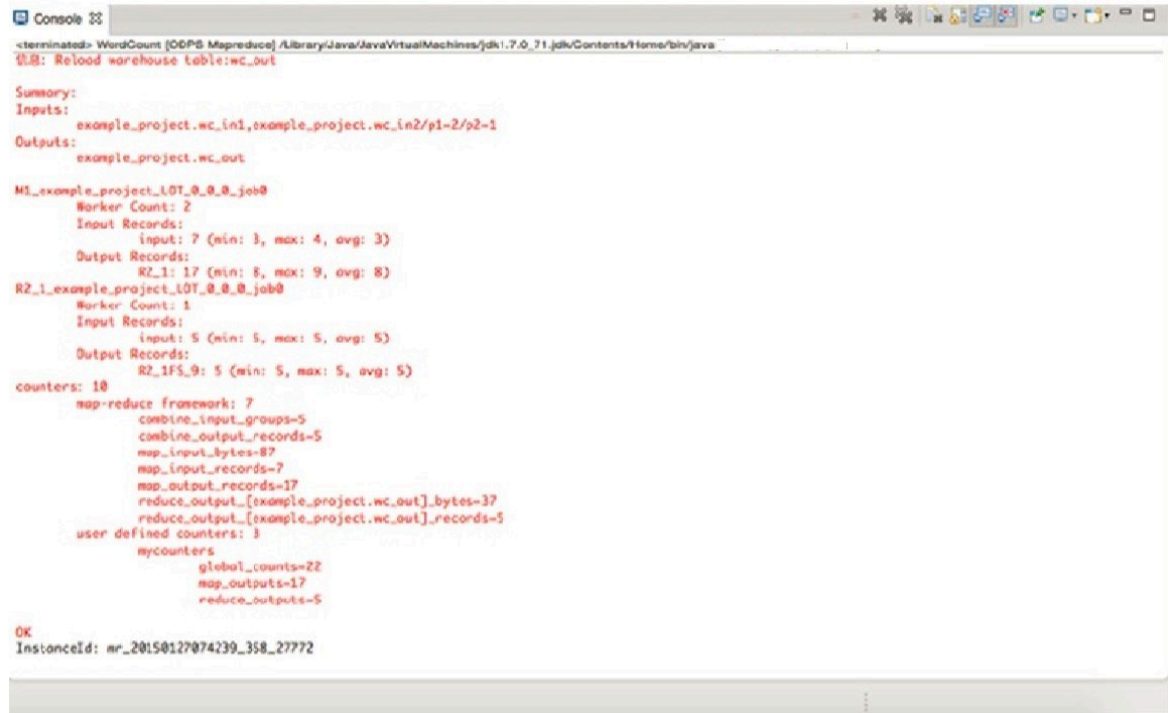
3. In the dialog box that appears, select `example_project` and click OK, as shown in the following figure.

Figure 1-28: Run the WordCount example



4. The results of the operation are displayed after the operation is executed, as shown in the following figure.

Figure 1-29: Execution result of the WordCount example



```
<terminated> WordCount [ODPS Mapreduce] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java
信息: Reload warehouse table:wc_out

Summary:
Inputs:
 example_project.wc_in1,example_project.wc_in2/p1-2/p2-1
Outputs:
 example_project.wc_out

M1_example_project_L0T_0_0_0_job0
 Worker Count: 2
 Input Records:
 input: 7 (min: 1, max: 4, avg: 3)
 Output Records:
 RZ_1: 17 (min: 8, max: 9, avg: 8)
R2_1_example_project_L0T_0_0_0_job0
 Worker Count: 1
 Input Records:
 input: 5 (min: 5, max: 5, avg: 5)
 Output Records:
 RZ_1FS_9: 5 (min: 5, max: 5, avg: 5)
counters: 10
 map-reduce framework: 7
 combine_input_groups=5
 combine_output_records=5
 map_input_bytes=87
 map_input_records=7
 map_output_records=17
 reduce_output_[example_project.wc_out].bytes=37
 reduce_output_[example_project.wc_out].records=5
 user defined counters: 3
 mycounters
 global_counts=22
 map_outputs=17
 reduce_outputs=5

OK
InstanceId: mr_20150127074239_358_27772
```

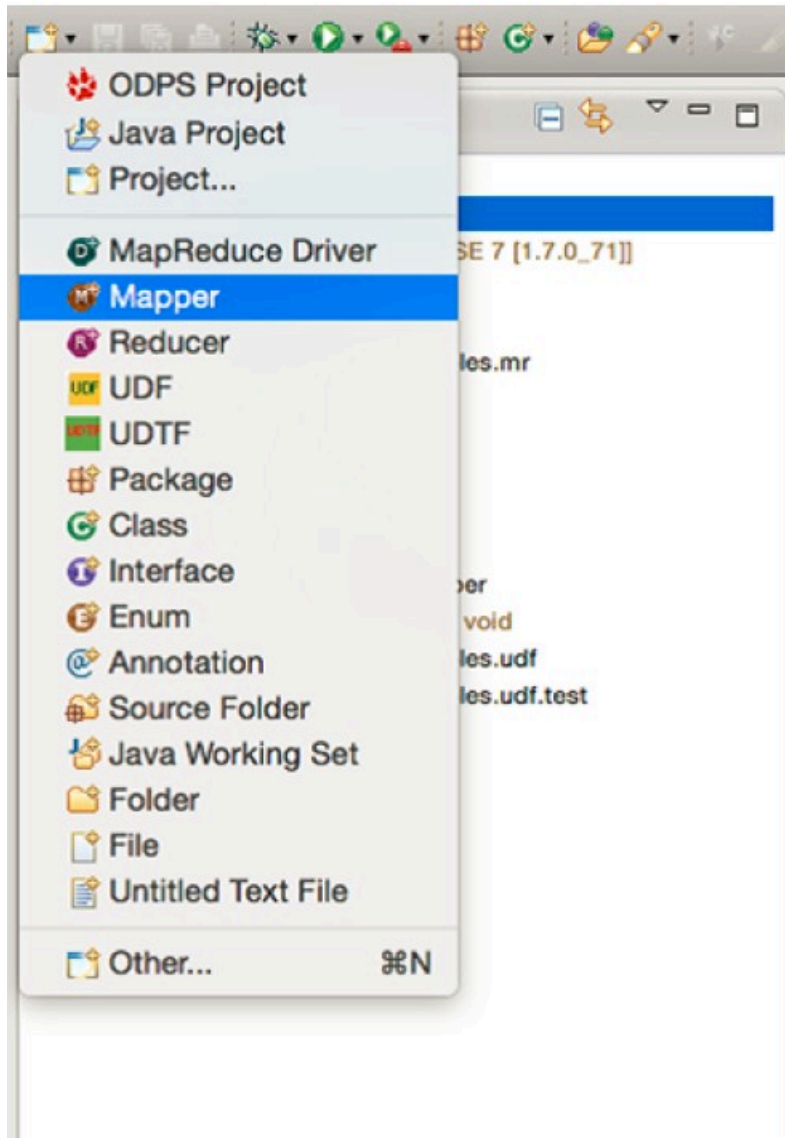
### 1.17.2.3.2 Run a custom MapReduce program

This topic provides an example on how to run a custom MapReduce program in the Eclipse plugin.

#### Procedure

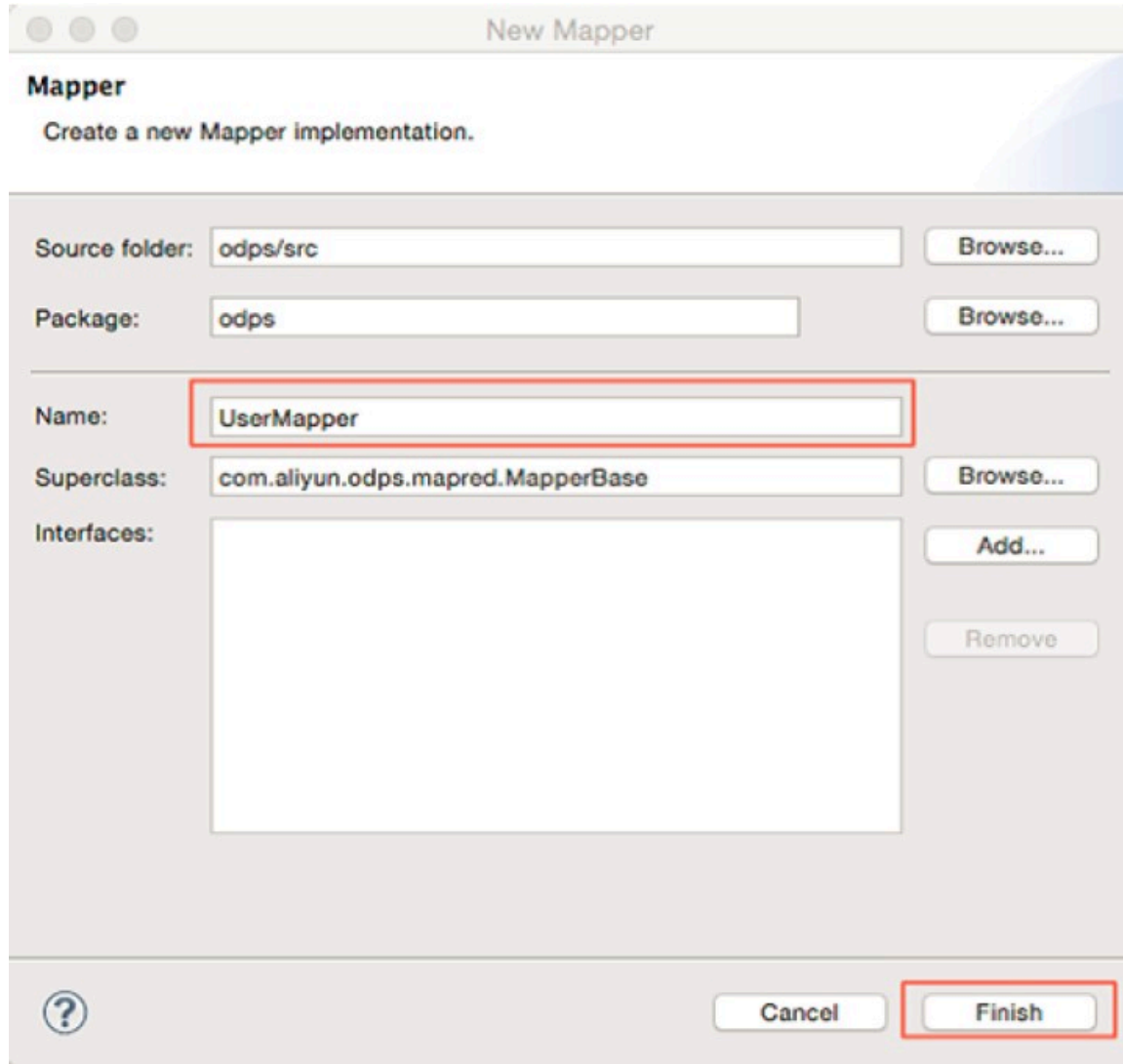
1. In Eclipse, right-click the src directory, and choose New > Mapper from the shortcut menu, as shown in the following figure.

Figure 1-30: Step 1



2. **Select Mapper.** A dialog box is displayed, as shown in the following figure. Enter the name of the Mapper class, and click Finish.

Figure 1-31: Step 2



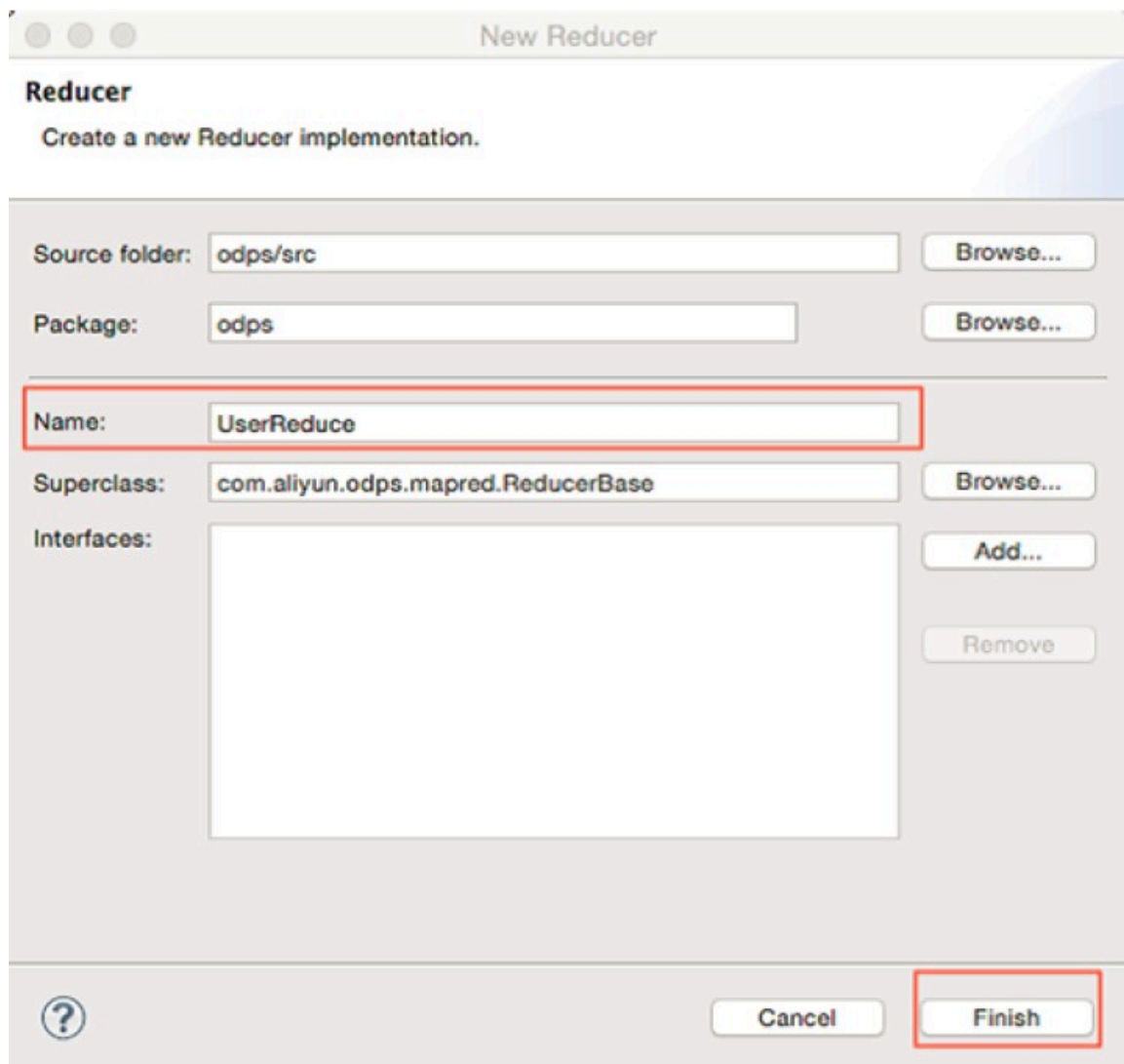
3. **On the left-side Package Explorer, the UserReducer.java file is generated in the src directory. The file contains a Mapper class template. The package name is odps by default. The template content is as follows.**

```
package odps;
import java.io.IOException;
import com.aliyun.odps.counter.Counter; import com.aliyun.odps.data.
Record;
import com.aliyun.odps.mapred.MapperBase;
public class UserMapper extends MapperBase {
Record word; Record one; Counter gCnt;
@Override
public void setup(TaskContext context) throws IOException {
word = context.createMapOutputKeyRecord(); one = context.createMapO
utputValueRecord(); one.set(new Object[] { 1L });
gCnt = context.getCounter("MyCounters", "global_counts");
```

```
}
@Override
public void map(long recordNum, Record record, TaskContext context)
throws IOException {
 for (int i = 0; i < record.getColumnCount(); i++) { String[] words
 = record.get(i).toString().split("\\s+"); for (String w : words) {
 word.set(new Object[] { w });
 Counter cnt = context.getCounter("MyCounters", "map_outputs"); cnt.
 increment(1);
 gCnt.increment(1); context.write(word, one);
 }
}
}
}
@Override
public void cleanup(TaskContext context) throws IOException {
}
}
```

4. In Eclipse, right-click the src directory, and choose New > Reduce from the shortcut menu, as shown in the following figure.

Figure 1-32: Reducer



5. In the dialog box that appears, enter the name of the Reduce class and click **Finish**.



**Note:**

**This example uses UserReducer.**

6. On the left-side **Package Explorer**, the **UserReducer.java** file is generated in the **src** directory. The file contains a Reduce class template. The package name is **odps** by default. The template content is as follows:

```
package odps;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.ReducerBase;
public class UserReducer extends ReducerBase {
 private Record result; Counter gCnt;
 @Override
 public void setup(TaskContext context) throws IOException { result
 = context.createOutputRecord();
 gCnt = context.getCounter("MyCounters", "global_counts");
 }
 @Override
 public void reduce(Record key, Iterator<Record> values, TaskContext
 context) throws IOException {
 long count = 0;
 while (values.hasNext()) { Record val = values.next(); count += (
 Long) val.get(0);
 }
 result.set(0, key.get(0)); result.set(1, count);
 Counter cnt = context.getCounter("MyCounters", "reduce_outputs");
 cnt.increment(1);
 gCnt.increment(1);
 context.write(result);
 }
 @Override
 public void cleanup(TaskContext context) throws IOException {
 }
}
```

7. In Eclipse, right-click the **src** directory, and choose **New > MapReduce Driver** from the shortcut menu.

8. A dialog box is displayed, as shown in the following figure. Set Name, Mapper, and Reducer, and then click Finish.

Figure 1-33: MapReduce Driver

**New MapReduce Driver**

**MapReduce Driver**  
Create a new MapReduce driver.

Source folder:

Package:

---

Name:

Superclass:

Interfaces:

---

Mapper:

Reducer:

9. On the left-side Package Explorer, the MyDriver.java file is generated in the src directory. The file contains a MapReduce Driver template. The package name is odps by default. The template content is as follows:

```
package odps;
import com.aliyun.odps.OdpsException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.examples.mr.WordCount.SumCombiner;
import com.aliyun.odps.examples.mr.WordCount.SumReducer;
import com.aliyun.odps.examples.mr.WordCount.TokenizerMapper;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
```

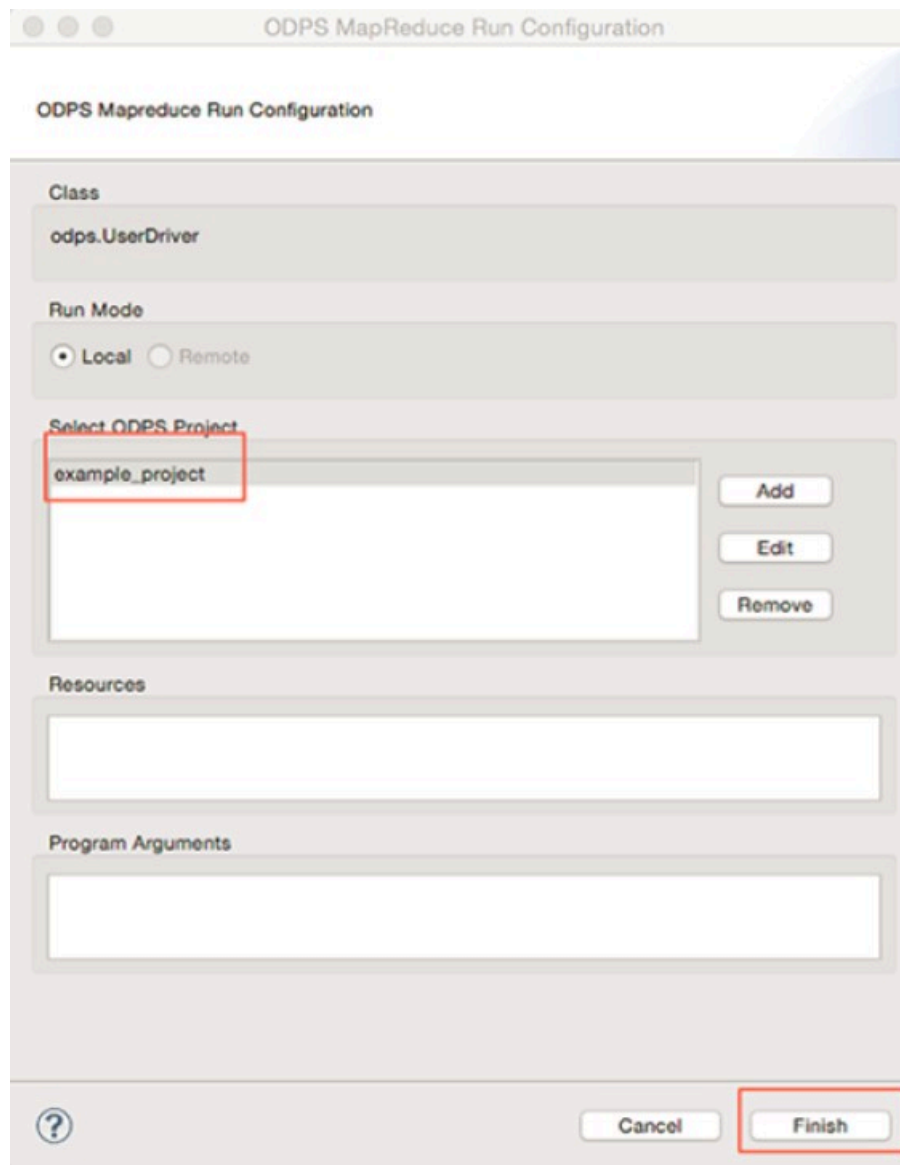
```
import com.aliyun.odps.mapred.utils.SchemaUtils;
public class UserDriver {
 public static void main(String[] args) throws OdpsException {
 JobConf job = new JobConf();
 job.setMapperClass(TokenizerMapper.class);
 job.setCombinerClass(SumCombiner.class);
 job.setReducerClass(SumReducer.class);
 job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
 job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
 InputUtils.addTable(
 TableInfo.builder().tableName("wc_in1").cols(new String[] { "col2",
 "col3" }).build(), job);
 InputUtils.addTable(TableInfo.builder().tableName("wc_in2").partSpec(
 "p1=2/p2=1").build(), job);
 OutputUtils.addTable(TableInfo.builder().tableName("wc_out").build(
), job);
 RunningJob rj = JobClient.runJob(job); rj.waitForCompletion();
 }
}
```



```
}
```

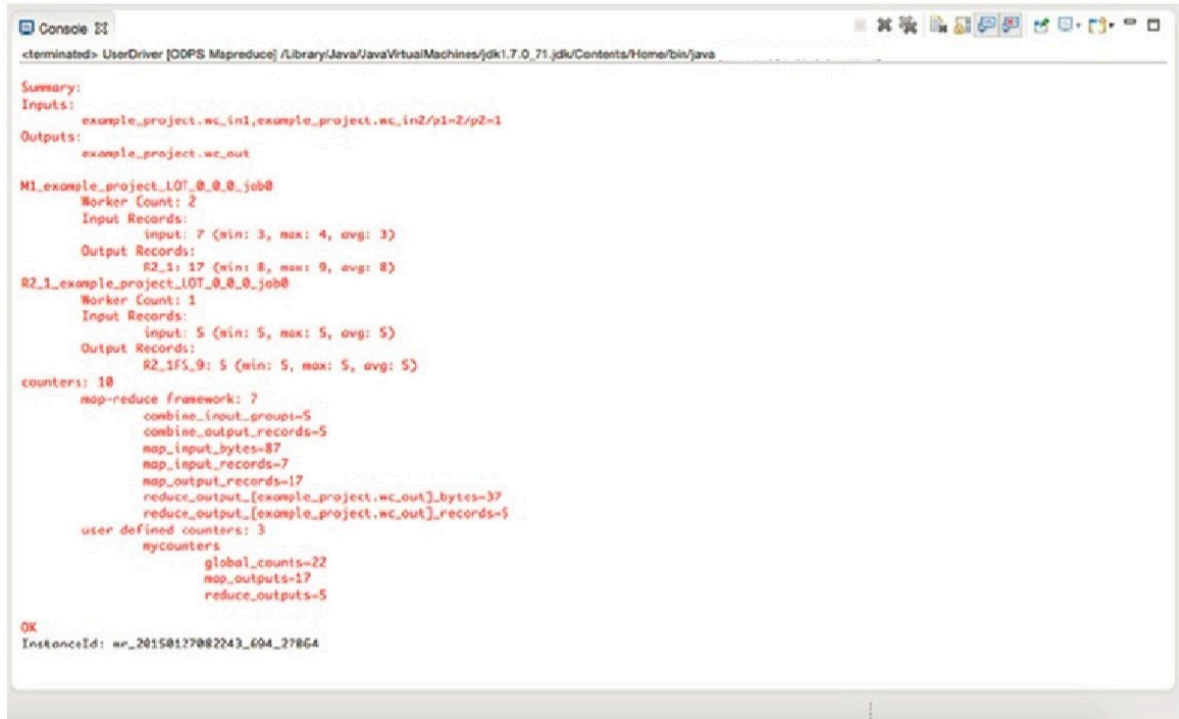
**10 Run the MapReduce program. Right-click `UserDriver.java` and choose **Run As > ODPS MapReduce** from the shortcut menu. In the dialog box that appears, click **OK**. A dialog box is displayed, as shown in the following figure.**

Figure 1-34: ODPS MapReduce Run Configuration



**11 Select example\_project as the MaxCompute project. Click Finish to start running the MapReduce program locally. If the output is as shown in the following figure, the local running is successful.**

Figure 1-35: Console



```

<terminated> UserDriver [OOPS Mapreduce] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java

Summary:
Inputs:
 example_project.wc_in1,example_project.wc_in2/p1=2/p2=1
Outputs:
 example_project.wc_out

M1_example_project_LOT_0_0_0.job0
Worker Count: 2
Input Records:
 input: 7 (min: 3, max: 4, avg: 3)
Output Records:
 R2_1: 17 (min: 8, max: 9, avg: 8)
R2_1_example_project_LOT_0_0_0.job0
Worker Count: 1
Input Records:
 input: 5 (min: 5, max: 5, avg: 5)
Output Records:
 R2_1FS_9: 5 (min: 5, max: 5, avg: 5)

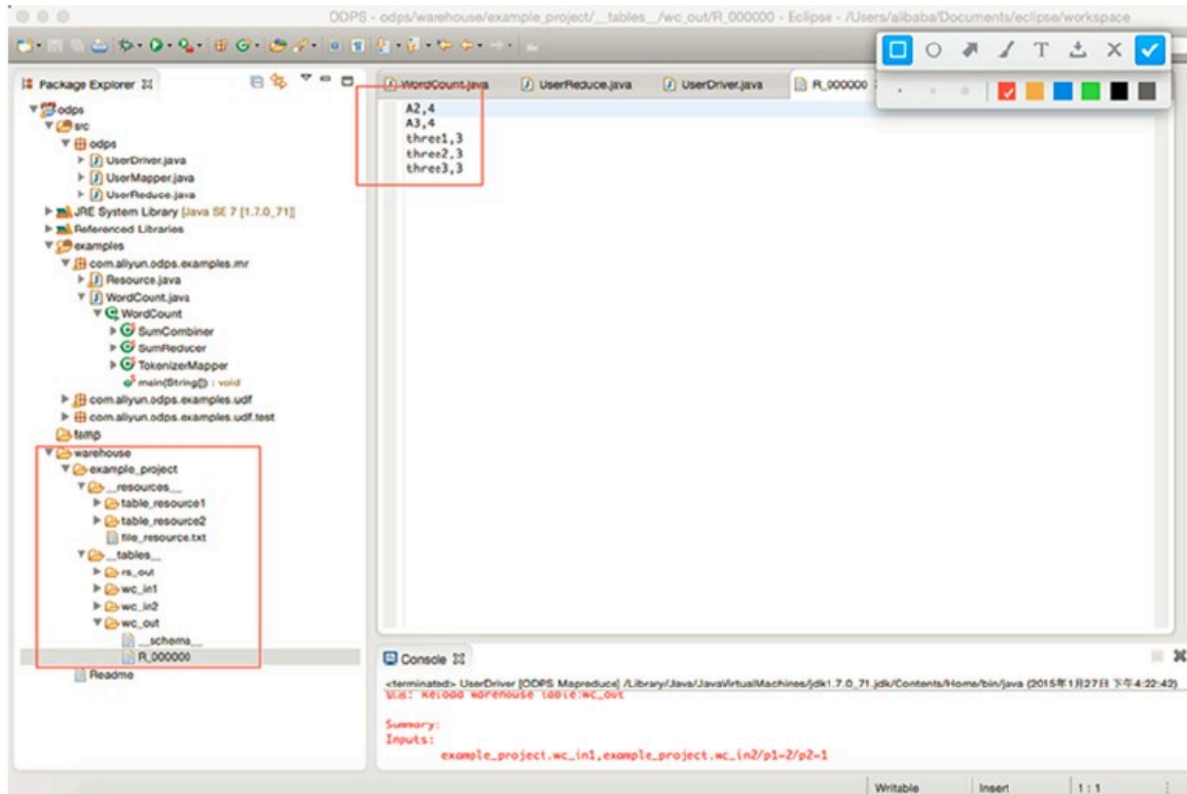
counters: 10
 map-reduce framework: 7
 combine_inout_groups=5
 combine_output_records=5
 map_input_bytes=87
 map_input_records=7
 map_output_records=17
 reduce_output_example_project_wc_out_bytes=37
 reduce_output_example_project_wc_out_records=5
 user defined counters: 3
 mycounters
 global_counts=22
 map_outputs=17
 reduce_outputs=5

OK
InstanceId: wc_20150127082243_604_27864

```

12.The execution result is stored in the warehouse directory. Refresh the ODPS project, as shown in the following figure.

Figure 1-36: Output



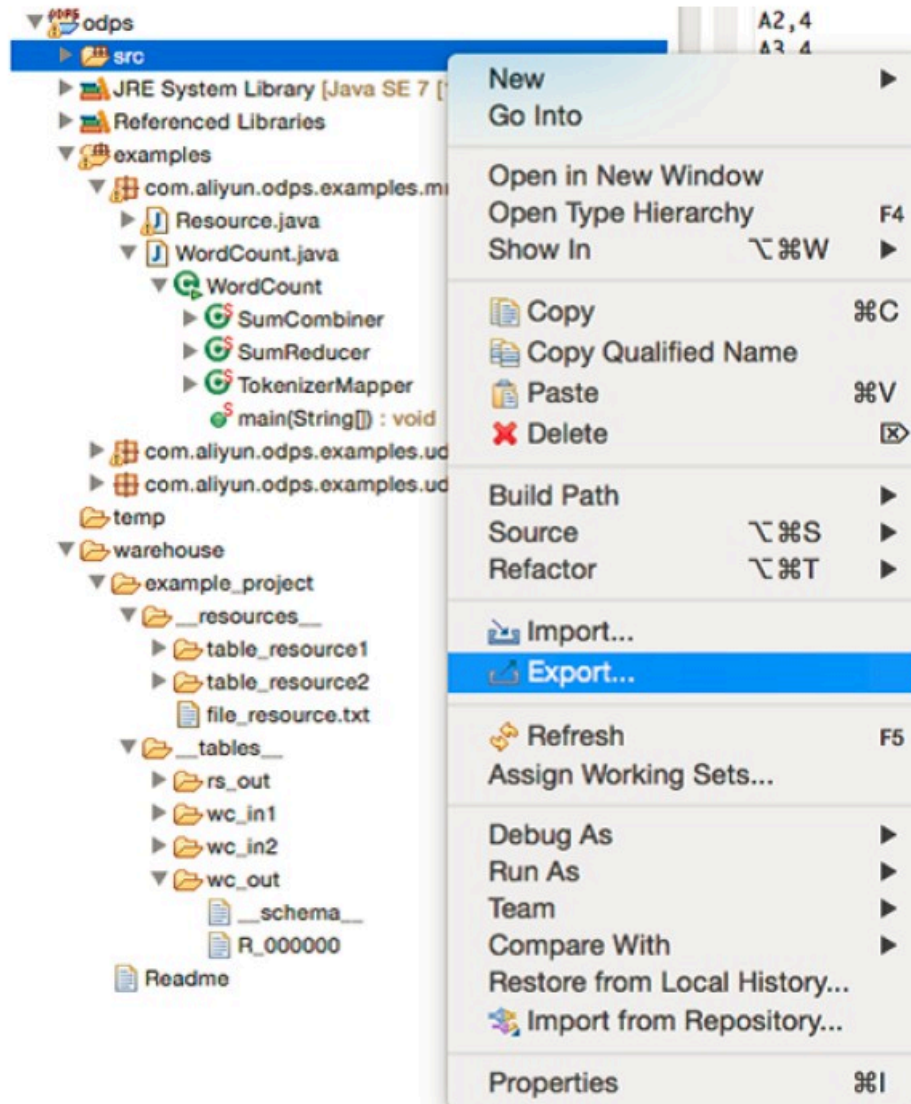
#### Note:

wc\_out is the output directory, and R\_000000 is the result file. After confirming that the result is correct through local debugging, you can use the export function of Eclipse to package the MapReduce program for subsequent use in the distributed environment.

### 13.The export procedure is as follows:

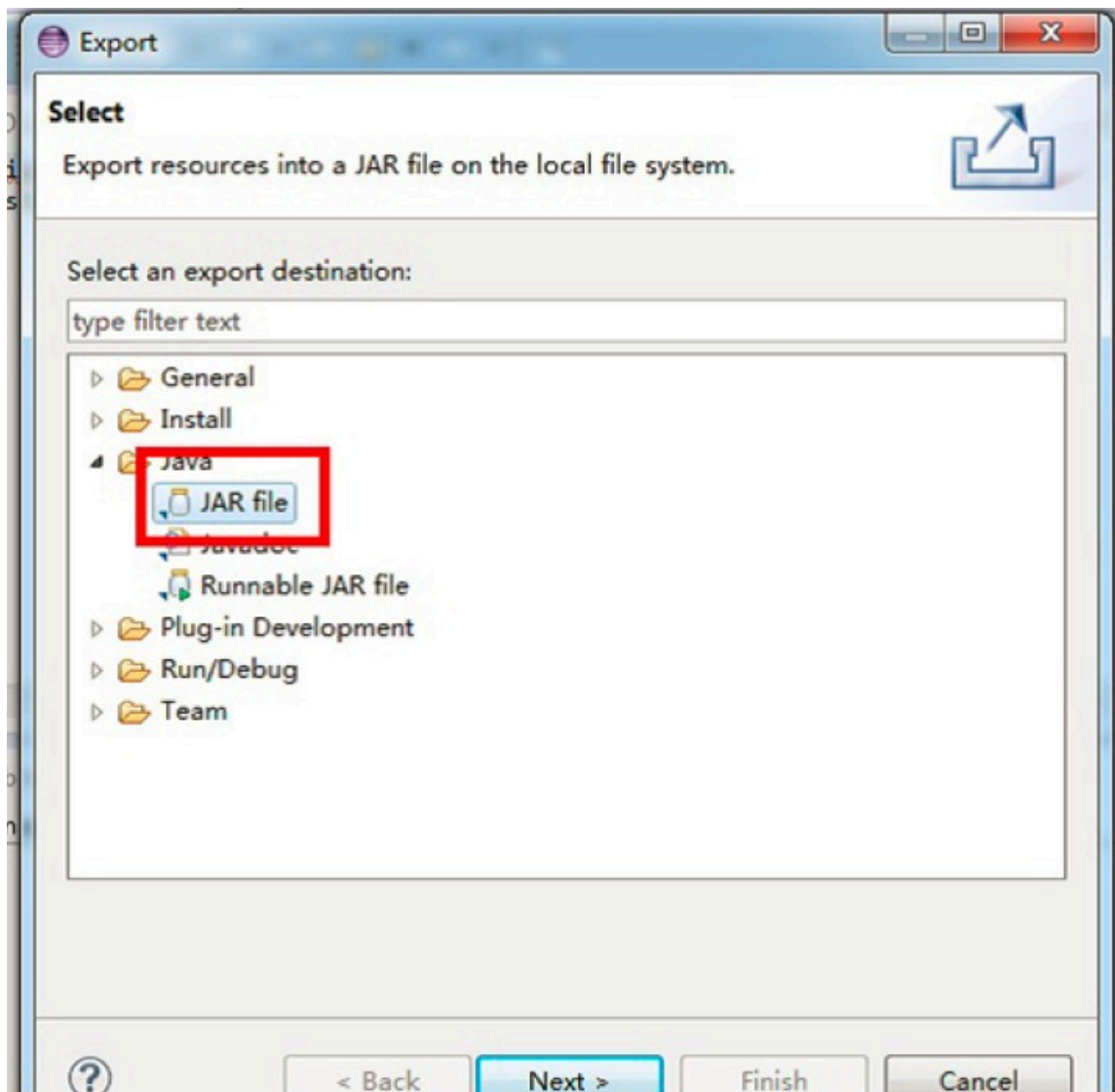
- a. Right-click the src directory and select Export from the shortcut menu, as shown in the following figure.

Figure 1-37: Step 1



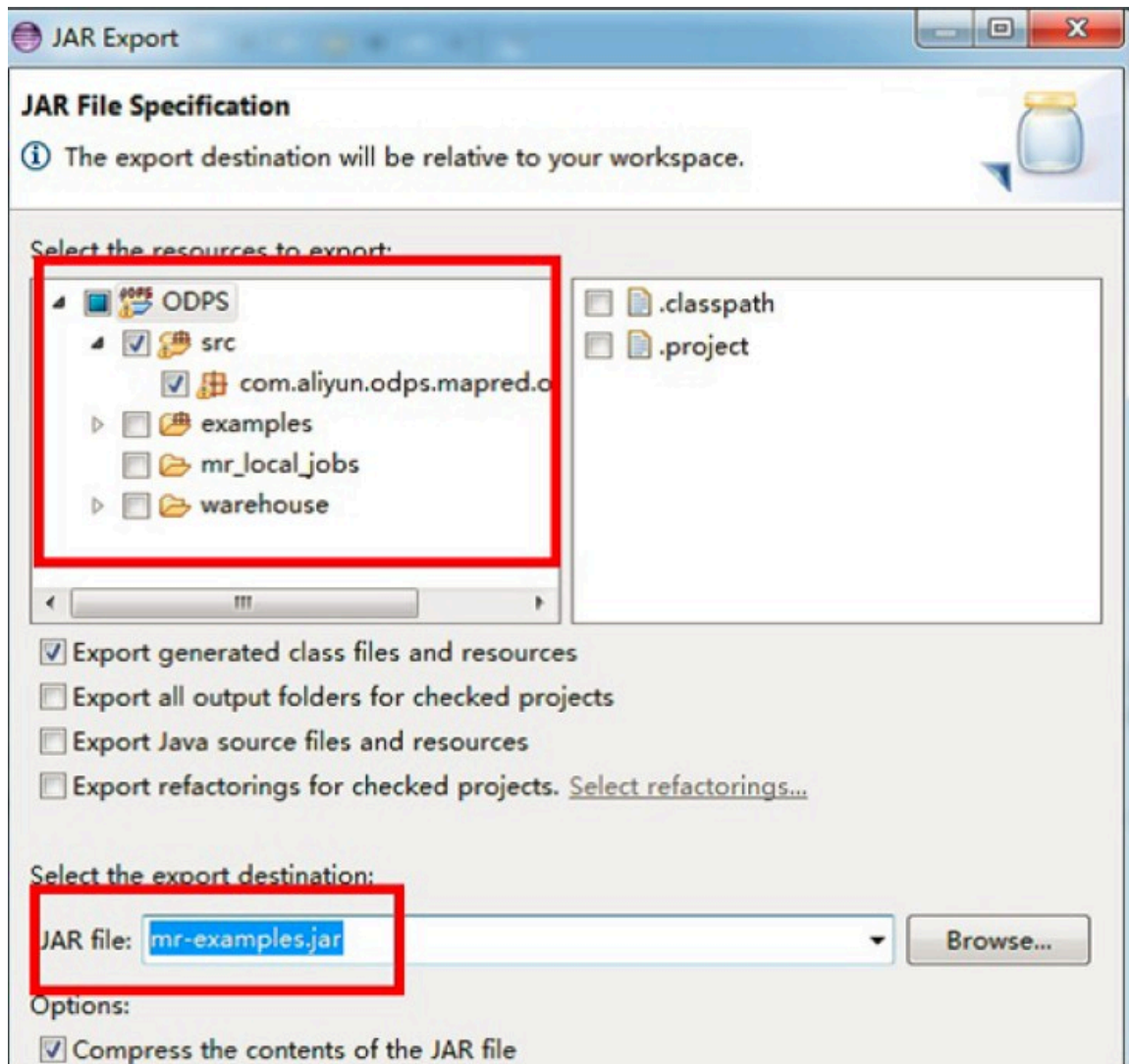
- b. Select JAR file as the export destination, as shown in the following figure.**

Figure 1-38: Step 2



- c. You only need to export the package (com.aliyun.odps.mapred.open.example) in the src directory. Set JAR file to mr-examples.jar, as shown in the following figure.

Figure 1-39: Step 3



**Note:**

In this example, the program package is named mr-examples.jar. You can name the package based on your actual requirements.

- d. Click OK. The export process is complete.

**14**If you want to create a new project locally, you can create a new subdirectory (at the same level as example\_project) under warehouse. The following figure shows the directory structure.



**An example of the schema file:**

**Non-partitioned table:**

```

project=project_name table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:
STRING
-- Partitioned table: project=project_name table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:
STRING partitions=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME
,col5:STRING
-- Note that the following data formats are supported: bigint,
double, boolean, datetime, and string. These formats correspond to

```

the following Java data types: long, double, boolean, java.util.Date, and java.lang.String.

#### An example of the data file:

```
1,1.1,true,2015-06-04 11:22:42 896,hello world
\n,\n,\n,\n,\n
-- Note that the time is in milliseconds. For all time formats, \n
is used to represent NULL.
```



#### Note:

- **Run the MapReduce program locally. The warehouse directory is checked for data tables or resources by default. If the tables or resources do not exist, data of the server is downloaded to the warehouse directory. Then the program runs locally.**
- **After running MapReduce, refresh the warehouse directory to view the generated result.**

### 1.17.2.4 UDF development and running example

#### 1.17.2.4.1 Local debug UDF programs

##### *1.17.2.4.1.1 Run a UDF from the menu bar*

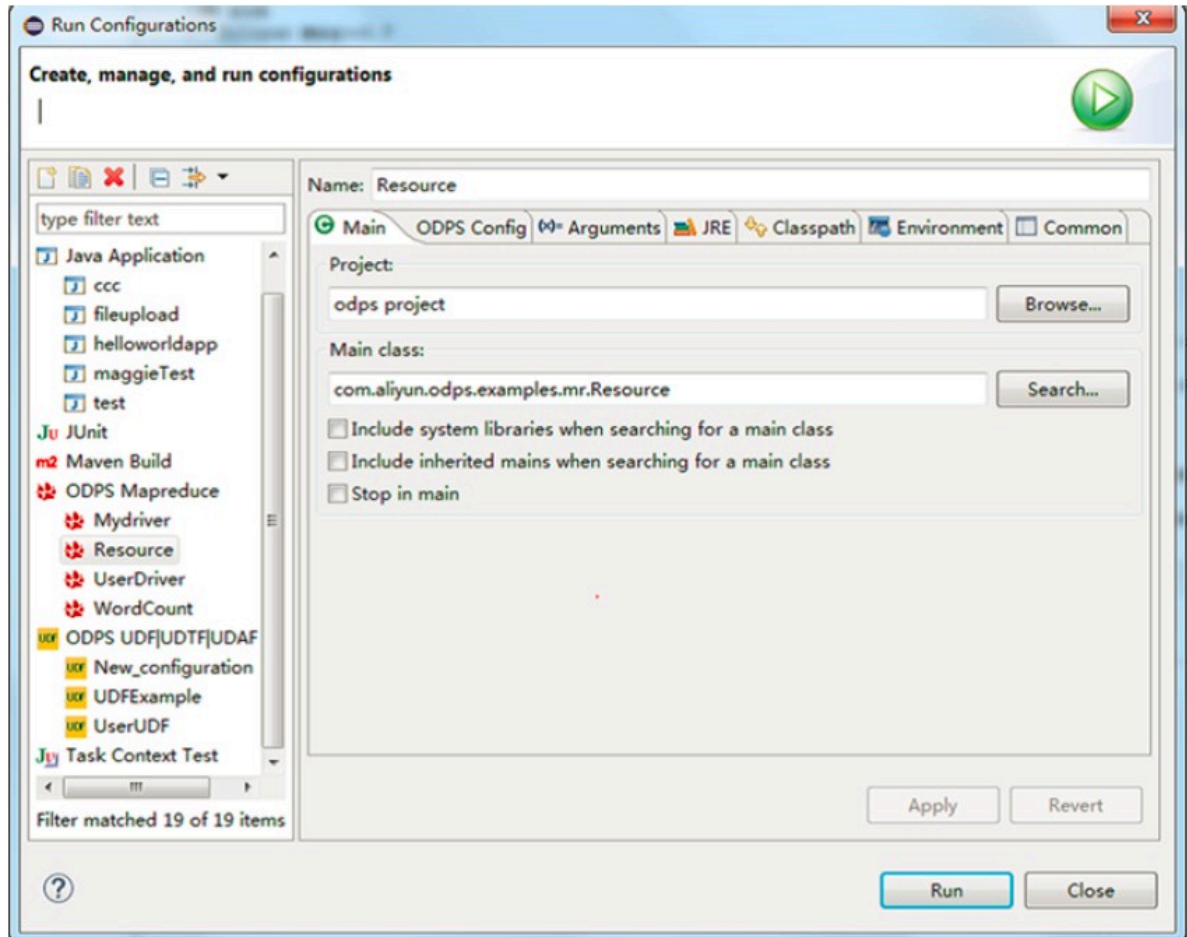
This topic describes how to quickly run a UDF from the menu bar of the Eclipse plugin.

#### Procedure



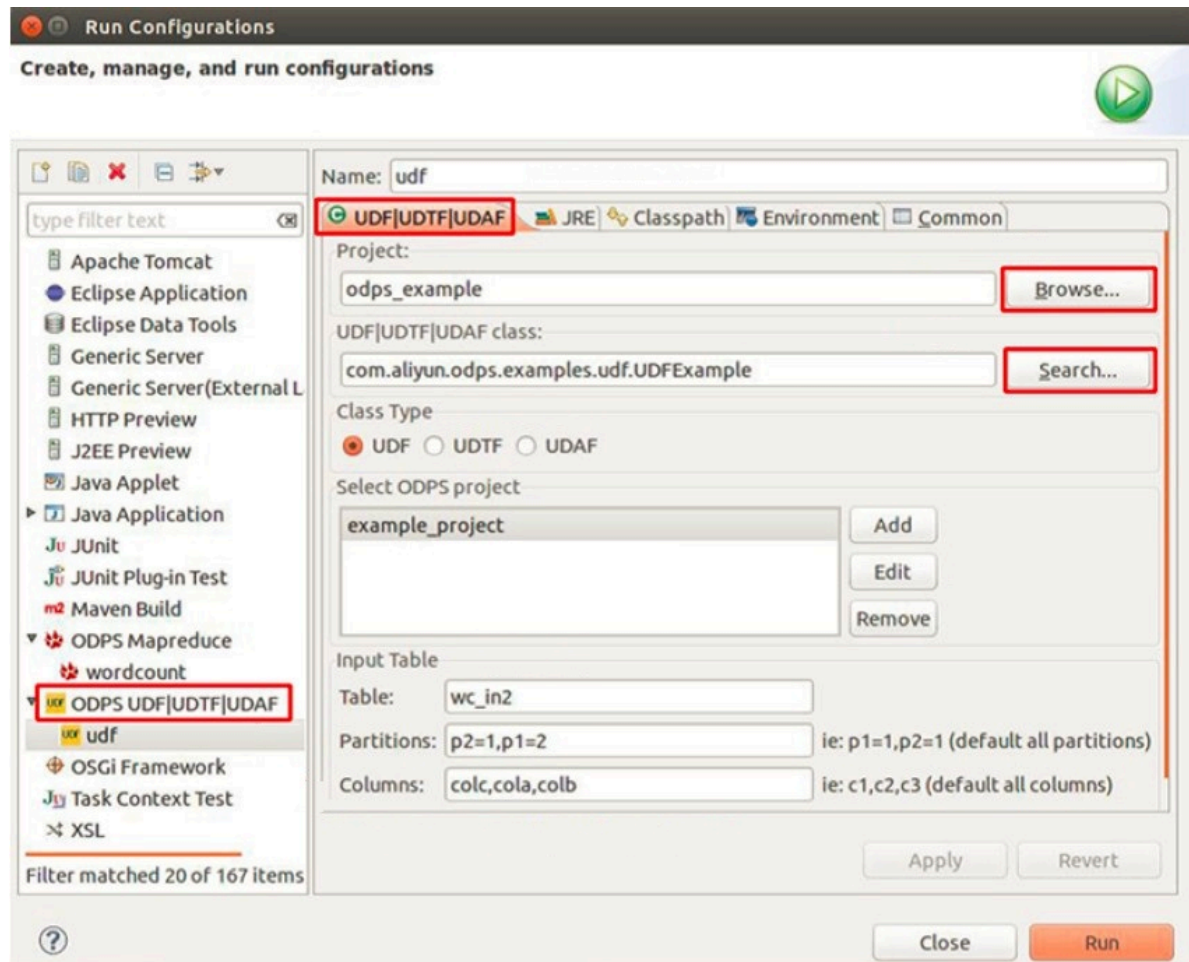
1. Choose **Run > Run Configurations** from the menu bar. A dialog box is displayed, as shown in the following figure.

Figure 1-40: Run Configurations 1



2. To create a run configuration, select the UDF class and type to be run, select an ODPS project, and fill in the input table information, as shown in the following figure.

Figure 1-41: Run Configurations 2

**Note:**

There are three parameters in the Input Table area: Enter the input table of the UDF in Table. Enter the partitions from which data is read in Partitions. Separate multiple partitions by commas (,). Enter the columns in which data is transmitted as UDF parameters in Columns. Separate multiple columns by commas (,).

3. Click Run. The running result is displayed in the console, as shown in the following figure.

Figure 1-42: Console



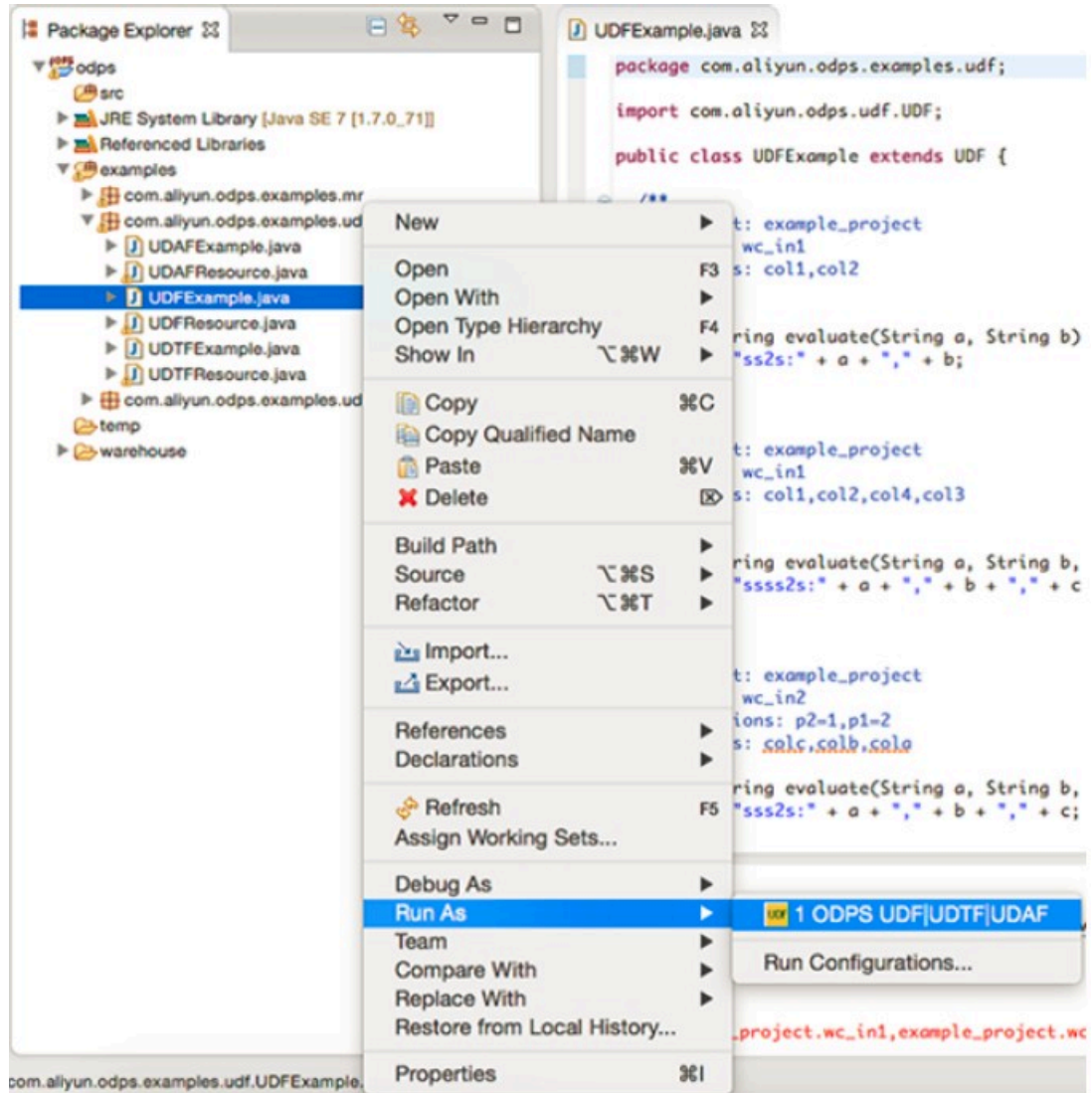
#### ***1.17.2.4.1.2 Use the right-click shortcut menu to quickly run a UDF***

This topic describes how to use the right-click shortcut menu to quickly run a UDF in the Eclipse development plugin.

#### **Procedure**

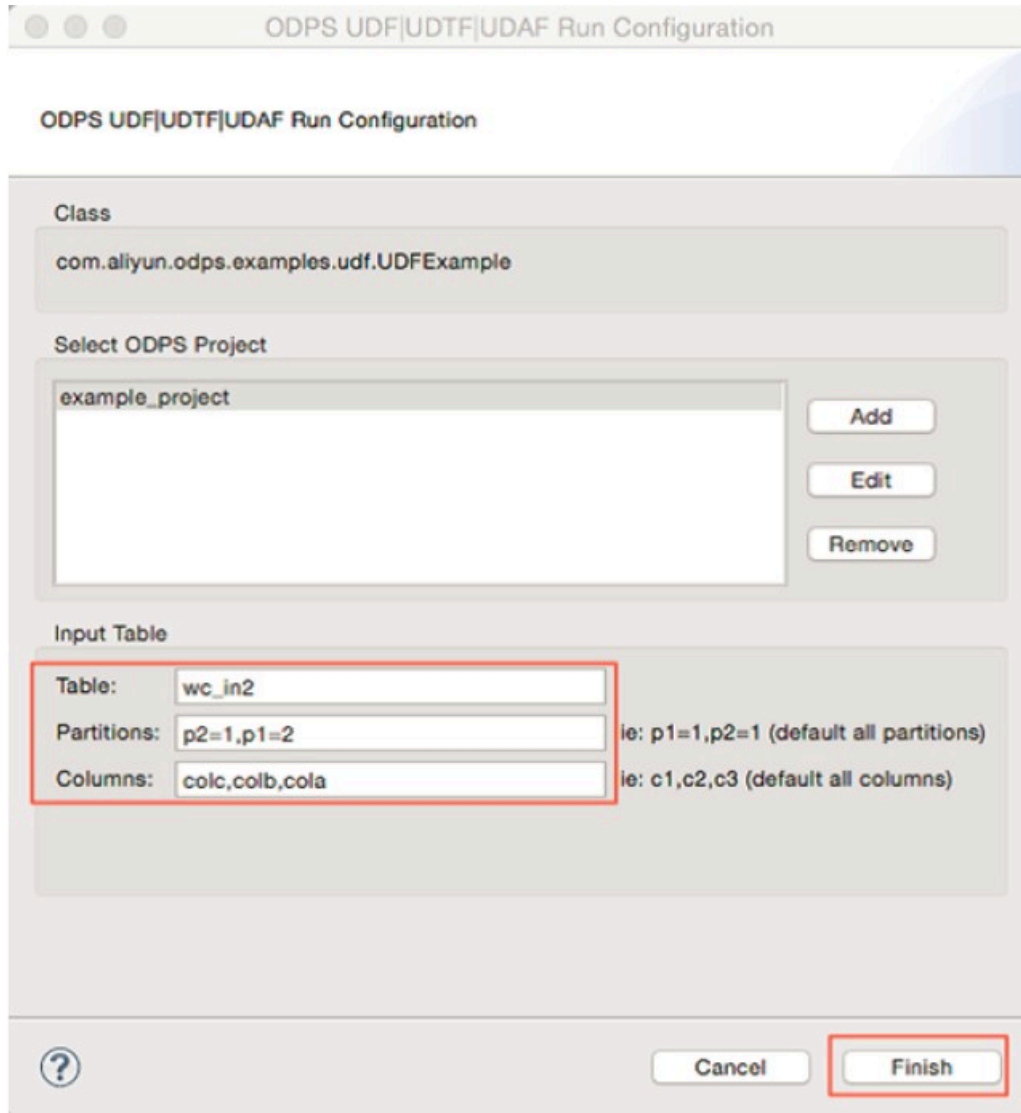
1. Right-click a udf.java file (such as UDFExample.java) and choose Run As > Run UDF|UDAF|UDTF from the shortcut menu, as shown in the following figure.

Figure 1-43: Step 1



2. In the dialog box that appears, configure the relevant parameters, as shown in the following figure.

Figure 1-44: Step 2



The image shows a dialog box titled "ODPS UDF|UDTF|UDAF Run Configuration". It contains several sections for configuring a User Defined Function (UDF). The "Class" section has a text field with the value "com.aliyun.odps.examples.udf.UDFExample". The "Select ODPS Project" section has a list box with "example\_project" selected and buttons for "Add", "Edit", and "Remove". The "Input Table" section has three rows: "Table:" with "wc\_in2", "Partitions:" with "p2=1,p1=2", and "Columns:" with "colc,colb,cola". To the right of these fields are hints: "ie: p1=1,p2=1 (default all partitions)" and "ie: c1,c2,c3 (default all columns)". At the bottom, there are "Cancel" and "Finish" buttons. The "Finish" button is highlighted with a red rectangle.

| Section             | Field        | Value                                   | Hint                                   |
|---------------------|--------------|-----------------------------------------|----------------------------------------|
| Class               | Class Name   | com.aliyun.odps.examples.udf.UDFExample |                                        |
| Select ODPS Project | Project Name | example_project                         |                                        |
| Input Table         | Table:       | wc_in2                                  |                                        |
|                     | Partitions:  | p2=1,p1=2                               | ie: p1=1,p2=1 (default all partitions) |
|                     | Columns:     | colc,colb,cola                          | ie: c1,c2,c3 (default all columns)     |

**Note:**

Table indicates the input table of the UDF. Partitions indicate the partitions from which data is read. Multiple partitions are separated by commas. Columns indicate the columns from which data is read. Multiple columns are separated by commas. These parameters are imported to the UDF as parameters.

3. Click Finish to run the UDF and obtain the result.

### 1.17.2.4.2 Run a UDF program

This topic describes how to run a UDF program in the Eclipse development plugin.

#### Procedure

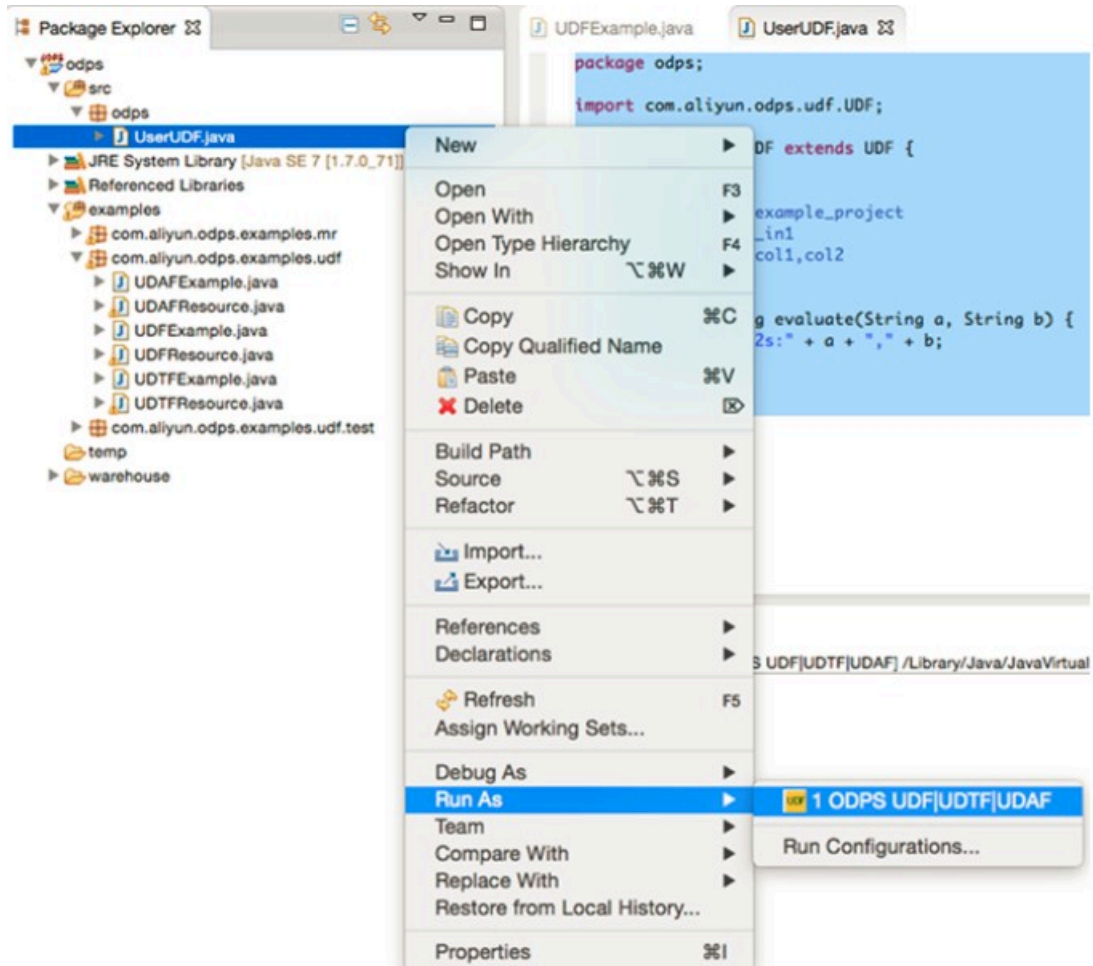
1. **Right-click a project and choose New > UDF (or choose File > New > UDF from the menu bar). Enter a UDF class name and click Finish. A Java file with the same name as the UDF class is generated in the src directory. Edit the content of the java file as follows:**

```
package odps;
import com.aliyun.odps.udf.UDF;
public class UserUDF extends UDF {
/**
 * project: example_project
 * table: wc_in1
 * columns: col1,col2
 *
 */
public String evaluate(String a, String b) { return "ss2s:" + a +
 "," + b;
}
```

}

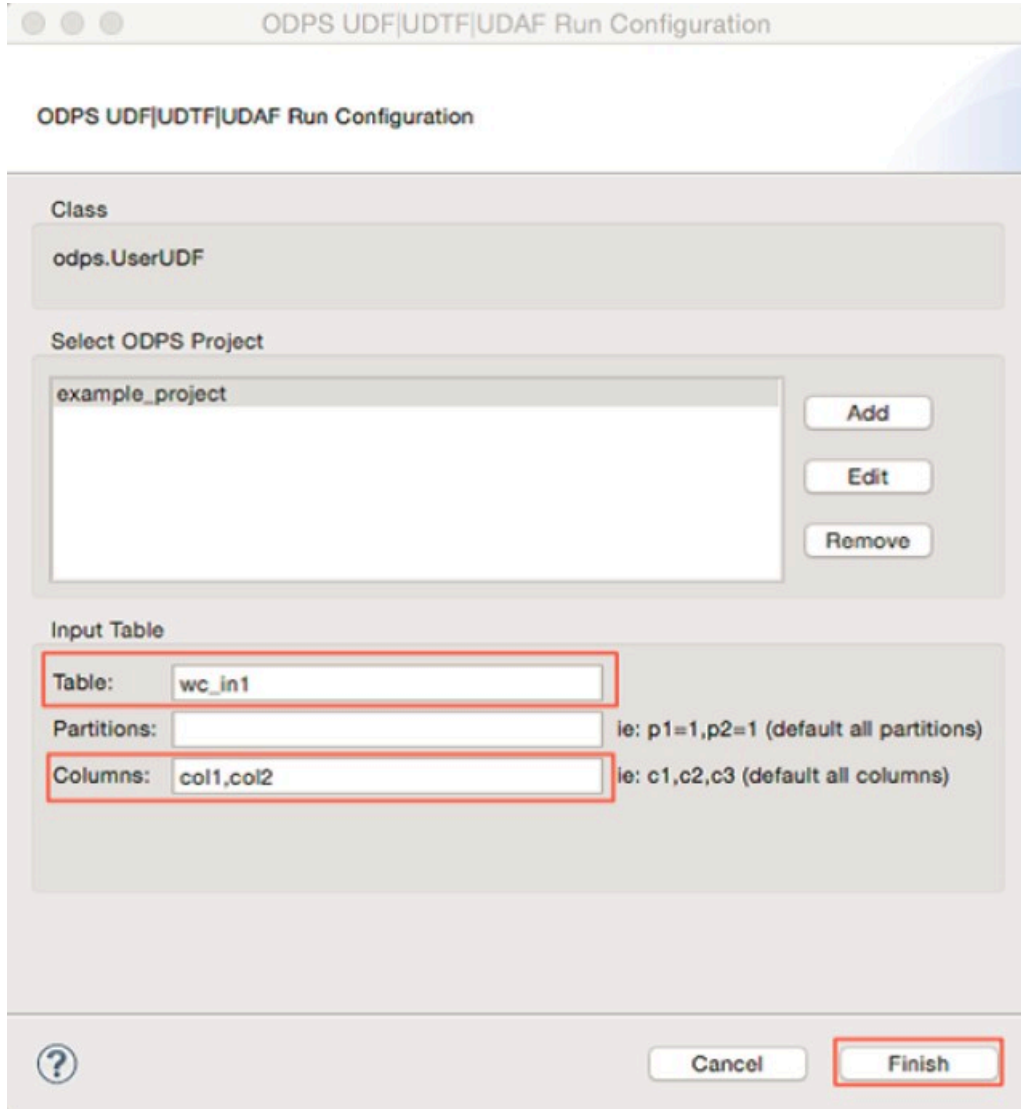
2. Right-click the Java file (such as UserUDF.java) and choose Run As > ODPS UDF|UDTF|UDAF from the shortcut menu, as shown in the following figure.

Figure 1-45: Step 1



3. In the dialog box that appears, configure the relevant parameters, as shown in the following figure.

Figure 1-46: Step 3-1



The image shows a dialog box titled "ODPS UDF|UDTF|UDAF Run Configuration". It contains the following fields and controls:

- Class:** A text field containing "odps.UserUDF".
- Select ODPS Project:** A list box containing "example\_project". To the right of the list box are three buttons: "Add", "Edit", and "Remove".
- Input Table:** A section containing three rows:
  - Table:** A text field containing "wc\_in1".
  - Partitions:** A text field (empty) followed by the text "ie: p1=1,p2=1 (default all partitions)".
  - Columns:** A text field containing "col1,col2" followed by the text "ie: c1,c2,c3 (default all columns)".
- Bottom:** A question mark icon on the left, and "Cancel" and "Finish" buttons on the right. The "Finish" button is highlighted with a red border.

4. Click Finish to obtain the result.

```
ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2
```



**Note:**

This example shows how to run a UDF program. You can use the same method to run a UDTF program.



### 1.17.2.5 Graph running example

After creating a MaxCompute project, you can write your own Graph program and debug it locally by performing the following steps.

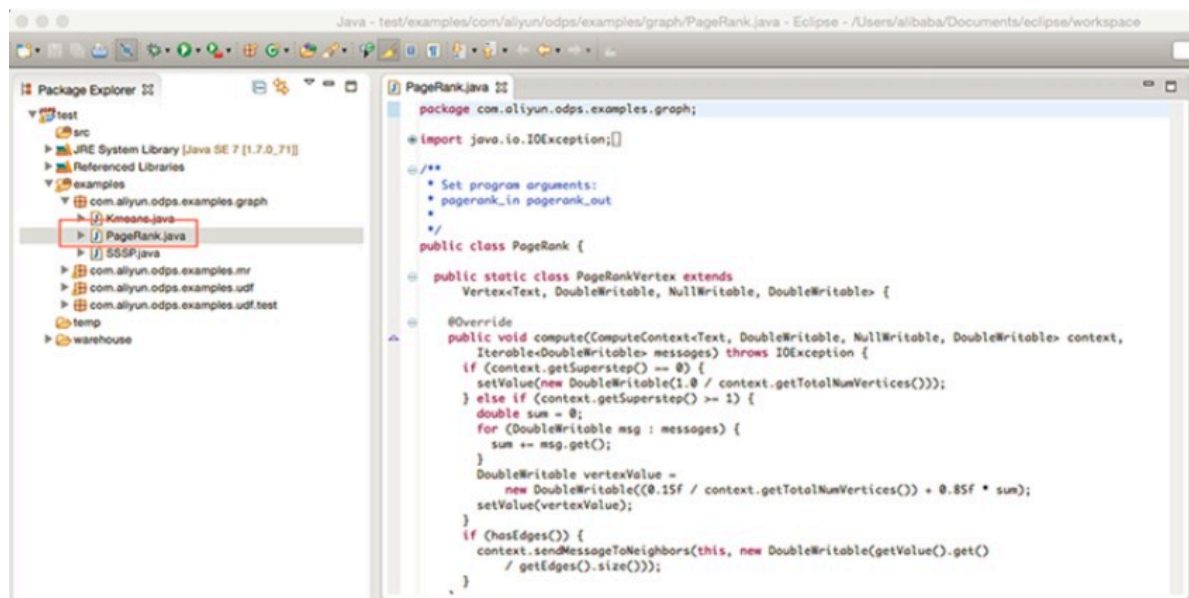
#### Context

In this example, you can use PageRank.java provided by the plugin to perform local debugging.

#### Procedure

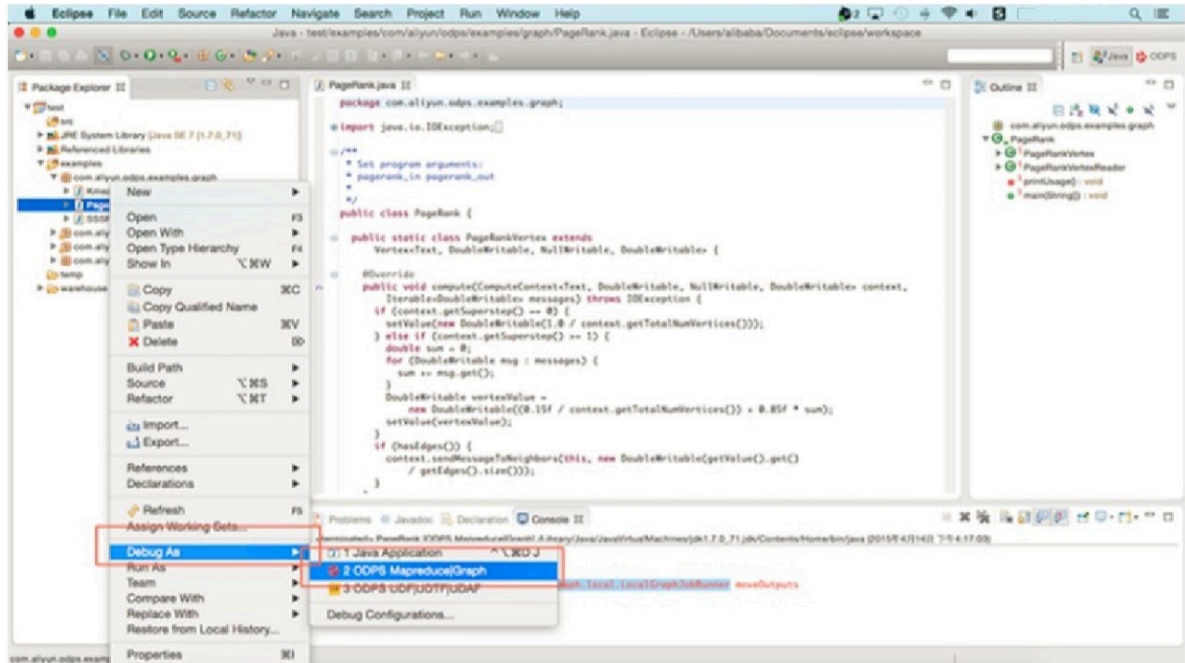
1. Choose examples > PageRank.java, as shown in the following figure.

Figure 1-47: PageRank.java code 1



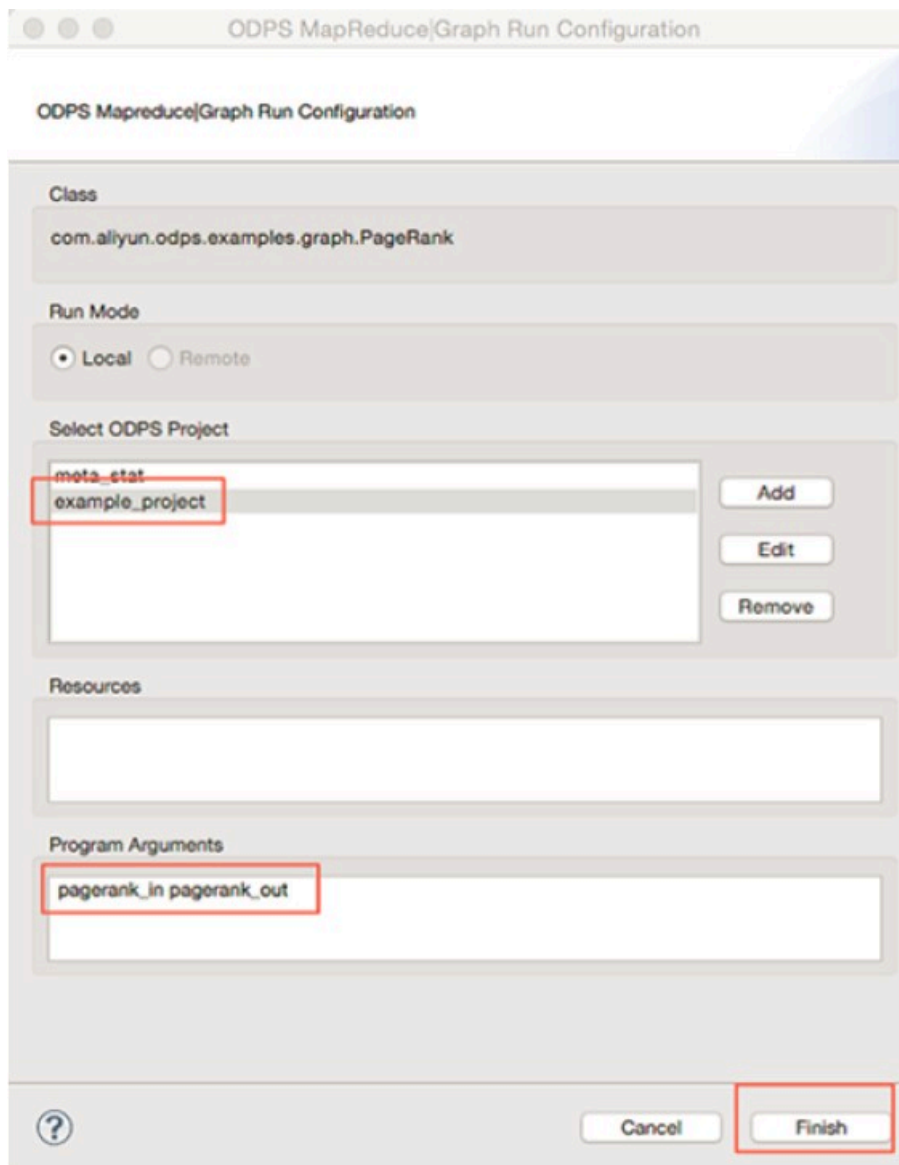
## 2. Right-click it and choose Debug As > ODPS MapReduce|Graph, as shown in the following figure.

Figure 1-48: PageRank.java code 2



3. In the pop-up dialog box, enter the information as shown below.

Figure 1-49: Configuration Drawings

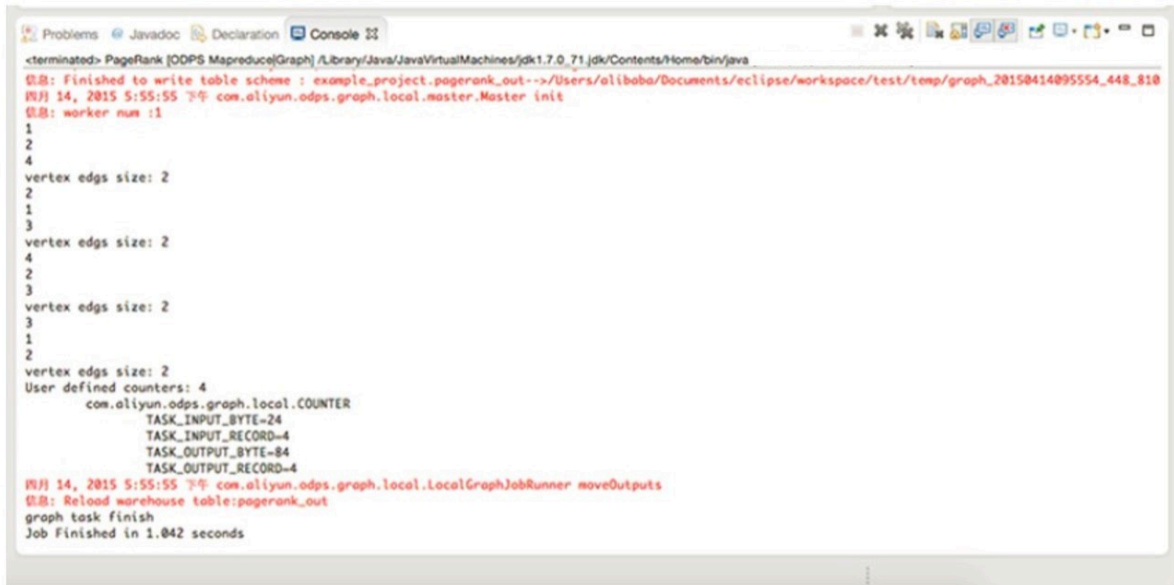


The image shows a dialog box titled "ODPS MapReduce/Graph Run Configuration". It contains several sections for configuring a job:

- Class:** A text field containing "com.aliyun.odps.examples.graph.PageRank".
- Run Mode:** Two radio buttons, "Local" (selected) and "Remote".
- Select ODPS Project:** A list box containing "meta\_stat" and "example\_project". The "example\_project" entry is highlighted with a red box. To the right of the list are three buttons: "Add", "Edit", and "Remove".
- Resources:** An empty text field.
- Program Arguments:** A text field containing "pagerank\_in pagerank\_out", which is highlighted with a red box.
- Footer:** A question mark icon on the left, and "Cancel" and "Finish" buttons on the right. The "Finish" button is highlighted with a red box.

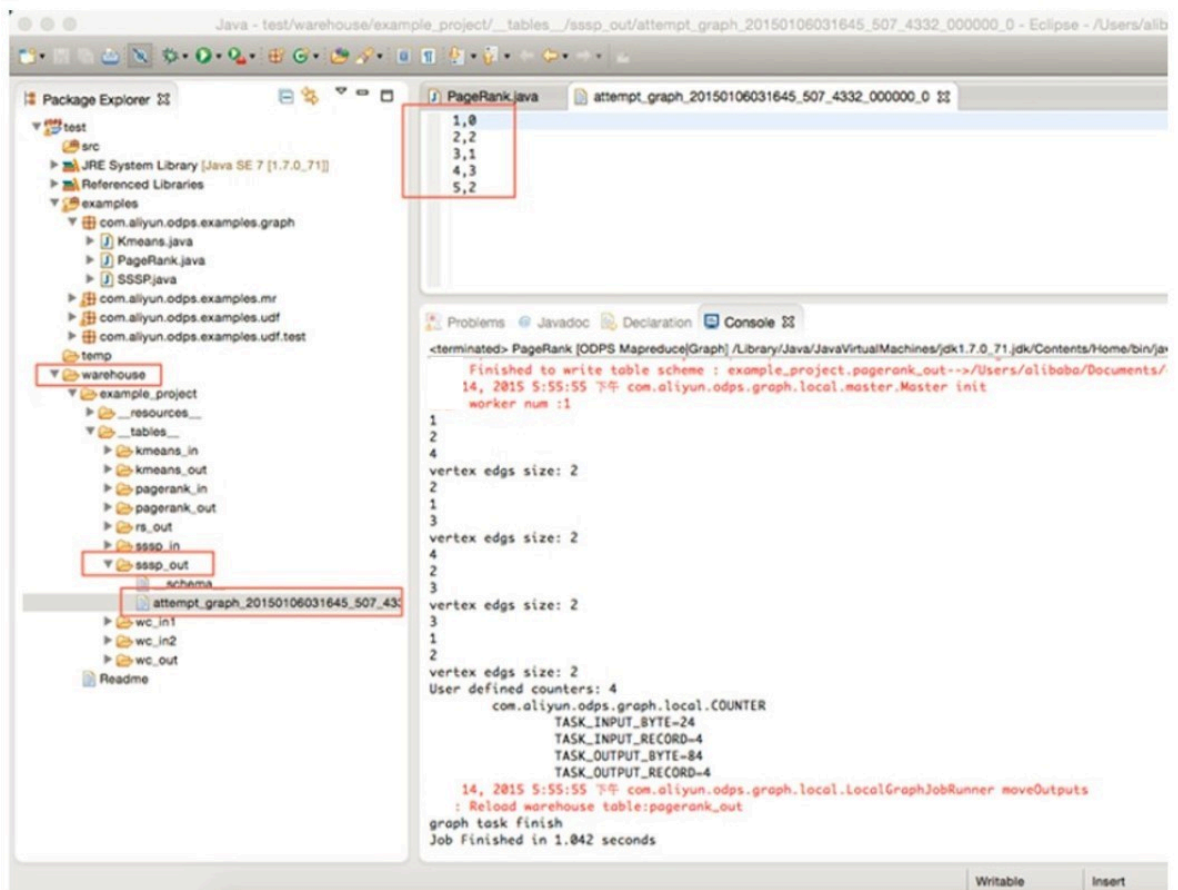
#### 4. Click Finish. Check the running result, as shown in the following figure.

Figure 1-50: Running Result



Check the local computing result as shown below:

Figure 1-51: Local computing result



After passing the debugging, you can package the program, upload it to MaxCompute in the form of JAR resource, and submit the Graph job.



**Note:**

- For more information about the packaging process, see [MapReduce running example](#).
- For more information about the directory structure of local results, see [MapReduce running example](#).
- For more information about uploading JAR resources, see [Compile and run a Graph job](#).

## 1.18 MaxCompute FAQ

**This topic describes MaxCompute FAQ and solutions.**

How to check MaxCompute resource usage when SQL statements are executed slowly?

**Log on to the MaxCompute AG as user admin and run the following commands. The procedure is as follows:**

- 1. Run the following command to view remaining resources on various hosts in the MaxCompute cluster in ascending order:**

```
r tfrl|sed 's/,//g'|sort -t "|" -k2 -n
```

- 2. Run the following command to view resource details on various hosts and total cluster resources in MaxCompute:**

```
r ttrl|sed 's/,//g'
```

- 3. You can judge MaxCompute resource usage through the ratio of remaining resources to total resources.**

How to handle the situation where the jobs submitted by a project are executed slowly even if the MaxCompute cluster has sufficient remaining resources?

**A possible reason for slow job execution is that the quota group resource has already been filled. You can increase the group resource quota by taking the following steps:**

1. Log on to the MaxCompute AG as user admin and run the following command to check quota group resource usage:

```
r quota
```

2. After confirming that the quota group resources for this project are exhausted, you can make modifications to the quota list through MaxCompute in Big Data Manager.

Modify quota group settings

1. Run the following command in the MaxCompute AG to create or modify a quota:

```
sh/apsara/deploy/rpc_wrapper/rpc.sh setquota -i $QUOTAID -a $
QUOTANAME -t fair -s$max_cpu_quota $max_mem_quota -m $min_cpu_quota
$min_mem_quota
```



**Note:**

If \$QUOTAID already exists, that quota will be modified. Otherwise, a quota with that ID will be created.

2. Log on to MaxCompute in Big Data Manager and complete relevant settings.

How to use the Console?

1. Log on to the MaxCompute AG as user admin.
2. Run the following commands:

```
/apsara/odps_tools/clt/bin/odpscmd
```

```
use meta;
```

3. Run the following command to view all tables in the metadatabase:

```
show tables;
```

4. Run the following command to obtain the description of a specific table:

```
desc <table>;
```

How to handle the situation where disk capacity is exhausted?

Generally, you can clear scripts to free disk capacity. A possible reason is that the root directory of MaxCompute AG or the /apsara directory occupies too much disk space. You need to clear scripts in these two directories.

How to find AccessKeys and configure correct AccessKeys?

1. **Access the framework cluster management page of the Apsara Stack data center. Choose Operations > Cluster Operations. Select the cluster whose AccessKey you want to find and configure, and access the cluster configuration page.**
2. **Double-click the kv.conf file in the file list. You can find the AccessKey information. You can also modify it directly in this file and save your modifications.**

How to blacklist a MaxCompute host?

1. **Log on to the Apsara AG as the admin user. Run the following command to enable the Fuxi blacklist function.**

```
r sgf fuximaster"{\"fuxi_Enable_BadNodeManager\":false}"
```

2. **Run the following command to view the Fuxi blacklist:**

```
/apsara/deploy/rpc_wrapper/rpc.shblacklist cluster get
```

3. **Run the following command to add a MaxCompute host to the Fuxi blacklist:**

```
/apsara/deploy/rpc_wrapper/rpc.shblacklist cluster add $hostname
```

4. **Run the following command to view the Fuxi blacklist and confirm that the host is added to the Fuxi blacklist:**

```
/apsara/deploy/rpc_wrapper/rpc.shblacklist cluster get
```

How to export data from MaxCompute?

**There are two methods to export MaxCompute data: The first is to use the tunnel command. The second is to configure synchronization tasks on DataWorks to export data from MaxCompute to other destinations.**

How to view MaxCompute version?

**Run the following commands to view the MaxCompute version:**

```
cat /apsara/odps_info/version|grep odps
```

```
cat /apsara/version
```

How to restart MaxCompute services?

1. **Save the resident MaxCompute services configurations to a file. This configuration file will be needed when restarting MaxCompute services.**

```
ssh odpsAG
cd /home/admin/
```

You can run the following commands to save the configurations of resident services into a file. If there are services listed below that you do not use, you can ignore them.

PS: You can use the `r al` command to view resident services.

```
r plan Odps/CGServiceControllerx > CGServiceControllerx
r plan sys/sqlonline-OTS > sqlonline-OTS
r plan Odps/MessengerService > MessengerService
r plan Odps/OdpsService > OdpsService
r plan Odps/HiveServerx > HiveServerx
r plan Odps/XStreamService > XStreamService
r plan Odps/QuotaService > QuotaService
r plan Odps/ReplicationService > ReplicationService
```

## 2. Stop MaxCompute services.

```
r sstop Odps/CGServiceControllerx
r sstop sys/sqlonline-OTS
r sstop Odps/MessengerService
r sstop Odps/OdpsService
r sstop Odps/HiveServerx
r sstop Odps/XStreamService
r sstop Odps/QuotaService
r sstop Odps/ReplicationService
```

## 3. Start MaxCompute services.

```
ssh odpsAG
cd /home/admin/
r start CGServiceControllerx
r start sqlonline-OTS
r start MessengerService.txt
r start OdpsService.txt
r start HiveServerx.txt
r start XStreamService.txt
r start QuotaService.txt
r start ReplicationService.txt
```

How to power on and off MaxCompute?

### 1. Save the resident MaxCompute services configurations to a file. This configuration file will be needed when restarting MaxCompute services.

```
ssh odpsAG
cd /home/admin/
PS: Run the following commands to save the configurations of
resident services into a file. If some services are not used, just
ignore them.
PS: You can use the r al command to view resident services.
r plan Odps/CGServiceControllerx > CGServiceControllerx
r plan sys/sqlonline-OTS > sqlonline-OTS
r plan Odps/MessengerService > MessengerService
r plan Odps/OdpsService > OdpsService
r plan Odps/HiveServerx > HiveServerx
r plan Odps/XStreamService > XStreamService
r plan Odps/QuotaService > QuotaService
r plan Odps/ReplicationService > ReplicationService
```

### 2. Stop MaxCompute services.

```
r sstop Odps/CGServiceControllerx
```



```
r sstop sys/sqlonline-OTS
r sstop Odps/MessengerService
r sstop Odps/OdpsService
r sstop Odps/HiveServer
r sstop Odps/XStreamService
r sstop Odps/QuotaService
r sstop Odps/ReplicationService
```

**3. Shut down the Apsara system.**

```
/home/admin/dayu/bin/allapsara stop
```

**4. Shut down computing nodes gracefully.**

```
Shutdown
```

**5. Start computing nodes.**

**6. Start the Apsara system.**

```
/home/admin/dayu/bin/allapsara start
```

**7. Start MaxCompute services.**

```
ssh odpsAG
cd /home/admin/
r start CGServiceController
r start sqlonline-OTS
r start MessengerService.txt
r start OdpsService.txt
r start HiveServer.txt
r start XStreamService.txt
r start QuotaService.txt
r start ReplicationService.txt
```

How to remove high load on a host?

- 1. Log on to the host experiencing a high load and run the top command to check whether task processes occupy many resources.**
- 2. If so, you have to wait until these tasks are completed or ask users to kill these tasks (usually user tasks occupy many resources).**

## 2 DataWorks

---

### 2.1 What is DataWorks?

**DataWorks is an end-to-end big data platform that uses MaxCompute as its compute engine. It integrates all processes from data collection to data display and from data analysis to application running. DataWorks provides various features to help you quickly and effectively complete the entire research and development (R&D) process. The entire R&D process involves data integration, data development, data governance, data service, data quality, and data security.**

**DataWorks is an all-in-one solution for collecting, presenting, and analyzing data, and driving application development. It not only supports offline processing, analysis, and data mining of large amounts of data, but also integrates core data-related technologies such as data development, data integration, production and operations and maintenance (O&M), real-time analysis, asset management, data quality, data security, and data sharing. In addition, it provides the Data Service and Machine Learning Platform for Artificial Intelligence (PAI) services.**

**In 2018, Forrester, a globally recognized market research company, named Alibaba Cloud DataWorks and MaxCompute as a world-leading cloud-based data warehouse solution. This solution is currently the only product from a Chinese company to receive such an acknowledgment. Building on the success of the previous version, DataWorks V2.0 incorporates several new additions, such as workflows and script templates. DataWorks V2.0 supports dual workspaces for development, isolates the development environment from the production environment, adopts standard development processes, and uses a specific mechanism to reduce errors in code.**

## Features

- **Cloud-hosted environment**

- **DataWorks provides powerful scheduling capabilities:**

- **DataWorks supports node triggering by time and dependency.**

- **DataWorks enables tens of millions of nodes to run accurately and on time based on directed acyclic graph (DAG) relationships every day.**

- **DataWorks supports node execution at custom intervals in minutes, days, hours, weeks, or months.**

- **DataWorks is a cloud-hosted environment that frees you from server deployment.**

- **DataWorks provides the isolation function to ensure that nodes between different tenants do not affect each other.**

- **DataWorks supports multiple node types, such as data synchronization, shell, ODPS SQL, and ODPS MR. It analyzes and processes complex data based on the dependency between nodes.**

- **Data conversion: By using the powerful computing capability of MaxCompute, DataWorks ensures the superior performance on analyzing and processing big data.**

- **Data synchronization: With the strong support of data integration, DataWorks supports more than 20 types of data stores and provides stable and efficient data transmission functions.**

- **Visualized code development**

**DataWorks provides visualized code development and workflow designer pages . You can complete complex data analysis nodes through simple drag-and-drop operations without using any development tools. A browser with Internet connection enables you to develop code anytime, anywhere.**

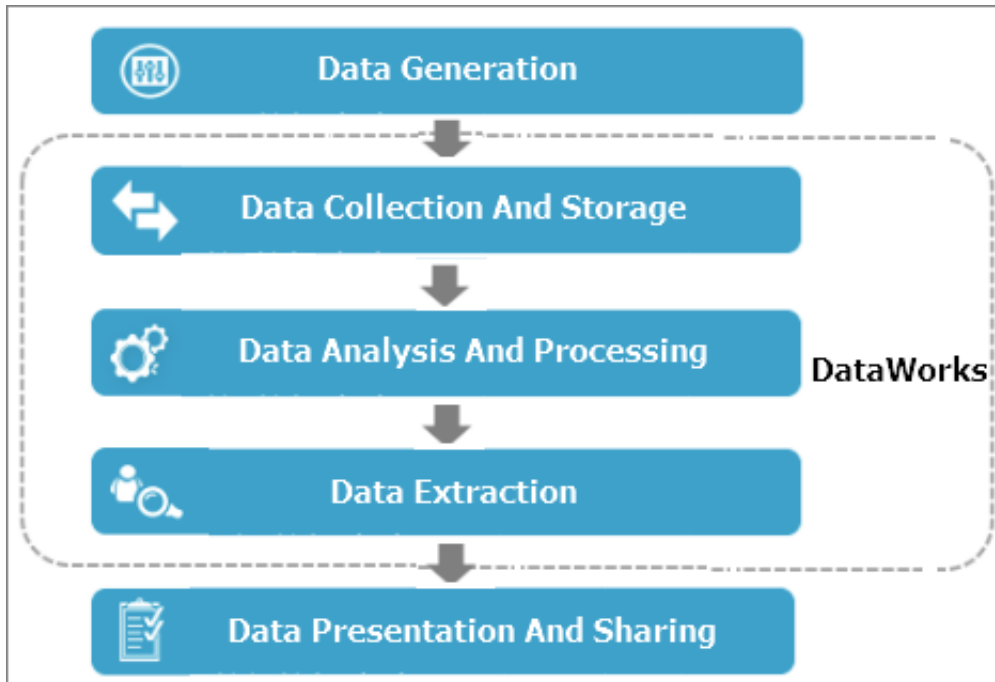
- **Monitoring and alerts**

**The Operation Center module provides a visualized node monitoring and management tool and displays the overall node execution status in a DAG.**

**You can easily configure various alert notification methods to promptly notify relevant staff when a node error occurs. This ensures normal business operation.**

## Development process

As shown in the figure, data development is a process of generating, collecting, storing, analyzing, extracting, presenting, and sharing data.

**Note:**

You can perform steps in dashed-line boxes through DataWorks.

1. **Generate data:** Each business system generates a large amount of structured data every day and stores the data in its own databases, such as MySQL, Oracle, and RDS databases.
2. **Collect and store data:** You need to synchronize data from various business systems to MaxCompute so that the data can be processed by MaxCompute. MaxCompute has powerful data storage and processing capabilities.  
  
DataWorks supports the integration of various data stores. It allows you to synchronize data from business systems to MaxCompute at the preset recurrence .
3. **Analyze and process data:** You can process data by using ODPS SQL and ODPS MR, and perform data analysis and mining in MaxCompute to discover data value.
4. **Extract data:** You can export data processing and analysis results to business systems for further processing.

5. **Present and share data:** You can present data processing and analysis results in multiple methods such as reports or geographic information system (GIS). You can also share the results with others.

## 2.2 Planning and preparation

### 2.2.1 Planning and preparation

Before logging on to the DataWorks console, you must create a department, users, and a project.

#### Prerequisites

- Before you log on to the Apsara Stack Console, ensure that you have obtained the IP address of the Apsara Stack Console or the domain name of the DataWorks server from deployment engineers. The URL of the Apsara Stack Console is **http://IP address of the Apsara Stack Console or domain name of the DataWorks server/***manage*.
- We recommend that you use the Google Chrome browser.

#### Procedure

1. In the address bar of the browser, enter the URL of the Apsara Stack Console, **http://IP address of the Apsara Stack Console or domain name of the DataWorks server/***manage*, and then press Enter.
2. Enter a valid username and the corresponding password, and click Log On. The Dashboard page appears.
  - The system has a default super administrator, whose username and password are both "super". The super administrator can create system administrators . A system administrator can create system users and notify the users of the default passwords by SMS or email.
  - The first time you log on to the Apsara Stack Console, you must modify the password as prompted. For security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain two or more types of the following characters: letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. In the left-side navigation pane, choose **User Center > Department Management**. Then, click **Add** to create a department.
4. In the left-side navigation pane, choose **User Center > User Management**. Then, click **Create** to create a user.
5. In the left-side navigation pane, choose **User Center > Project Management**. Then, click **Add Projects** to create a project.
6. In the left-side navigation pane, choose **Big Data > MaxCompute**. On the **Accounts** tab, click **Create Account** to create an account.
7. Click the **Projects** tab, and click **Create**. In the dialog box that appears, set parameters to create a MaxCompute project.

**Note:**

- You can create workspaces and manage MaxCompute projects in a visualized manner in the DataWorks console only after you have created a department, users, and MaxCompute projects in the Apsara Stack Console.
- Users only have access to DataWorks but cannot use DataWorks services after being added to organizations by administrators. They can use DataWorks services only after workspace administrators assign the developer role to them.

## 2.2.2 Workspace types

DataWorks provides basic and standard workspaces.

### Standard workspaces

You need to associate a standard workspace to two different MaxCompute projects . One of the projects serves as the development environment, and the other serves as the production environment. Standard workspaces help ensure standard code development and strict table permission control. Standard workspaces imposes limits on table deletion in the production environment to ensure data security.

- You can only modify nodes in the development environment, and cannot modify them in the production environment. This mechanism ensure the stability of code in the production environment.
- The development environment does not allow periodic scheduling by default. This aims to protect the development environment from occupying excessively

large amounts of resources and ensure sufficient resources and high stability for the production environment.

- The production environment runs with a default production environment system account. All tables created in the production environment belongs to this account. Before operating on a table, you need to submit a request for table permissions.

#### Basic workspaces

You can associate a basic workspace to only one MaxCompute project. Basic workspaces do not separate development and production environments, which support administration (such as on table permissions) not as powerful as standard workspaces.

The advantage and disadvantage of basic workspaces are described as follows:

- **Advantage:** Code of a node takes effect immediately after you submit the node without the need of publication.
- **Disadvantage:** Developers can delete any table in the workspace, which is risky.

Basic workspaces are suitable for agile projects or test projects with small data volumes.

## 2.3 Quick start

### 2.3.1 Log on to the DataWorks console

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

#### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.

**3. Enter the correct username and password.**

- **The system has a default super administrator with the username super. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.**
- **You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).**

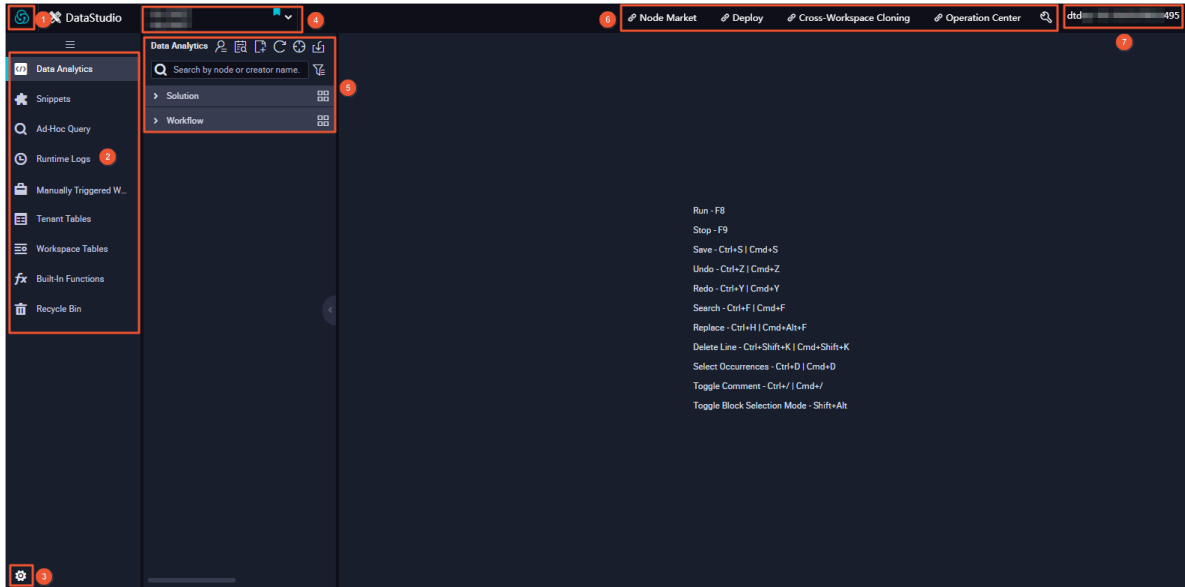
**4. Click LOGIN to go to the Dashboard page.**

**5. In the left-side navigation pane, choose Big Data > DataWorks.**

**6. Click Management Console in the upper-right corner. On the DataWorks page, select the relevant department, and click DataWorks.**



7. After you open the DataStudio page, move the pointer over the DataWorks main menu, and click the required DataWorks service.



The following table describes the console features.

| No. | Feature              | Description                                                                                                                                                                                          |
|-----|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | DataWorks main menu  | From the main menu, you can go to the following services: DataStudio, Data Quality, PAI, Data Service, Data Protection, Operation Center, Data Asset, Realtime Analysis, and Security Center.        |
| 2   | DataStudio           | The modules involved in DataStudio include Data Analytics, Snippets, Ad-Hoc Query, Runtime Logs, Manually Triggered Workflows, Tenant Tables, Workspace Tables, Built-in Functions, and Recycle Bin. |
| 3   | Configuration center | In the configuration center, you can configure various items, such as module arrangement, editor style, and font. You can also manage the table hierarchy and table folders.                         |
| 4   | Workspace name       | From the workspace drop-down list, you can click to switch to the DataWorks workspace you want to access.                                                                                            |

| No. | Feature                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5   | DataStudio submenus                        | <p>The submenus of DataStudio consist of Workflow and Solution. A workflow is an organic combination of modules arranged by business attributes. These modules include Data Integration, Data Analytics, Table, Resource, Function, Machine Learning, and Control. You can combine different types of correlated nodes to facilitate code development by business.</p> <p>A solution is a flexible combination of different workflows. You can combine workflows into a solution to develop code more efficiently.</p> |
| 6   | Shortcut buttons in the top navigation bar | You can click these buttons to switch to Deploy, Operation Center, and Project Manage.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7   | Account management                         | You can click here to view the user details or log off from the current account.                                                                                                                                                                                                                                                                                                                                                                                                                                       |

### 2.3.2 Create a DataWorks workspace

This topic describes how to create a DataWorks workspace.

#### Procedure

1. [Log on to the DataWorks console](#).
2. In the left-side navigation pane, choose Project Management > Workspaces.  
Then, click Create Workspace.

3. Set parameters in the Basic Information section.

Two workspace modes are available, which are Basic Mode (Production Environment Only) and Standard Mode (Development and Production Environments). We recommend that you select Standard Mode to create a workspace.

4. Set parameters in the Advanced Settings section. You can select whether to enable periodic scheduling and whether to allow downloading SELECT query results. You also need to associate the workspace with MaxCompute projects.
5. Click OK.

### 2.3.3 Create a workflow

This topic describes how to create an automatically triggered workflow as an example.

#### Context

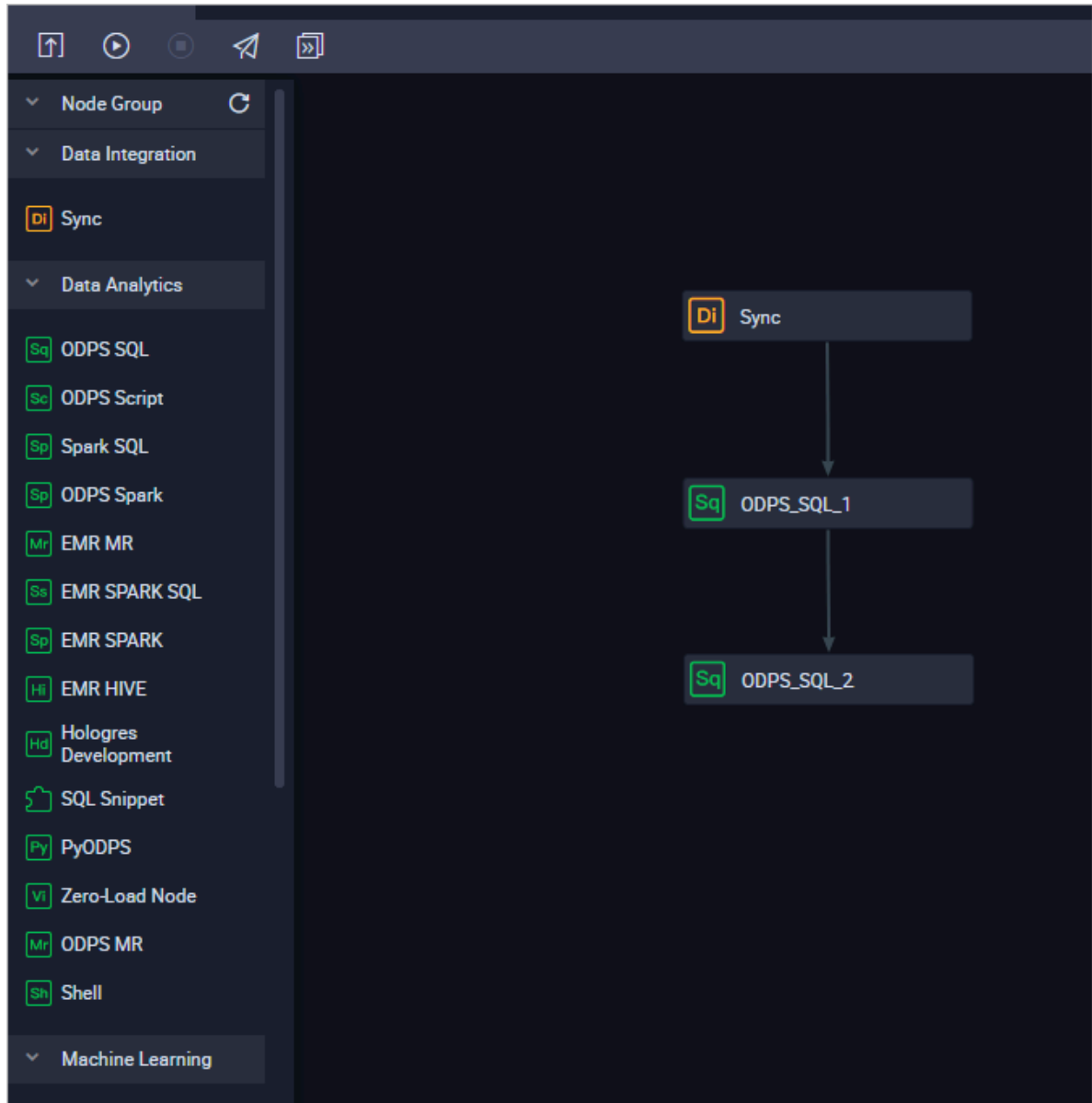
Workflows are defined in compliance with the following rules:

- Each workflow has only one directed acyclic graph (DAG) and one root node.
- The name of each workflow must be unique in a workspace.
- Each node in a workflow can be regarded as a single object so that nodes in other workflows can depend on it.
- Workflows in different workspaces can be dependent on each other.
- Ancestor nodes are optional. Nodes in a workflow can be scheduled separately based on their own attributes.

#### Create a workflow

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
3. In the Create Workflow dialog box that appears, set Workflow Name and Description.
4. Click Create.
5. Open the workflow development DAG, and drag one data synchronization node and two MaxCompute SQL nodes to the DAG. After each node is created, click Commit.
6. By drawing lines between nodes, configure the data synchronization node as the ancestor node of ODPS\_SQL\_1, and enable ODPS\_SQL\_2 to depend on

ODPS\_SQL\_1. The following figure shows the dependency between the three nodes.



7. Configure each node separately. After the configuration is complete, click Save in the upper-left corner of the DAG.



**Note:**

This operation only saves the configuration of the current node (including node code and node attributes) to the development environment of the current workspace. However, this operation does not commit the node to the scheduling system.

## Commit the workflow

**Return to the workflow development DAG, and click Submit in the upper-left corner. In the dialog box that appears, select the required nodes, and commit them to the scheduling system.**

- 1. Return to the workflow development DAG.**
- 2. Click Submit in the upper-left corner.**
- 3. In the Commit dialog box that appears, select the required nodes, enter information in Description, and then select Ignore I/O Inconsistency Alerts.**
- 4. Click Commit. The message Committed successfully appears.**



### Note:

- **After nodes are committed to the scheduling system, they are ready for testing and deployment.**
- **Before running or deploying nodes to the development environment of another workspace, you must commit them to the scheduling system of the current workspace.**
- **A successfully committed workflow will be locked.**

## Deploy the workflow

**After you commit the workflow, the nodes in the workflow have entered the development environment. You need to deploy the configured nodes in the production environment because nodes in the development environment cannot be automatically scheduled.**

- 1. Click Deploy to open the deployment page.**
- 2. Select the nodes to be deployed, and click Add to List.**
- 3. Find To-Be-Deployed Node List in the upper-right corner, and click Deploy All.**
- 4. Open the View Deploy Tasks page, and view the deployed nodes.**

## Run the workflow in the production environment

- 1. After you deploy the workflow, click Operation Center in the upper-right corner.**
- 2. Choose Nodes > Auto Triggered Node, and select the required workflow.**
- 3. Right-click a node in the DAG, and choose Create Retroactive Instance > Current and Descendent Nodes Retroactively.**

4. Select a node that requires retroactive execution, specify the business date, and then click OK. The system automatically redirects to the Retroactive Instances page.
5. Click Refresh until the instance status is displayed as Successful.

### 2.3.4 Configure monitoring policies

This section describes how to configure monitoring policies.

Create a baseline

1. Create a baseline.

Navigate as follows: **Operation Center > Alarm > Baseline Manage**. Click **Create Baseline**.

2. Configure a baseline.

Specify the following configuration items: baseline name, workspace, owner, type, involved tasks, priority, committed time, and buffer threshold.

3. View baselines.

A few minutes after you create the baseline, you can view the status of involved tasks on the Baseline tab.

View alerts

You can view the details of generated alerts for troubleshooting.

1. Navigate as follows: **Operation Center > Alarm > Alarm Info**.
2. You can search for alerts by rule ID or name, recipient, alert time, alert method, and rule type.
3. Click View to check the alerts that match the criteria.

### 2.3.5 Create a resource

MaxCompute provides APIs for you to read and use resources. Currently, the following types of MaxCompute resources are available: File, Archive, JAR, and Python. This topic describes how to create a JAR resource as an example.

Procedure

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.

3. In the Create Workflow dialog box that appears, set Workflow Name and Description. Then, click Create.
4. Expand the created workflow in the left-side navigation pane. Right-click Resource, and choose Create Resource > JAR.
5. In the Create Resource dialog box that appears, specify a resource name according to the naming convention, and set the resource type to JAR. Select a local JAR package, and click OK.

**Note:**

- If the selected JAR package has been uploaded from the MaxCompute client, clear Upload to MaxCompute. If you do not clear it, an error will occur during the upload process.
- The resource name can be different from the name of the uploaded file.
- Convention for naming resources: A resource name can contain letters (case insensitive), digits, underscores (\_), and periods (.). It must be 1 to 128 characters in length. A JAR resource name must end with .jar.

6. After the resource is uploaded, click Submit.
7. In the Commit Node dialog box that appears, enter information in Description and click Ok.
8. When the message Committed successfully appears, click Deploy in the upper-right corner to deploy the resource. For more information, see [Publish nodes](#).

### 2.3.6 Import local files to MaxCompute

Alibaba Cloud DataWorks supports importing local files to MaxCompute. The maximum file size is 10 MB, and the file type must be TXT or CSV.

#### Procedure

1. Log on to DataWorks as a developer.
2. In the left-side navigation pane, click Data Analytics.
3. On the Data Analytics tab, click the Import icon in the tool bar.
4. Select a local file and click Open.

5. In the Import Local Data dialog box, **deselect** First Line as Field Names, specify other parameters, and click Next.

The following table describes the parameters in the Import Local Data dialog box.

| Parameter                 | Description                                                                                                       |
|---------------------------|-------------------------------------------------------------------------------------------------------------------|
| Selected Files            | The name of the selected local file to be imported.                                                               |
| Delimiter                 | The options include a comma (,), semicolon (;), vertical bar ( ), number sign (#), ampersand (&), space, and tab. |
| Original Character Set    | The options include GBK, UTF-8, CP936, and ISO-8859.                                                              |
| Import First Line         | The first row of data to be imported. You can select from 1 to 11.                                                |
| First Line as Field Names | It is selected by default.                                                                                        |

6. Select a MaxCompute table from the current workspace to which data is imported, and specify field settings.



**Note:**

Fuzzy match is supported for table names.

7. Click Import.

## 2.3.7 Use Spark

This section describes how to use Spark in DataWorks.

### Procedure

1. Right-click the Business Flow on the Data Analytics tab, and select Create Business Flow.
2. Right-click the Data Analytics folder, and choose Create Data Analytics Node > ODPS SQL.
3. In the new business flow folder, right-click the Resource folder, and choose Create Resource > JAR.
4. The Create Resource dialog box appears. Specify a resource name according to the naming convention, set the resource type to JAR, select a local JAR package, and click Submit.



## 5. Upload a configuration file.

It is not required that the configuration file of an ODPS SQL node contain MaxCompute settings. DataWorks can automatically obtain these settings from MaxCompute.

```
spark.hadoop.odps.project.name
spark.hadoop.odps.access.id
spark.hadoop.odps.access.key
spark.hadoop.odps.end.point
```

## 6. Right-click the added JAR resource and select Insert Resource and enter Spark statements.

### Sample code

#### • Java Spark

```
--@resource_reference{"spark.conf"}
--@resource_reference{"spark_examples.jar"}
jar -classpath
/opt/taobao/tbdpapp/spark-on-odps/2.1.0/__spark_libs__.zip,
spark_examples.jar
com.aliyun.odps.cupid.tools.OdpsCltWrapper
"--class" com.aliyun.odps.spark.examples.SparkPi
"--properties-file" spark.conf
"--conf" spark.yarn.archive=/opt/taobao/tbdpapp/spark-on-odps/2.1.
0/__spark_libs__.zip
"--master" yarn-cluster
spark_examples.jar
```

#### • Python Spark

```
##@resource_reference{"odps_table_rw.py"}
##@resource_reference{"spark.conf"}
##@resource_reference{"odps.zip"}
export PYSPARK_ARCHIVES_PATH=/opt/taobao/tbdpapp/spark-on-odps/2
.1.0/python/lib/py4j-0.10.4-src.zip,/opt/taobao/tbdpapp/spark-on-
odps/2.1.0/python/lib/pyspark.zip,odps.zip
java -cp /opt/taobao/tbdpapp/spark-on-odps/2.1.0:/opt/taobao
/tbdpapp/spark-on-odps/2.1.0/jars/* org.apache.spark.deploy.
SparkSubmit \
"--properties-file" spark.conf \
"--jars" /opt/taobao/tbdpapp/spark-on-odps/2.1.0/cupid/odps-spark-
datasource-2.0.4.jar \
"--conf" spark.yarn.archive=/opt/taobao/tbdpapp/spark-on-odps/2.1.
0/__spark_libs__.zip \
"--master" yarn-cluster \
odps_table_rw.py
```



### Note:

- You can find the address of the uploaded file from [odps.zip](#).

- The address of the Python file, for example, `odps_table_rw.py`, which includes the entry point, is [https://github.com/aliyun/aliyun-cupid-sdk/blob/master/examples/spark-examples/src/main/python/odps\\_table\\_rw.py](https://github.com/aliyun/aliyun-cupid-sdk/blob/master/examples/spark-examples/src/main/python/odps_table_rw.py).
- If shell nodes are used, the configuration file must contain MaxCompute settings.

## 2.3.8 Submit and publish nodes

In standard DataWorks workspaces, you can submit and publish nodes to the production environment and then run corresponding node tasks. Only standard workspaces support publishing nodes.

### Procedure

1. **Submit a node.** DataWorks automatically checks node dependencies. If no error is found, the node is successfully submitted.
  - a) Click **Submit** on the business flow DAG tab.
  - b) Select nodes to be submitted, and click **OK**.
  - c) Wait until the nodes are successfully submitted.
2. **On the Deploy page, publish the nodes to the production environment.**
  - a) Click **Deploy** in the upper-right corner.
  - b) Select all the submitted nodes and click **Add to Nodes to Publish**.
  - c) Click **View Nodes to Publish**. On the tab that appears, click **Publish All**.
  - d) You can view node tasks on the **Operation Center > Cycle Task** page. On this page, you can also perform tests, initiate retroactive task execution, pause nodes, and configure monitoring rules.

## 2.4 Data analytics

### 2.4.1 Solution

In a DataWorks workspace, you can group multiple business flows in a solution.

Workspace > Solution > Business flow

In DataWorks V2.0, each business flow consists of various nodes, which enables business-oriented code development. You can put one or more business flows into one solution so that you can manage them as a whole. You can put nodes, business flows, and solutions into separate workspaces.

- **Workspaces are the organizational unit for code, member, role, and permission management. All nodes, business flows, and solutions in a workspace can be collaboratively developed and managed by workspace members.**
- **Solutions have the following advantages:**
  - **You can group multiple business flows in a solution.**
  - **You can add a business flow to multiple solutions.**
  - **All solutions in a workspace can be collaboratively developed and managed by workspace members.**
- **Business flows bring the following advantages:**
  - **Business flows facilitate business-oriented code development. Nodes in a business flow are organized by type. Hierarchical directory structure is supported. We recommend that you limit the creation of directories and sub-directories to four levels.**
  - **You can view and optimize each entire business flow from the business perspective.**
  - **You can view each business flow in a DAG.**
  - **You can publish each entire business flow at a time. With the Administration service, you can also manage each business flow as a whole.**

#### Collaborative development

If you double-click a solution in the left-side pane, the solution details tab appears and the left-side pane only displays business flows in the solution.

1. **Jump to the DataStudio page and create a solution.**
2. **Add business flows to the solution.**
3. **Right-click the created solution, and select Solution Dashboard to view the added business flows. You can also modify the solution on the tab that appears.**
4. **Double-click the solution. The directory of this solution appears in the left-side navigation pane. You can edit business flows that form this solution. For more information, see [Description](#).**

## 5. Jump to another page.

- **Click Publish.** The Task Publish page appears, listing the nodes in the current solution that are in the To Be Published status.
- **In the left-side navigation pane, hover over a solution and click the Administration icon to go to the Operation Center > Cycle Instance tab.** This page that appears displays all recurring task instances created from nodes in the current solution.

You can add a business flow to one or more solutions. After you add a business flow to multiple solutions, if the business flow is changed in a solution, the changes take effect to all the solutions that contain the business flow.

## 2.4.2 SQL coding guidelines and specifications

### SQL coding guidelines

The SQL coding guidelines are as follows:

- **Ensure that the code is comprehensive.**
- **Ensure that code lines are clear, neat, well-organized, and structured.**
- **Consider the optimal execution speed during SQL coding.**
- **Provide comments whenever necessary to enhance the readability of your code.**
- **The guidelines impose non-mandatory constraints on the coding behavior of developers. In practice, understandable deviations are allowed when developers obey general rules. As the guide to daily code development, the guidelines are continuously improved and enriched.**
- **Use lowercase letters for all keywords and reserved words. Keywords and reserved words include select, from, where, and, or, union, insert, delete, group, having, and count.**
- **In addition to keywords and reserved words, other code such as field names and table alias must be in lowercase.**
- **A unit of indentation contains four spaces. All indentations must be the integral multiple of an indentation unit. The code is aligned according to its hierarchy.**
- **The `select *` operation is prohibited. The column name must be specified for all operations.**
- **Matching opening and closing parentheses must be placed in the same column.**

## SQL coding specifications

The SQL coding specifications are as follows:

- **Code header**

The code header contains information such as the subject, description, author, and date. Reserve a line for change log and a title line so that later users can add change records. Note that each line contains no more than 80 characters. The header template is as follows:

```
-- MaxCompute(ODPS) SQL
--

-- ** Subject: Transaction
-- ** Description: Transaction refund analysis
-- ** Author: Youma
-- ** Created on: 20170616
-- ** Change log:
-- ** Modified on Modified by Content
-- yyymmdd name comment
-- 20170831 Wuma Add a comment on the biz_type=1234 transaction
--

```

- **Field arrangement**

- Use a line for each field that is selected for the SELECT statement.
- Reserve one unit of indentation between the SELECT word and the first selected field. That is, the first field is two indentions away from the line start.
- Start each of the other field names in a new line with two units of indentation and a comma (,).
- Place the comma (,) between two fields right before the second field.
- Place the AS statement in the same line as its corresponding field. Keep AS statements of multiple fields in the same column.

```
select channel_id as channel_id
 ,trade_channel_desc as trade_channel_desc
 ,trade_channel_edesc as trade_channel_edesc
 ,inst_date as inst_date
 ,trade_iswap as trade_iswap
 ,channel_type as channel_type
 ,channel_second_desc as channel_second_desc
from (
```

- **Clause arrangement for an INSERT statement**

Arrange the clauses of an INSERT statement in the same line.

- **Clause arrangement for a SELECT statement**

The clauses such as FROM, WHERE, GROUP BY, HAVING, ORDER BY, JOIN, and UNION in a SELECT statement must be arranged according to the following requirements:

- Use a line for each clause.
- Ensure that the clauses are left aligned with the SELECT statement.
- Use two units of indentation between the first letter of a clause and its content.
- Keep the logical operators such as AND and OR in a WHERE clause left aligned with WHERE.
- If the length of a clause name exceeds two units of indentation such as ORDER BY and GROUP BY, add a space between the clause name and its content.

```
select trim(channel) channel
 ,min(id) id
from ods_trd_trade_base_dd
where channel is not null
and dt = ${tmp_uuuuumdd}
and trim(channel) <> ''
group by trim(channel)
order by trim(channel)
```

- **Spacing before and after operators**

Reserve one space before and after each arithmetic operator and logical operator. Keep all the operators in the same line unless the length of the code exceeds 80 characters.

```
select trim(channel) channel
 ,min(id) id
from ods_trd_trade_base_dd
where channel is not null
and dt = ${tmp_uuuuumdd}
and trim(channel) <> ''
group by trim(channel)
order by trim(channel)
```

- **SELECT CASE statements**

The SELECT CASE statement is used to evaluate the value of a variable. Correct compiling of the SELECT CASE statement helps improve the readability of the code.

Observe the following conventions when compiling the CASE statement:

- Write the WHEN clause one unit of indentation after the CASE statement in the same line.
- Use a line for each WHEN clause. Wrap a line if the clause is too long.
- The CASE statement must contain the ELSE clause. The ELSE clause must be aligned with the WHEN clause.

```
, case when p1.trade_from = '3008' and p1.trade_email is null then 2
 when p1.trade_from = '4000' and p1.trade_email is null then 1
 when p9.trade_from_id is not null then p9.trade_from_id
end as trade_from_id
,pl.trade_email as partner_id
```

- **Nested query**

Nested queries are often used in implementing the extract, transform, and load (ETL) process of data warehouse systems. Therefore, it is essential to arrange the code in a hierarchical manner. Example:

```
select p.channel
from (
select s1.channel
from (
select trim(channel) as channel
 ,min(id) as id
from ods_trd_trade_base_dd
where channel is not null
and dt = ${tmp_yyyymmdd}
and trim(channel) <> ''
group by trim(channel)
) s1
left outer join
 dim_trade_channel s2
on s1.channel = s2.trade_channel_edesc
where s2.trade_channel_edesc is null
order by id
) p
;
```

- **Table alias**

- Specify an alias for each table. Once an alias is defined for a table in a SELECT statement, use the alias whenever you reference the table in the statement. Use a simple and concise alias and avoid using keywords.
- We recommend that you define the table aliases by using letters in alphabetical order.
- In the nested query, levels 1 to 4 of SQL statements are named part, segment, unit, and detail, which are abbreviated as P, S, U, and D. You can also use a, b, c, and d to represent levels 1 to 4. To differentiate multiple clauses at the same level, add numbers such as 1, 2, 3, and 4 after the letter that represents the level. Add a comment for a table alias if required.

```
select p.channel
 ,rownumber() order_id
from (
 select s1.channel
 ,s1.id
 from (
 select trim(channel) as channel
 ,min(id) as id
 from ods_trd_trade_base_dd
 where channel is not null
 and dt = ${tmp_yyyymmdd}
 and trim(channel) <> ''
 group by trim(channel)
) s1
 left outer join
 dim_trade_channel s2
 on s1.channel = s2.trade_channel_edesc
 where s2.trade_channel_edesc is null
 order by id
) p
;
```



- SQL comments

```

-- STEP1: Clean up data partition on tmp_dws_tbd_alijr_user_relation_dd_5
-- that day.

```

- Add a comment for each SQL statement.
- Use a separate line for the comment of each SQL statement and place the comment in front of the SQL statement.
- Place the comment of a field right after the field.
- Add comments for clauses that are difficult to understand.
- Add a comment for important code.
- If a statement is long, we recommend that you add comments based on the purposes of each segment.
- (Required) Add a description for a constant or variable. (Optional) Add a comment to explain the valid value range.

## 2.4.3 Business flows

### 2.4.3.1 Description

In DataStudio, you can organize nodes in a business flow. DataStudio provides you with a directed acyclic graph (DAG) for nodes in each business flow. DataStudio also provides professional tools and supports administrative operations for business flows, which promotes intelligent development and management.

#### Node organization

You can configure one or more types of compute engines for a single workspace. A workspace contains multiple business flows. Each business flow is a collection of various nodes that are associated with each other. You can view each business flow in an automatically generated DAG. Supported node types include data integration, data analytics, table, resource, function, algorithm, and operation flow.

Each node is stored in a folder. You can create subfolders in each folder. However, for easier management, we recommend that you create a maximum of four folder levels. If more than four folder levels are required, we recommend that you split the business flow to two or more and add the split business flows to the same solution.

#### Business flow nodes

- **Data integration:** For more information, see [Data synchronization node](#).

- **Data analytics:** For more information, see [Node types](#).
- **Table:** For more information, see [Tables](#).
- **Resources:** For more information, see [Introduction to resources](#).
- **Functions:** For more information, see [Create a function](#).

You can double-click a business flow folder to view the DAG that displays the dependencies between nodes in the business flow.

#### Business flow dashboard

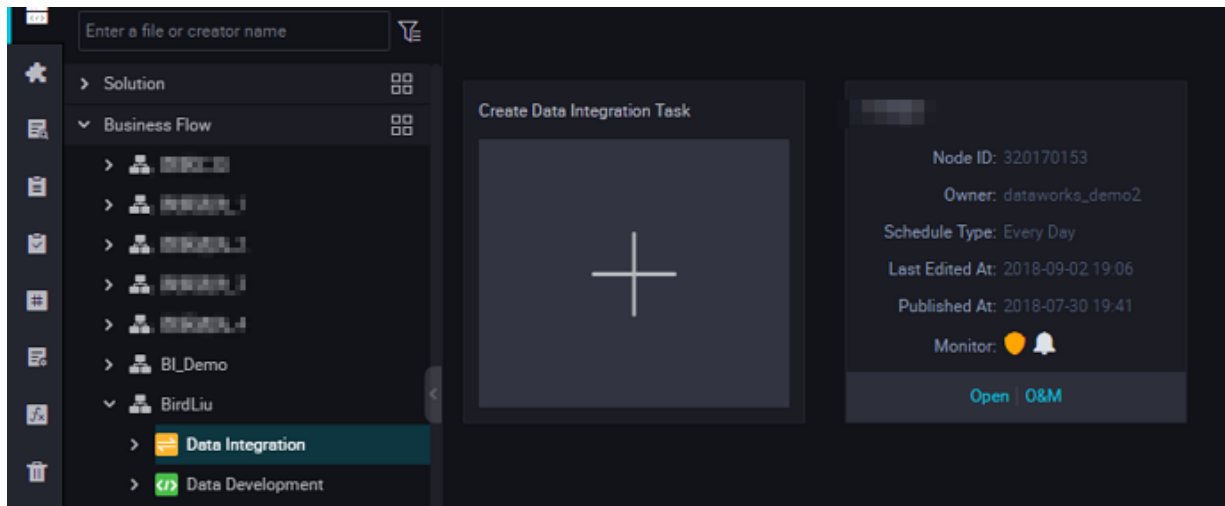
To view all business flows, click the Business Flow folder and select All Business Flows, or click the Business Flow icon next to the Business Flow folder.

#### Kanban for each node type

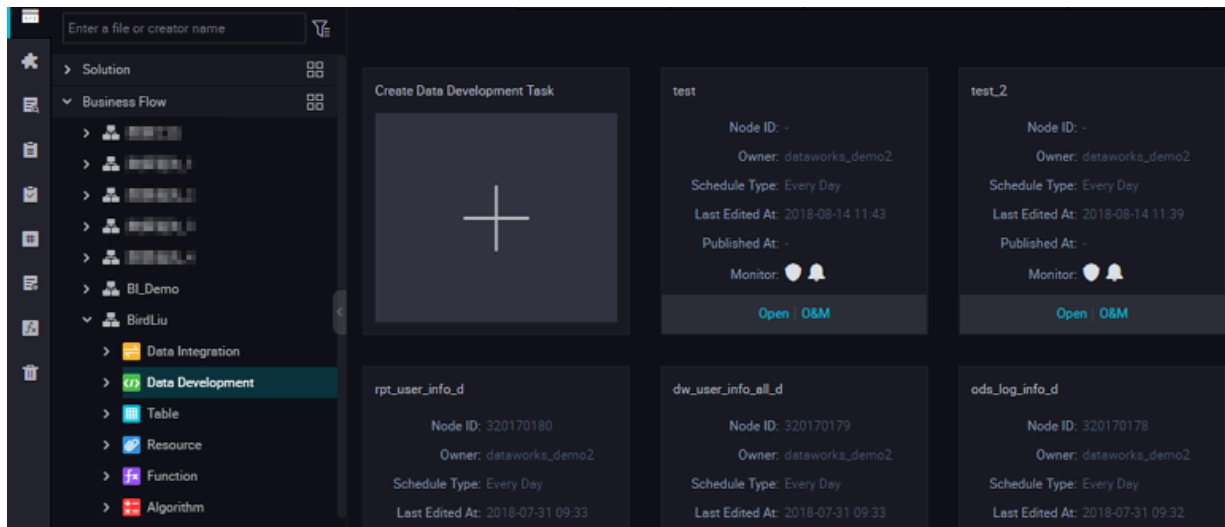
In a business process, a kanban is provided for each type of nodes. Each node is presented by a card with administration suggestions provided.

For example, the card of each data analytics node provides two indicators showing that whether baseline-based monitoring and event notification are enabled for the node. To open a kanban for a node type, you can right-click the folder of the node type and select Kanban, or double-click the folder.

#### Kanban of data integration nodes



## Kanban of data analytics nodes



## Create a business flow

On the Data Analytics page, right-click the Business Flow folder and select Create Business Flow.

## 2.4.3.2 Resource

If your code or function requires resource files such as .jar files, you can upload resources to your workspace and reference them.

If the existing built-in functions of the system do not meet your requirements, DataWorks enables you to create user-defined functions (UDFs) and customize processing logic. You can upload the required JAR package to the workspace so that you can reference it when creating UDFs.

The resources that you can upload to MaxCompute include text files, MaxCompute tables, Python code, and compressed packages in .zip, .tgz, .tar.gz, .tar, and .jar formats. You can read or use these resources when running UDFs or MapReduce.

MaxCompute provides APIs for you to read and use resources. Currently, the following types of MaxCompute resources are available:

- File
- Archive: compressed files identified by the resource name extension. Supported file types include .zip, .tgz, .tar.gz, .tar, and .jar.
- JAR: compiled Java JAR packages.

- **Python:** the Python code you have written. It is used for registering a Python UDF.

Currently, DataWorks only supports adding JAR and file resources in a visualized manner. Follow similar steps to create JAR and file resources. The differences are described as follows:

- To add a JAR resource, you need to compile the Java code in the offline Java environment, compress the code into a JAR package, and upload the package as the JAR resource to DataWorks.
- To add a file resource that is smaller than or equal to 500 KB in size, you can directly create and edit it in the DataWorks console.
- To add a file resource that is larger than 500 KB in size, you can select Larger than 500 KB and upload a local file.

**Note:**

The resource package for upload cannot exceed 30 MB.

Create a JAR resource

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
3. In the Create Workflow dialog box that appears, set Workflow Name and Description. Then, click Create.
4. Expand the created workflow in the left-side navigation pane. Right-click Resource, and choose Create Resource > JAR.
5. In the Create Resource dialog box that appears, specify a resource name according to the naming convention, and set the resource type to JAR. Select a local JAR package, and click OK.

**Note:**

- If the selected JAR package has been uploaded from the MaxCompute client, clear Upload to MaxCompute. If you do not clear it, an error will occur during the upload process.
- The resource name can be different from the name of the uploaded file.

- **Convention for naming resources:** A resource name can contain letters (case insensitive), digits, underscores (\_), and periods (.). It must be 1 to 128 characters in length. A JAR resource name must end with .jar.

6. After the resource is uploaded, click Submit.
7. In the Commit Node dialog box that appears, enter information in Description and click Ok.
8. When the message Committed successfully appears, click Deploy in the upper-right corner to deploy the resource. For more information, see [Publish nodes](#).

#### Reference and download resources

- For more information about how to reference resources for functions, see [Create a function](#).
- For more information about how to reference resources for nodes, see [ODPS MR nodes](#).

To download a resource, double-click Resource in the left-side navigation pane. In the resource list, select the required resource, and click Download.

### 2.4.3.3 Create a function

DataWorks provides the SQL computing function. You can use the system built-in SQL functions in MaxCompute SQL for data analysis and computing. This topic describes how to create a Java user-defined function (UDF).

#### Context

Click Built-in Functions in the navigation bar on the left of the DataStudio page. A list of built-in functions appears.

You can also define your own functions, that is, UDFs, in Java and Python 2. UDFs are used in the same way as the built-in functions of MaxCompute SQL.

#### Procedure


1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
3. In the Create Workflow dialog box that appears, set Workflow Name and Description. Then, click Create.
4. Create a JAR resource. For more information, see [Create a resource](#).

**5. Create a function.**

- a) Expand the created workflow in the left-side navigation pane. Right-click Function, and select Create Function.
- b) In the Create Function dialog box that appears, enter a name in Function Name, and click Commit.

**6. Register the function.**

- a) Set parameters in the Register Function dialog box.

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Function Type         | The type of the function.                                                                                                                                                                                                                                                                                                                                                    |
| Function Name         | The name of the function, that is, the name used to reference the function in SQL. The function name must be globally unique and cannot be modified after the function is registered.                                                                                                                                                                                        |
| Class Name            | <div>The name of the class for implementing the function. This parameter is required.</div> <div> <b>Note:</b><br/>If the resource type is Python, the following class name format is used: Python resource name.Class name. Do not include the .py extension in the resource name.</div> |
| Resources             | The list of resources. You can search for existing resources in the current workspace in fuzzy search mode. This parameter is required.                                                                                                                                                                                                                                      |
| Description           | The description of the function.                                                                                                                                                                                                                                                                                                                                             |
| Expression Syntax     | The instructions on how to use the function.                                                                                                                                                                                                                                                                                                                                 |
| Parameter Description | The description of supported input and output parameter types.                                                                                                                                                                                                                                                                                                               |

- b) Click Submit to complete the registration of the function.

**Note:**

You must develop and compile functions in compliance with the MaxCompute UDF framework.

- By default, MaxCompute UDFs support Java 1.7 and Python 2.7.
- Java UDFs must inherit from the `com.aliyun.odps.udf.UDF` class.

- A Python UDF must have its signature specified through annotation.

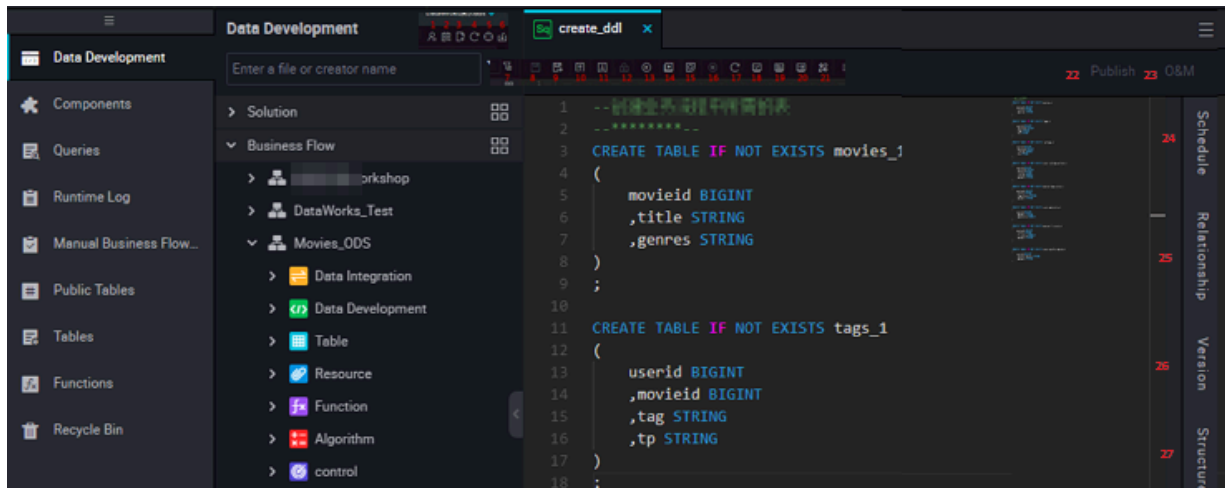
```

/****-----Sample of specifying the signature of a Python UDF
through annotation-----****/
from odps.udf import annotate
@annotate("bigint,bigint->bigint")
/****-----****/

```

## 2.4.4 Console features

### 2.4.4.1 Wizard



Wizard functions are described as follows:

| No. | Icon or Tab        | Description                                                                   |
|-----|--------------------|-------------------------------------------------------------------------------|
| 1   | My Files           | View your own nodes.                                                          |
| 2   | Search Code        | Search for a node or a code segment.                                          |
| 3   | Create (+)         | Create a solution, business flow, folder, node, table, resource, or function. |
| 4   | Refresh            | Refresh the directory tree.                                                   |
| 5   | Locate             | Locate the position of the selected file.                                     |
| 6   | Import             | Import local data to an online table. Pay attention to the encoding.          |
| 7   | Filter             | Filter nodes based on specified conditions.                                   |
| 8   | Save               | Save the code.                                                                |
| 9   | Save as Query File | Save the code in a node, which is listed on the Query tab.                    |
| 10  | Submit             | Submit the node.                                                              |
| 11  | Submit and Unlock  | Submit and unlock the node.                                                   |

| No. | Icon or Tab                                        | Description                                                                                             |
|-----|----------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| 12  | Steal Lock                                         | Edit a node that is not owned by you.                                                                   |
| 13  | Run                                                | Run the code.                                                                                           |
| 14  | Run with Parameters                                | Run the code after specifying parameters.                                                               |
| 15  | Precompile                                         | Precompile the parameters in the node.                                                                  |
| 16  | Stop                                               | Stop running the code.                                                                                  |
| 17  | Reload                                             | Reload the code. The code will be restored to the version last saved, and unsaved changes will be lost. |
| 18  | Run Smoke Test in Development Environment          | Test the code in the development environment.                                                           |
| 19  | View Smoke Test Log in Development Environment     | View the log of the node that runs in the development environment.                                      |
| 20  | Go to Scheduling System of Development Environment | Navigate to the Administration service of the development environment.                                  |
| 21  | Format                                             | Format the code based on keywords to avoid excessively long code in single lines.                       |
| 22  | Publish                                            | Redirect to the node publishing page. You can publish some or all nodes to the production environment.  |
| 23  | Administration                                     | Navigate to the Administration service of the production environment.                                   |
| 24  | Schedule                                           | Configure the information such as schedule, parameters, and resource group for the node.                |
| 25  | Lineage                                            | View the lineage between tables and nodes.                                                              |
| 26  | Version                                            | View the published versions of the node.                                                                |
| 27  | Thumbnail View                                     | View the minimap of the node. The minimap is very useful for quick navigation and code understanding.   |



## 2.4.4.2 Version

The version tab displays all submission and publication records of the current node . You can check the status, change type, and description.




**Note:**

Only submitted nodes have submission and publication records.

Click Version on the right of the code editor to view submission and publication records.

| Parameter    | Description                                                                                                                                                                                                                                                                                |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File ID      | The ID of the current node.                                                                                                                                                                                                                                                                |
| Version      | The version. A version is created for each publication. V1 indicates version 1, V2 indicates version 2, and so on.                                                                                                                                                                         |
| Submitted By | The user who submitted the node.                                                                                                                                                                                                                                                           |
| Submitted At | The time when the version was submitted. If a version is submitted and then published, the value of Submitted At is updated to the time when the node is published. This column records the time when the version is last operated.                                                        |
| Change Type  | Operation history of the current node. It is set to Added if the node is first released, and set to Modified if the node is modified.                                                                                                                                                      |
| Status       | Operation status record of the current node.                                                                                                                                                                                                                                               |
| Description  | Change description of the current node when it is submitted . It facilitates other personnel to locate the related version when operating the node.                                                                                                                                        |
| Actions      | Two actions are available: View Code and Roll Back. <ul style="list-style-type: none"><li>• View Code: View the code of the current version.</li><li>• Roll Back: Roll back the node to the current version. After rolling back a node, you need to submit and publish it again.</li></ul> |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Compare   | <p>Compare code and parameters between two selected versions.</p> <p>Click View Details to view code and schedule changes.</p> <div> <b>Note:</b><br/>You can only compare two version, and cannot compare one or more than two versions at a time.</div> |

### 2.4.4.3 Structure

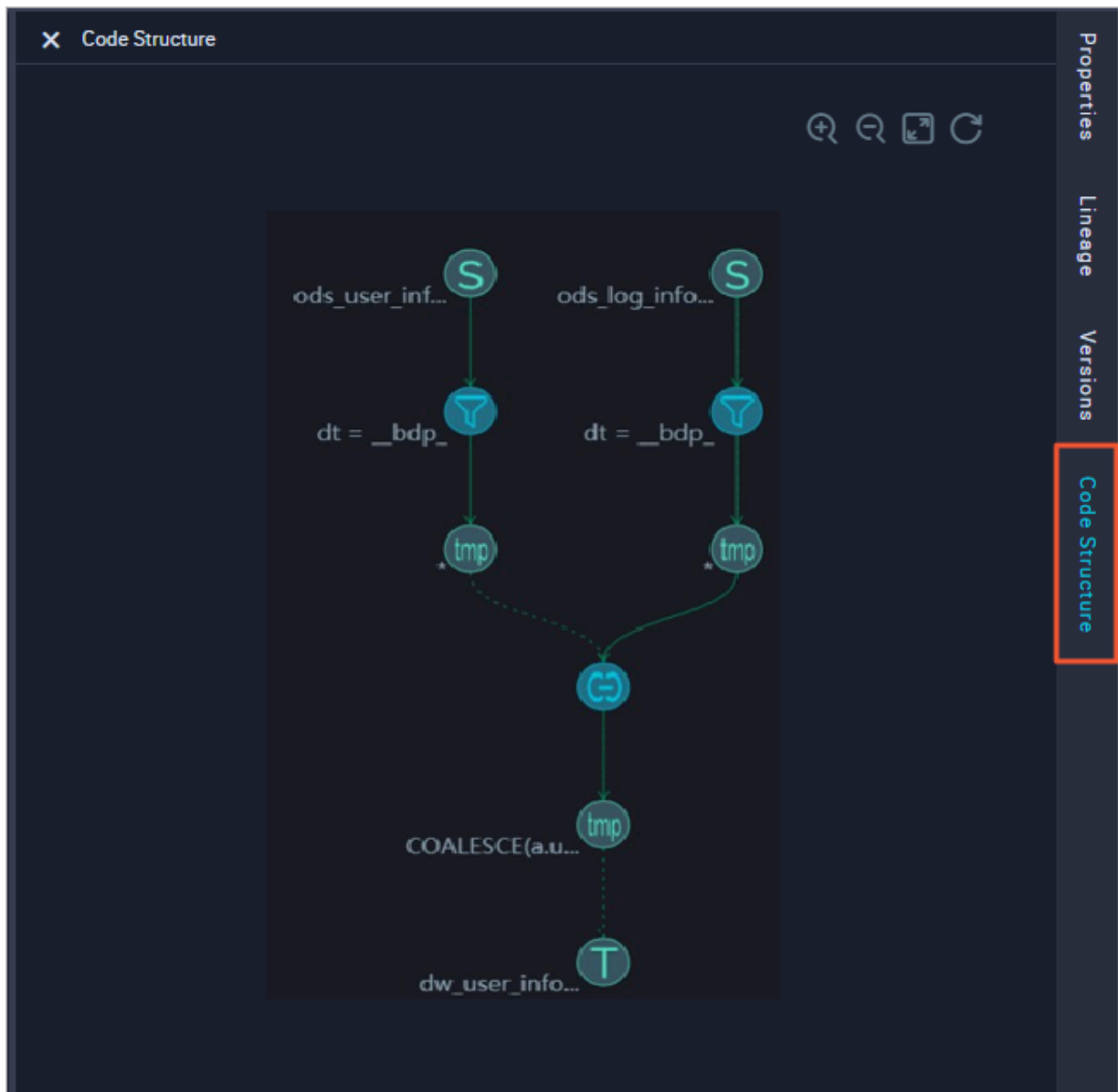
DataStudio can generate the SQL code structure based on the code specified, which helps you easily modify and review the code.

Structure

A sample of SQL code is provided as follows:

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
 , b.gender
 , b.age_range
 , b.zodiac
 , a.region
 , a.device
 , a.identity
 , a.method
 , a.url
 , a.referer
 , a.time
FROM (
 SELECT *
 FROM ods_log_info_d
 WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
 SELECT *
 FROM ods_user_info_d
 WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;
```

The following figure shows the structure generated from the sample code.



Hover over a circle, and a description occurs.

1. **Source table:** the table that you want to query by using a SELECT statement.
2. **Filter:** finds the partitions in the table that you want to query.
3. **First intermediate table (view):** a temporary table that stores the query results.
4. **Join:** joins the query results.
5. **Second intermediate table (view):** a temporary table that stores the joined query results. The temporary table is deleted three days after its initial creation.
6. **Destination table (insert):** the destination table to which the query results are inserted by using an INSERT OVERWRITE statement.

## 2.4.4.4 Lineage

**Lineage indicates the relationship between the current node and other nodes. You can check node dependency and lineage parsed from node code.**

### Dependency

**You can check whether the node dependency meets your expectation. If not, you can return to the Schedule page to reconfigure the node dependency.**

### Lineage parsed from node code

**The lineage is parsed based on the node code. For example:**

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
 , b.gender
 , b.age_range
 , B. flavdiac
 , a.region
 , a.device
 , a.identity
 , a.method
 , a.url
 , a.referer
 , a.time
FROM (
 VALUES
 From fig
 WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
 SELECT *
 FROM ods_user_info_d
 WHERE dt = ${bdp.system.bizdate}
) b
on a.uid = b.uid ;
```

**The lineage is parsed from the SQL statements shown in the following figure. The dw\_user\_info\_all\_d table is an output table when an SQL join is performed on the ods\_log\_info\_d and ods\_user\_info\_d tables.**



## 2.4.5 Node types

### 2.4.5.1 Node types

DataWorks supports seven node types.

#### Virtual nodes

A virtual node is an organizational node, which only supports dry-run scheduling and does not generate or change any data. It usually serves as the root node of a business flow. For more information, see [Virtual nodes](#).



#### Note:

You can configure an output table for a virtual node so that the output table can be used as an input table of another node. However, the virtual node does not process the table data.

#### ODPS SQL nodes

You can edit and debug SQL code in ODPS SQL nodes. DataWorks supports a great number of features such as cooperative code editing, records code versions, and supports automatic lineage capture. For more information, see [ODPS SQL nodes](#).

DataWorks uses MaxCompute projects as development and production environment workspaces. Therefore, the code must follow MaxCompute SQL syntax. The MaxCompute SQL syntax is similar to the Hive syntax, and is a subset of standard SQL syntax. However, MaxCompute projects cannot serve as databases, because

they are lack of many database features such as transactions, primary key constraints, and indexes.

#### ODPS MR nodes

MaxCompute supports MapReduce Java APIs. You can use the APIs to process MaxCompute data. You can create ODPS MR nodes and create recurring tasks from them. For more information, see [ODPS MR nodes](#).

#### Shell nodes

MaxCompute provides a Python SDK, which can be used to access MaxCompute.

DataWorks also supports PyODPS nodes, which are integrated with the Python SDK of MaxCompute. You can edit Python code in PyODPS nodes to access MaxCompute. For more information, see [PyODPS nodes](#).

#### SQL component nodes

SQL components are SQL templates that involve multiple input and output parameters. Each SQL component node involves one or more source tables. You can filters source table data, join source tables, and aggregate them. For more information, see [SQL component nodes](#).

#### Data synchronization nodes

Data synchronization nodes implement stable and efficient data synchronization between various data sources. You can use data synchronization nodes to synchronize your data from your business system to MaxCompute. For more information, see [Data synchronization node](#).

### 2.4.5.2 Data synchronization node

You only need to specify the name of the source table and that of the destination table to complete the configuration of a most basic node.

Data synchronization nodes support various data stores, including MaxCompute, MySQL, PostgreSQL, Oracle, MongoDB, DB2, Table Store, Table Store Stream, OSS, FTP, HBase, LogHub, HDFS, and Stream. For more information about supported data stores, see [Supported data sources](#).

When you enter a table name, the drop-down list displays all matched tables. Currently, only exact match is supported. Ensure that you specify a complete table name. Certain tables are labeled as unsupported if they are not supported by data

synchronization nodes. If you move the pointer over a table in the list, the details of the table appear, including the database, IP address, and owner. After you select a table, the column information is automatically filled in. You can also edit columns for non-MaxCompute tables, including moving, deleting, and adding columns.

**Note:**

DataWorks supports synchronizing partitioned tables of MaxCompute across multiple partitions.

Create a synchronization node

**For more information, see** [Create a synchronization node](#).

Configure the node schedule

**Click Properties on the right side, and set the relevant parameters. For more information, see** [Properties](#).

Commit the node

**After finishing the schedule configuration, click Save in the upper-left corner and commit the node to the development environment. After you commit the node, it is unlocked.**

Deploy the node

**For more information, see** [Deploy](#).

Test the node in the production environment

**For more information, see** [Recurring tasks](#).

### 2.4.5.3 ODPS SQL nodes

ODPS SQL nodes adopt a syntax similar to SQL, and enables processing TB-level data in batches in distributed mode. It is an online analytical processing (OLAP) application designed to deal with large amounts of data. Each ODPS SQL node can process ten thousands of transactions although the process takes a long time from preparation to submission for each job.

1. Log on to the DataWorks console.
2. Create a business flow.

**Click Data Analytics in the left-side navigation pane, right click the Business Flows folder, and select Create Business Flow.**

### 3. Create an ODPS SQL node.

Right-click the Data Analytics and choose **Create Data Analytics Node > ODPS SQL**.

### 4. Edit the code of the ODPS SQL node. The code must conform to the syntax.

**Example: Create a table, insert data into the table, and query data in the table.**

#### a. Create a table named test1.

```
CREATE TABLE IF NOT EXISTS test1
(id BIGINT COMMENT ' ',
 name STRING COMMENT ' ',
 age BIGINT COMMENT ' ',
 sex STRING COMMENT ' ');
```

#### b. Insert data into the table.

```
INSERT INTO test1 VALUES (1,'Zhang San',43,'Male');
INSERT INTO test1 VALUES (1,'Li Si',32,'Male');
INSERT INTO test1 VALUES (1,'Chen Xia',27,'Female');
INSERT INTO test1 VALUES (1,'Wang Wu',24,'Male');
INSERT INTO test1 VALUES (1,'Ma Jing',35,'Female');
INSERT INTO test1 VALUES (1,'Zhao Qian',22,'Female');
INSERT INTO test1 VALUES (1,'Zhou Zhuang',55,'Male');
```

#### c. Query data in the table.

```
select * from test1;
```

#### d. After you have specified the preceding SQL statements in the node editor, click the Run icon in the tool bar or press F8. DataWorks runs your SQL statements from top to bottom and generates logs.



#### Note:

**When DataWorks runs INSERT INTO statements, it logs !!! Warn!!!.**

**INSERT INTO statements can result in unexpected data duplication because DataWorks may rerun corresponding tasks although it does not rerun single INSERT SQL statements. We recommend that you avoid using INSERT INTO statements. If you continue to use INSERT INTO statements, we deem that you**



are aware of the associated risks and are willing to take the consequences of potential data duplication.

Do not rerun statements to ensure data insertion. If you rerun statements, data can be duplicated.

After you have configured the node, click the Save icon in the tool bar or press Ctrl+S to save the SQL code.

#### 5. Display query results.

DataWorks can display query results in a workbook and enables you to operate on the query results in the workbook.

You can also copy the query results to a local Excel file.

| Operation   | Description                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------|
| Hide Column | Select one or more columns and click Hide Column to hide the selected columns.                |
| Copy Row    | Select one or more rows in the left-side bar and click Copy Row to copy the selected rows.    |
| Copy Column | Select one or more columns in the top bar and click Copy Column to copy the selected columns. |
| Copy        | Click it to copy any selected cells.                                                          |
| Search      | After you click it, a search box appears in the upper right corner of the Results area.       |

#### 6. Schedule the node.

Click Schedule on the right of the code editor to open the Schedule tab. For more information, see [Schedule](#).

#### 7. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

#### 8. Publish the node.

For more information, see [Deploy](#).

#### 9. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

## 2.4.5.4 ODPS Spark node

DataWorks supports the ODPS Spark node type. This topic describes how to create and configure an ODPS Spark node.

### WordCount

1. Log on to the DataWorks console.
2. Create a workflow.
  - a. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
  - b. In the Create Workflow dialog box that appears, set Workflow Name and Description.
  - c. Click Create.
3. Create a JAR resource. For more information, see [Create a resource](#).
4. Create an ODPS Spark node.
  - a. Expand the created workflow in the left-side navigation pane. Right-click Data Analytics, and choose Create Data Analytics Node > ODPS Spark.
  - b. In the Create Node dialog box that appears, enter a name in Node Name.
  - c. Click Commit.
5. Configure the ODPS Spark node.
6. After the configuration is complete, click Save and Submit.

### Python

1. Expand the created workflow in the left-side navigation pane. Right-click Resource, and choose Create Resource > Python. Then, upload the prepared Python resource.
2. Create and configure an ODPS Spark node.
3. After the configuration is complete, click Save and Submit.

### Lenet (BigDL)

1. Upload the JAR package and data (a mnist.zip file of the archive resource type).
2. Create and configure an ODPS Spark node.
3. After the configuration is complete, click Save and Submit.

## 2.4.5.5 ODPS MR nodes

MaxCompute is interfaced with MapReduce. You can create and run ODPS MR nodes in DataWorks. You can also call MapReduce Java API operations to develop MapReduce programs for processing data in MaxCompute.

You need to upload, submit, and publish required resources before creating ODPS MR nodes.

Create a resource

1. Right-click the Business Flow folder on the Data Analytics tab, and select **Create Business Flow**.
2. In the new business flow folder, right-click the Resource folder, and choose **Create Resource > JAR**.
3. The Create Resource dialog box appears. Specify a resource name according to the naming convention, set the resource type to JAR, select a local JAR package, and click **Submit**.



**Note:**

- If the selected JAR package has been uploaded from the ODPS client, deselect **Upload to ODPS**. Otherwise, an error will occur during the upload process.
- The resource name can be different from the name of the uploaded file.
- Convention for naming resources: A resource name can contain letters (case insensitive), numbers, underscores (\_), and periods (.). It must be 1 to 128 characters in length. A JAR resource name must end with **.jar**, and a Python resource name must end with **.py**.

4. Click **Submit** to submit the resource to the development environment.
5. Publish the resource.

For more information, see [Deploy](#).

Create an ODPS MR node

1. Right-click the Business Flow folder on the Data Analytics tab, and select **Create Business Flow**.
2. In the new business flow folder, right-click the Data Analytics folder, and choose **Create Data Analytics Node > ODPS MR**.
3. Specify a node name in the Create Node dialog box and click **Submit**.

#### 4. Edit the code of the ODPS MR node. Double-click the new ODPS MR node.

##### Sample code:

```
jar -resources base_test.jar -classpath ./base_test.jar com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll
```

##### Code description:

- `-resources base_test.jar`: the name of the JAR resource.
- `-classpath`: the path of the JAR resource. You can right-click the resource and select **Insert Resource** to insert the path of the JAR package into the code.



##### Note:

Ensure that the configuration tab of the ODPS MR node is active when you attempt to insert a JAR package path.

- `com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll`: the main class in the JAR package. It must be consistent as specified in the JAR package.

If you use multiple JAR resources in a single ODPS MR node, separate each resource path with a comma (,) as follows: `-classpath ./xxxx1.jar,./xxxx2.jar`.

#### 5. Schedule the node.

Click **Schedule** on the right of the code editor to open the **Schedule** tab. For more information, see [Schedule](#).

#### 6. Submit the node.

After the node schedule information is complete, click the **Save** in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

#### 7. Publish the node.

For more information, see [Deploy](#).

#### 8. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

### 2.4.5.6 PyODPS nodes

DataWorks is integrated with the MaxCompute SDK for Python. You can specify Python code to process data in MaxCompute.



**Note:**

- The Python version of PyODPS nodes is 2.7.
- We recommend that you use SQL or DataFrame to process data, and do not use third-party tools such as Pandas.
- Each PyODPS node supports processing a maximum of 50 MB data, occupies a maximum of 1 GB memory. Otherwise, DataWorks terminates corresponding node tasks.

Create a PyODPS node

1. Click Data Analytics in the left-side navigation pane, right click the Business Flows folder, and select Create Business Flow.
2. Right-click the Data Analytics and choose Create Data Analytics Node > ODPS SQL.
3. Edit the code of the PyODPS node.

- a. Do not specify the entry point.

Each PyODPS node contain a global variable "odps" or "o", which is the entry point. Therefore, you do not need to manually specify the entry point.

```
print(odps.exist_table('pyodps_iris'))
```

- b. Run SQL statements.

You can query data by running MaxCompute SQL statements, and obtain the query results. You can use `execute_sql` and `run_sql` functions to create task instances.



**Note:**

You need to call certain functions to run statements that are not directly compatible with the MaxCompute console. For example, statements other than DDL and DML. To run GRANT and REVOKE statements, you need to call

**the `run_security_query` function. To run PAI commands, you need to call the `run_xflow` or `execute_xflow` function.**

```
o.execute_sql('select * from dual') # (Synchronous mode)
Blocked until the SQL statement is executed.
instance = o.run_sql('select * from dual') # Asynchronous mode.
print(instance.get_logview_address()) # Print the address of
LogView.
instance.wait_for_success() # Blocked until the SQL statement is
executed.
```

**c. Set runtime parameters.**

**You can use the `hints` parameter to set the runtime parameters. The type of the `hints` parameter is `DICT`.**

```
o.execute_sql('select * from PYODPS_iris', hints={'odps.sql.mapper
.split.size': 16})
```

**If you set the `sql.settings` parameter, you need to set runtime parameters each time you run the code.**

```
from odps import options
options.sql.settings = {'odps.sql.mapper.split.size': 16}
o.execute_sql('select * from PYODPS_iris') # The hints parameter
is automatically set based on global settings.
```

**d. Obtain SQL query results.**

**You can use the `open_reader` function to obtain query results if the SQL statement returns structured data.**

```
with o.execute_sql('select * from dual').open_reader() as reader:
for record in reader: # Process each record.
```

**You can also use this function to obtain raw query results if a `DESC` statement is run.**

```
with o.execute_sql('desc dual').open_reader() as reader:
print(reader.raw)
```



**Note:**

**If you use a custom time variable, you need to fix the variable to a time. PyODPS nodes does not support relative time variables.**

**4. Schedule the node.**

**Click `Schedule` on the right of the code editor to open the `Schedule` tab. For more information, see [Schedule](#).**

### 5. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

### 6. Publish the node.

For more information, see [Publish nodes](#).

### 7. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

## 2.4.5.7 Shell nodes

Shell nodes support standard shell syntax but not interactive shell syntax. Shell nodes can be run on the default resource group. If your shell nodes need to access an IP address or a domain name, add the IP address or domain name to the sandbox whitelist on the Project Management tab.

### Procedure

1. Right-click the Business Flow folder in the left-side Data Analytics pane, and select Create Business Flow.
2. Right-click the Data Analytics folder and choose Create Data Analytics Node > Shell.
3. Specify the node name, select the target folder, and click Submit.
4. Edit the code of the shell node.

Edit code in the editor of the shell node.

If you need to use relative time parameters, use the following statement:

```
echo "$1 $2 $3"
```



#### Note:

Separate parameters with a space character. For more information about relative time parameters, see [Parameter configuration](#).

### 5. Schedule the node.

Click Schedule on the right of the code editor to open the Schedule tab. For more information, see [Schedule](#).

## 6. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

## 7. Publish the node.

For more information, see [Publish nodes](#).

## 8. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

### Scenarios

If you use shell nodes to connect to databases:

- If the database is hosted on Alibaba Cloud and the region is China (Shanghai), whitelist the following IP addresses for the database:

10.152.69.0/24, 10.153.136.0/24, 10.143.32.0/24, 120.27.160.26, 10.46.67.156, 120.27.160.81, 10.46.64.81, 121.43.110.160, 10.117.39.238, 121.43.112.137, 10.117.28.203, 118.178.84.74, 10.27.63.41, 118.178.56.228, 10.27.63.60, 118.178.59.233, 10.27.63.38, 118.178.142.154, 10.27.63.15, and 100.64.0.0/8



#### Note:

If the database is hosted on Apsara Stack but the region is not China (Shanghai), we recommend that you connect to the database over a public network.

Alternatively, create an ECS instance in the same region as the database, add the ECS instance to DataWorks as a custom resource group, and run the shell node on the custom resource group.

- If the database is hosted on the premises, we recommend that you connect to the database over the Internet and whitelist the preceding IP address for the database.



#### Note:

If the shell node is run on a custom resource group, you also need to whitelist the server IP address of the custom resource group for the database.



### 2.4.5.8 SQL component nodes

SQL components are SQL templates that involve multiple input and output parameters. Each SQL component node involves one or more source tables. You can filters source table data, join source tables, and aggregate them.

#### Procedure

1. Click Data Analytics in the left-side navigation pane, right-click the Business Flow folder, and select Create Business Flow.
2. Right-click the Data Analytics folder and choose Create Data Analytics Node > SQL Component.
3. To improve development efficiency, you can create data analytics nodes based on components that are created by workspace and organization members.
  - The Workspace-Specific tab lists the components created by workspace members.
  - The Public tab lists the components created by organization members.

In the Create Node dialog box, select SQL Component as the node type and specify a node name.

Specify parameters for the component node.

Enter a parameter name, and set the parameter type to Table or String.

Specify three parameters for `get_top_n` in sequence.

If you have selected Table, specify the `test_project.test_table` as the output table.

4. Schedule the node.

Click Schedule on the right of the code editor to open the Schedule tab. For more information, see [Deploy](#).

5. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

6. Publish the node.

For more information, see [Publish nodes](#).

7. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

## Update the version of an SQL component

**When a new version is published for a component, you can update the version of the component used in your nodes to the latest version.**

**This improves efficiency. An example is provided as follows:**

**User A uses the version V1.0 of a component that belongs to user B. User B then submits a new version for the component V2.0. User A receives a notification of the new version. After comparing the two versions, user A can update the component version to V2.0 if required.**

**SQL component nodes are easy to upgrade because you develop them based on a template. To update the version update, ensure that your parameter configurations are valid, and adjust according to the version description. Then, save the node so that it is ready for publishing.**

## Component configuration wizard

**The component configuration wizard is described as follows:**

| No. | Feature              | Description                                                                                                            |
|-----|----------------------|------------------------------------------------------------------------------------------------------------------------|
| 1   | Save                 | Save the code for the node.                                                                                            |
| 2   | Steal Lock           | Edit a node that is not owned by you.                                                                                  |
| 3   | Submit               | Submit the component to the development environment.                                                                   |
| 4   | Publish Component    | Publish a component to the entire organization so that all members in the organization can view and use the component. |
| 5   | Parse I/O Parameters | Parse input and output parameters from the code.                                                                       |
| 6   | Precompile           | Edit custom and system parameters for the component.                                                                   |
| 7   | Run                  | Run the component in the development environment.                                                                      |
| 8   | Stop                 | Stop running the component.                                                                                            |
| 9   | Format               | Format the code based on keywords.                                                                                     |
| 10  | Parameters           | View and configure the component information, input parameters, and output parameters.                                 |
| 11  | Version              | View the published versions of the component.                                                                          |

| No. | Feature           | Description                            |
|-----|-------------------|----------------------------------------|
| 12  | Reference Records | View the use records of the component. |

### 2.4.5.9 Virtual nodes

A virtual node is an organizational node, which only supports dry-run scheduling and does not generate or change any data. It usually serves as the root node of a business flow.



**Note:**

You can configure an output table for a virtual node so that the output table can be used as an input table of another node. However, the virtual node does not process the table data.

Create a virtual node task

1. Click **Data Analytics** in the left-side navigation pane, right-click the **Business Flow** folder, and select **Create Business Flow**.
2. Right-click the **Data Analytics** folder and choose **Create Data Analytics Node > Virtual Node**.
3. Specify the node type as **Virtual Node**, enter a node name, select a destination folder, and click **Submit**.
4. You do not need to edit code for virtual nodes.
5. Schedule the node.

Click **Schedule** on the right of the code editor to open the **Schedule** tab. For more information, see [Schedule](#).

6. Submit the node.

After the node schedule information is complete, click the **Save** in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

7. Publish the node.

For more information, see [Deploy](#).

8. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

### 2.4.5.10 Cross-tenant collaboration node

Cross-tenant collaboration nodes are typically used to associate nodes from different tenants. Cross-tenant collaboration nodes are classified into sender nodes and receiver nodes.

#### Prerequisites

A sender node and its receiver node must use the same Cron expression. You can click **Properties** on the right side and view the Cron expression in the **Schedule** section.

#### Create a cross-tenant collaboration node

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the **Create** icon, and then choose **Control > Cross-Tenant Collaboration**.



**Note:**

Alternatively, you can navigate to the corresponding workflow, right-click **Control**, and then choose **Create Control Node > Cross-Tenant Collaboration**.

3. In the **Create Node** dialog box that appears, set the relevant parameters.
4. Click **Commit**.

#### Configure the cross-tenant collaboration node

1. On the **Cross-Tenant Collaboration** page, set the relevant parameters.

| Parameter                | Description                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Type                     | The type of the cross-tenant collaboration node. Two node types are available, which are <b>Sender</b> and <b>Receiver</b> .                                                                        |
| Location                 | The path of the cross-tenant collaboration node. The node path cannot be modified.                                                                                                                  |
| Collaborative Workspaces | The workspace name and Alibaba Cloud account of the peer node. This example sets the node type to <b>Sender</b> . Therefore, you need to enter the workspace name and account of the receiver node. |

2. After the sender node is created, follow the same procedure to create the receiver node under the account and workspace to which the receiver node belongs.

Set the node type to Receiver. Afterward, the information about available sender nodes appears. You must also set the Timeout parameter. This parameter indicates the timeout period of the receiver node after it starts running.

The sender node first sends a message to the message center. Once the message is delivered, the status of the sender node is set to successful. The receiver node continuously pulls messages from the message center. Once a message is received within the timeout period, the status of the receiver node is set to successful.

If the receiver node does not receive any messages within the timeout period, the receiver node fails. The lifecycle of a message is 24 hours.

Assume that a recurring instance was run on October 8, 2018. A message indicating the completion of the instance was then sent to the message center.

If you create a retroactive instance for the receiver node with the business date set to October 7, 2018, the status of the generated receiver node instance is set to successful.

3. After the configuration is complete, click Save and Submit.

#### 2.4.5.11 Assignment node

The assignment node is a special type of node. It allows you to assign values to output parameters by writing code in the node and passes them with the node context to descendant nodes for reference.

Create an assignment node

1. Open the DataStudio page, move the pointer over the Create icon, and then choose Control > Assignment Node.



**Note:**

Alternatively, you can navigate to the corresponding workflow, right-click Control, and then choose Create Control Node > Assignment Node.

2. In the Create Node dialog box that appears, set the relevant parameters.
3. Click Commit.

Write the value assignment logic

The assignment node has a fixed output parameter named **outputs**. You can view the information about this parameter in the **Parameters** section. You can use ODPS SQL, Shell, or Python to write code for assigning values to parameters. The value of each parameter can be computed after the corresponding code is run. Only one language can be specified for a single assignment node.



**Note:**

- The value of the **outputs** parameter is taken only from the output of the last line of the code.
  - For ODPS SQL, use the output of the **SELECT** statement in the last line.
  - For Shell, use the output of the **ECHO** statement in the last line.
  - For Python, use the output of the **PRINT** statement in the last line.
- The passed value of the **outputs** parameter is limited to 2 MB in size. If the output of the assignment statement exceeds this limit, the assignment node fails to run.

Use an assignment node as a parent node

An assignment node can be added as the parent node of a node. After setting the dependency, you can define the output of the assignment node as the input parameter of the child node in the node context. Then, reference the node in the code. In this way, the child node can obtain the specific values from the output of the assignment node.

Example of the assignment node

1. Create a workflow and then create the nodes.
2. When you configure an assignment node, the system displays the **outputs** parameter by default. Its value can be found in the **Parameters** section after you run the assignment node.
3. The **outputs** parameter of the parent node serves as the input parameter of the child node.

Run the assignment node



**Note:**

The preceding configuration parameters can take effect through data patching in Operation Center, but the parameters for operation test cannot take effect.

1. After a node is configured and submitted for scheduling, a running instance is generated on the next day.
2. When the node is running, you can view the input and output parameters in the node context, and click the link next to each parameter to view your input and output results.
3. In the operational logs, you can view the final output of the code in finalResult.

#### Summary



#### Note:

The value of the outputs parameter is the output of the last line of the code.

This section describes the general usage of ODPS SQL, Shell, and Python arrays. In the examples, the input parameter of a Shell node is used to generate output data.

- ODPS SQL: The output is a one-dimensional or two-dimensional array.

In this example, the query result is a two-dimensional array.

```
2,this is name6
1,this is name5
```

The following figure shows the output code in Shell.

```
echo ${input[0][0]};
echo ${input[0][1]};
echo ${input[1][0]};
echo ${input[1][1]};
echo ${input[0]};
```

The following figure shows the output.

```
2019-02-21 11:11:55 INFO --- Invoking Shell command line now ---
2019-02-21 11:11:55 INFO =====
2
this is name6
1
this is name5
2,this is name6
```

- **Shell:** The output is a one-dimensional array.

The following figure shows the output code in Shell.

```
#*****
echo ${input};
echo ${input[0]};
echo ${input[1]}
```

The following figure shows the output.

```
2019-02-21 11:59:13 INFO --- Invoking Shell command line now ---
2019-02-21 11:59:13 INFO =====
this is password,ok
this is password
ok
```

- **Python:** The output is a one-dimensional array.

The following figure shows the output code in Shell.

```
#*****#
echo ${input};
echo ${input[0]};
echo ${input[1]}
```

The following figure shows the output.

```
2019-02-21 12:07:13 INFO --- Invoking Shell command line now ---
2019-02-21 12:07:13 INFO =====
this is second python,ok
this is second python
ok
2019-02-21 12:07:13 INFO =====
```

### 2.4.5.12 Branch node

The branch node is a type of logical control node in DataStudio. It can define the branch logic and the direction of branches under different logical conditions.



**Note:**

Generally, branch nodes need to be used with assignment nodes.

Create a branch node

1. Go to the DataStudio page. Move the pointer over the Create icon and then choose **Control > Branch Node**.



**Note:**



Alternatively, you can navigate to the corresponding workflow, right-click **Control**, and then choose **Create Control Node > Branch Node**.

2. In the **Create Node** dialog box that appears, set the relevant parameters.
3. Click **Submit**.

Define the branch logic

1. After creating the branch node, add a branch on the **Definition** page.
2. Click **Add Branch**. In the **Branch Definition** dialog box that appears, specify **Condition**, **Associated Node Output**, and **Description**.

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Condition              | <ul style="list-style-type: none"><li>• You can only use Python comparison operators to define logical conditions for branch nodes.</li><li>• If the result of the expression is true when the node is running, the corresponding branch condition is met. Otherwise, the branch condition is not met.</li><li>• If the expression fails to be parsed when the node is running, the whole branch node fails.</li><li>• You can configure branch conditions with global variables and parameters defined in the node context. For example, the <code>\${Input}</code> parameter in the figure can be used as an input parameter of the branch node.</li></ul> |
| Associated Node Output | <ul style="list-style-type: none"><li>• The node output is used to configure dependencies for the child nodes of the branch node.</li><li>• If the branch condition is met, the child node corresponding to the node output is run. If the child node also depends on the output of other nodes, the statuses of these nodes need to be considered.</li><li>• If the branch condition is not met, the child node corresponding to the node output will not be run and will be set to the <b>Not Running</b> state.</li></ul>                                                                                                                                 |
| Description            | The description of the branch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

3. After the configuration is completed, click **OK**.

Configure node scheduling

After the branch condition is defined, the output names are automatically added to the **Outputs** column on the **Properties** tab. Then, you can associate child nodes with the branch node based on the output names.

**Note:**

The dependencies established by drawing lines between nodes in the editing panel of a workflow are not recorded on the Properties tab. You must manually enter these dependencies.

Example: Configure child nodes for a branch node

You can associate child nodes with different output results of a branch node to define the branches under different conditions. For example, in the workflow shown in the following figure, the branches Branch\_1 and Branch\_2 are both child nodes of the branch node.

Branch\_1 depends on the output autotest.fenzhi121902\_1.

Branch\_2 depends on the output autotest.fenzhi121902\_2.

Submit the workflow

Submit the workflow to Operation Center. In this example, the output of the branch node is autotest.fenzhi121902\_1, which is associated with Branch\_1.

- If a branch meets the specified condition, the branch is run. You can select the branch and view the running details in the Runtime Log section.
- If a branch does not meet the specified condition, the branch is skipped. You can select the branch and view related information in the Runtime Log section.

Supported Python comparison operators

In the following table, we assume that the value of the variable a is 10 and that of the variable b is 20.

| Comparison operator | Description                                          | Example                                              |
|---------------------|------------------------------------------------------|------------------------------------------------------|
| ==                  | Equal: checks whether two objects are equal.         | (a==b) returns False.                                |
| !=                  | Not equal: checks whether two objects are not equal. | (a!=b) returns True.                                 |
| <>                  | Not equal: checks whether two objects are not equal. | (a<>b) returns True. This operator is similar to !=. |

| Comparison operator | Description                                                                                                                                                                                                                                                                    | Example               |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| >                   | Greater than: checks whether the variable on the left side of the operator is greater than that on the right side.                                                                                                                                                             | (a>b) returns False.  |
| <                   | Less than: checks whether the variable on the left side of the operator is less than that on the right side. If the return result is 0 or 1, 0 represents False and 1 represents True. These two results are equivalent to the special variables True and False, respectively. | (a<b) returns True.   |
| >=                  | Greater than or equal to: checks whether the variable on the left side of the operator is greater than or equal to that on the right side.                                                                                                                                     | (a>=b) returns False. |
| <=                  | Less than or equal to: checks whether the variable on the left side of the operator is less than or equal to that on the right side.                                                                                                                                           | (a<=b) returns True.  |

### 2.4.5.13 MERGE node

This topic describes the definition of MERGE nodes and how to create a MERGE node and define the merging logic. An example is also provided to show the scheduling configuration and running details of a MERGE node.

#### Definition

- The MERGE node is a type of logical control node in DataStudio.
- A MERGE node can merge the running results of its parent nodes, regardless of their running statuses. It aims to fix the issue that a node that depends on the output of the child nodes of a branch node only starts to run after its parent nodes are successfully run.
- Currently, you cannot change the running status of a MERGE node. A MERGE node merges multiple child nodes of a branch node and sets the running status to Successful. To guarantee the proper running of a node that depends on the

output of the child nodes of a branch node, you can configure the node to directly depend on the MERGE node.

For example, the branch node C defines two logically exclusive branches C1 and C2. These two branches use different logic to write data to the same MaxCompute table. Assume that node B depends on the output of this MaxCompute table. To make sure that node B can run properly, you must use the MERGE node J to merge branches C1 and C2, and then set the MERGE node J as the parent node of node B. If node B directly depends on branches C1 and C2, one of the branches will fail to run because only one branch meets the branch condition each time the branch node runs. In this case, node B cannot be triggered as scheduled.

Create a MERGE node

1. Go to the DataStudio page. Move the pointer over the Create icon and then choose **Control > MERGE Nodes**.



**Note:**

Alternatively, you can navigate to the corresponding workflow, right-click **Control**, and then choose **Create Control Node > MERGE Nodes**.

2. In the Create Node dialog box that appears, set the relevant parameters.
3. Click **Submit**.

Define the merging logic

After creating the MERGE node, specify the branches to be merged for the node. Enter the output name or output table name of the child node of a branch node, and click the Add icon. You can view the running status in the execution results. Currently, the running statuses include Successful and Failed.

Click the Properties tab in the right sidebar to configure the scheduling properties of the MERGE node.

Example of the MERGE node

You can associate child nodes with different output results of a branch node to define the branches under different conditions. For example, in the workflow shown in the following figure, the branches Branch\_1 and Branch\_2 are both child nodes of the branch node.

Branch\_1 depends on the output autotest.fenzhi121902\_1.

**Branch\_2 depends on the output autotest.fenzhi121902\_2.**

Run the MERGE node

**If a branch meets the specified condition, the branch is run. You can select the branch and view the running details in the Runtime Log section.**

**If a branch does not meet the specified condition, the branch is skipped. You can select the branch and view related information in the Runtime Log section.**

**The child node of the MERGE node runs properly.**

#### 2.4.5.14 do-while node

**You can define mutually dependent nodes, including a loop decision node named end, on a do-while node. DataWorks repeatedly runs the nodes and exits the loop only when the end node returns False.**



**Note:**

**A loop can be repeated for a maximum of 128 times. If the loop count exceeds this limit, an error occurs.**

**The do-while node supports the ODPS SQL, Shell, and Python languages. If you use ODPS SQL, you can use a `case when` statement to evaluate whether the specified condition for exiting the loop is met.**

Simple example

**This section describes how to use a do-while node to repeat a loop five times and display the loop count each time the loop runs.**

- 1. On the DataStudio page, click Data Analytics in the left-side navigation pane.  
Move the pointer over the Create icon and choose Control > do-while.**
- 2. In the Create Node dialog box that appears, set the parameters and click Submit.**

### 3. Double-click the created do-while node and define the loop body.

The do-while node consists of the start, sql, and end nodes.

- The start node marks the startup of a loop and does not have any business effect.
- DataWorks provides the sql node as a sample business processing node. You must replace the sql node with your own business processing node, for example, a Shell node named Display loop count.
- The end node marks the end of a loop and determines whether to start the loop again. In this example, it defines the condition for exiting the loop for the do-while node.

The end node only assigns values True and False, indicating whether to start a loop again or exit the loop.

The `${dag.loopTimes}` variable is used in both the Display loop count node and the end node. It is a reserved variable of the system. This variable indicates the loop count and increments from 1. The internal nodes of the do-while node can directly reference this variable.

In the code shown in the preceding figure, the value of the `${dag.loopTimes}` variable is compared with 5 to limit the total number of loops. The value of the `${dag.loopTimes}` variable is 1 for the first loop, 2 for the second loop, and so on. When the loop runs for the fifth time, the value is 5. In this case, the conditional statement `${dag.loopTimes}<5` is False, and the do-while node exits the loop.

#### 4. Run the do-while node.

You can configure the scheduling settings for the do-while node as needed and submit it to Operation Center for running.

- **do-while node:** The do-while node is displayed as a whole node in Operation Center. To view the loop details about the do-while node, right-click the node and select View Internal Nodes.
- **Internal loop body:** This view is divided into three parts.
  - The left pane of the view lists the rerun history of the do-while node. A record is generated for each run of the whole do-while instance.
  - The middle pane of the view shows a loop record list. Each record corresponds to each run of the do-while node. The running status of the node for each run is also displayed.
  - The right pane of the view shows the details about the do-while node each time the loop runs. You can click a record in the loop record list to view the running status of the corresponding instance.

#### 5. View the running result.

Access the internal loop body. In the loop record list, click the record corresponding to the third run. The loop count is 3 in the runtime logs.

You can also view the runtime logs of the end node that are generated when the loop runs for the third time and for fifth time, respectively.

As shown in the preceding figures, the conditional statement  $3 < 5$  is True when the loop runs for the third time, while the conditional statement  $5 < 5$  is False when the loop runs for the fifth time. Therefore, the do-while node exits the loop after the fifth run.

Based on the preceding simple example, the do-while node works in the following process:

1. Run from the start node.
2. Run nodes in sequence based on the defined node dependencies.
3. Define the condition for exiting a loop for the end node.
4. Run the conditional statement of the end node after the loop ends for the first time.

5. Record the loop count as 1 and start the loop again if the conditional statement returns True in the runtime logs of the end node.
6. Exit the loop if the conditional statement returns False in the runtime logs of the end node.

#### Complex example

Besides the preceding simple scenarios, do-while nodes can also be used in complex scenarios where each row of data is processed in sequence by using a loop . Before processing data in such scenarios, make sure that:

- You have deployed a parent node that can export queried data to the do-while node. You can use an assignment node to meet this condition.
- The do-while node can obtain the output of the parent node. You can configure the context and dependencies to meet this condition.
- The internal nodes of the do-while node can reference each row of data. In this example, the existing node context is enhanced and the system variable `${dag.offset}` is assigned to help you reference the context of the do-while node.

This section describes how to use the do-while node to respectively display records 0 and 1 in two rows of the `tb_dataset` table each time the loop runs.

1. On the DataStudio page, click Data Analytics in the left-side navigation pane. Move the pointer over the Create icon and choose Control > do-while.
2. In the Create Node dialog box that appears, set the parameters and click Submit.
3. Double-click the created do-while node and define the loop body.
  - a. Create a parent node named Initialize dataset for the do-while node. The parent node generates a test dataset.
  - b. Click Properties in the upper-right corner to configure a dedicated context for the do-while node. Set Parameter Name to input and Value Source to the output of the parent node.
  - c. Type the code for the business processing node named Print each data row.
    - `${dag.offset}`: a reserved variable of DataWorks. This variable indicates the offset of the loop count to 1. The offset is 0 for the first run, 1 for the second run, and so on. The offset equals to the loop count minus 1.
    - `${dag.input}`: the context that you configure for the do-while node.As mentioned above, the do-while node is configured with the input



parameter, with Value Source set to the output of the parent node named Initialize dataset.

The internal nodes of the do-while node can directly use `${dag.${ctxKey}}` to reference the context. In this example, `${ctxKey}` is set to `input`.

Therefore, you can use `${dag.input}` to reference the context.

- `${dag.input[${dag.offset}]}`: The node Initialize dataset exports a table. DataWorks can obtain a row of data in the table based on the specified offset. The value of `${dag.offset}` increments from 0. Therefore, the returned results are `${dag.input[0]}`, `${dag.input[1]}`, and so on until all data in the dataset is returned.

- d. Define the condition for exiting the loop for the end node. As shown in the following figure, the values of the `${dag.loopTimes}` and `${dag.input.length}` variables are compared. If the value of `${dag.loopTimes}` is less than that of `${dag.input.length}`, the end node returns True and the do-while node continues the loop. Otherwise, the end node returns False and the do-while node exits the loop.



**Note:**

The system automatically sets the `${dag.input.length}` variable to the number of rows in the array specified by the input parameter based on the context configured for the do-while node.

4. Run the nodes and view the running result.

The loop count is less than the number of the rows when the loop runs for the first time. Therefore, the end node returns True and the loop continues. The loop count equals to the number of the rows when the loop runs for the second time. Therefore, the end node returns False and the loop stops.

## Summary

- Compared with the while, foreach, and do...while statements, a do-while node:
  - Contains a loop body that runs a loop before evaluating the conditional statement, providing the same function as the do...while statement. A do-

while node can also use the system variable `${dag.offset}` and the node context to implement the function of the foreach statement.

- Cannot achieve the function of the while statement because a do-while node runs a loop before evaluating the conditional statement.
- The do-while node works in the following process:
  1. Run nodes in the loop body starting from the start node based on node dependencies.
  2. Run the code defined for the end node.
    - Run the loop again if the end node returns True.
    - Stop the loop if the end node returns False.
- Method to use the context: The internal nodes of the do-while node can use `${dag.ctxKey}` to reference the context defined for the do-while node.
- System parameters: DataWorks automatically issues the following system variables for the internal nodes of the do-while node:
  - `${dag.loopTimes}`: the loop count, starting from 1.
  - `${dag.offset}`: the offset of the loop count to 1, starting from 0.

## 2.4.5.15 Custom node type

### 2.4.5.15.1 Overview

DataStudio supports default node types such as ODPS SQL and Shell nodes. You can also create custom node types to meet your requirements.

To create a custom node type, you need to create a custom wrapper and use it to define a custom node type.

Entry

1. Log on to the DataWorks console.
2. Click Node Market in the upper-right corner to go to the node configuration page.



**Note:**

**Only the workspace owner and administrators can access this page.**


View the list of wrappers

The Wrappers page displays all the wrappers you have created. You can click Create in the upper-right corner to create a custom wrapper.

The values displayed in the Latest Version, Version in Development, and Version in Production Environment columns for the created wrappers follow these rules:

- If a created wrapper has not been deployed, the values of both the Version in Development and Version in Production Environment columns are Not Deployed.
- If a wrapper has been deployed, the version and the deployment time appear in these columns.
- If a wrapper is under deployment, the values of both the Version in Development and Version in Production Environment columns are Deploying.

You can click Settings, View Versions, or Delete in the Actions column of each wrapper.

| Action        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Settings      | You can click Settings to configure the wrapper. The page that appears depends on the wrapper status. The Deploy in Production Environment page appears if the wrapper has been deployed in the production environment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| View Versions | <p>You can click View Versions to view all historical versions of the wrapper.</p> <ul style="list-style-type: none"><li>• <b>View:</b> You can click this button to view the settings of the selected version.</li><li>• <b>Roll Back:</b> You can click this button to roll back to the selected version. After you click this button, the system creates a new version for the wrapper. In the new version, the wrapper uses the basic settings and the resource file of the selected version. The new version number equals the latest version number among all the versions plus 1.</li><li>• <b>Download:</b> You can click Download to download the resource file of the selected version.</li></ul> |
| Delete        | <p>If an error occurs while a node type is using the wrapper, you need to delete the node type.</p> <div> <b>Note:</b><br/>Before deleting a wrapper, ensure that no node type is associated with the wrapper.</div>                                                                                                                                                                                                                                                                                                                                                                                                     |

#### Create a custom wrapper

A wrapper is the core processing logic of a node type. For example, after you compile an SQL statement in the editor for an ODPS SQL node and submit the

statement, the system calls the corresponding wrapper to parse and run the statement. You need to create a wrapper before creating a custom node type. Currently, only the Java programming language is supported.

The procedure of creating a wrapper includes four steps: specify settings for the wrapper, deploy the wrapper in the development environment, test the wrapper in the development environment, and deploy the wrapper in the production environment. For more information, see [Create a custom wrapper](#).

View the list of custom node types

The Custom Node Types page displays all custom node types in the workspace. You can click Create in the upper-right corner to create a custom node type. For more information, see [Create a custom node type](#).

Currently, you can only create custom node types in DataStudio.

The workspace owner or node type creator can change or delete existing node types.

- **Change:** You can click Change to edit the settings for the node type as needed.
- **Delete:** You can click this button to delete the node type that no node uses. If any node uses the node type, a message appears, indicating that you need to disable the node first before deleting the node type.

Use a custom node type

After creating a custom node type, go to the Data Analytics page.

Move the pointer over the Create icon and click Data Analytics. In the list that appears, select the created node type to create a node.



#### 2.4.5.15.2 Create a custom wrapper

The procedure of creating a wrapper includes four steps: specify settings for a wrapper, deploy the wrapper in the development environment, test the wrapper in the development environment, and deploy the wrapper in the production environment.

Specify settings for a wrapper

1. Click Wrappers in the left-side navigation pane. On the page that appears, click Create in the upper-right corner.

## 2. Specify the parameters in the Settings step.

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name              | The name of the wrapper. It must start with a letter and can only contain letters, digits, and underscores (_).                                                                                                                                                                                                                               |
| Owner             | The owner of the wrapper. You can select an owner from the workspace members. You are not allowed to edit wrappers owned by other members even if you are an administrator. Only the workspace owner can edit the wrappers of other members.                                                                                                  |
| Resource Type     | The type of the resource package for configuring the wrapper. Valid values: JAR and Archive. The size of the resource package can be up to 50 MB.                                                                                                                                                                                             |
| Resource File     | <p>The local resource file or OSS object for configuring the wrapper.</p> <div> <b>Note:</b><br/>The size of a local file can be up to 50 MB, and the size of a file that is stored in an OSS bucket can be up to 200 MB.</div>                              |
| Class Name        | The full path of the class for implementing the user wrapper.                                                                                                                                                                                                                                                                                 |
| Parameter Example | The parameters designed based on the JAR package you upload.                                                                                                                                                                                                                                                                                  |
| Version           | <p>The version of the configured wrapper. Select Create Version if you are creating a new wrapper. Select Overwrite Version if you are editing and rolling back a version.</p> <div> <b>Note:</b><br/>The version number is automatically generated.</div> |
| Description       | The description of the wrapper version.                                                                                                                                                                                                                                                                                                       |

## 3. Click Save and then click Next.



### Note:

The settings are updated to the database after you click Save.

- If you only modify basic settings of a wrapper without changing the resource file, the modification takes immediate effect after you click Save.
- If you change the resource file, the change only applies after deployment.

Deploy the wrapper in the development environment

**After you specify the parameters in the Settings step and click Next, the information in the Deploy in Development Environment step is updated accordingly. You can identify the changes by checking the file name and MD5 checksum.**

**Click Deploy in Development Environment. You can view the deployment progress in real time. After the wrapper is deployed, click Next.**

Test the wrapper in the development environment

**Specify the parameters for testing and click Test to send the parameters to the wrapper. This step is to validate the deployment and logic of the wrapper. You can also locally test the wrapper before uploading it for deployment.**

**After the test, review the output logs in the Test Results section on the right to determine whether the test is passed. If the test is passed, select Test Passed and click Next.**



**Note:**

**The Next button is operable only after you select Test Passed.**

Deploy the wrapper in the production environment

**Click Deploy in Production Environment. In the Confirm dialog box that appears, click OK. The wrapper is deployed in the production environment. You can view the deployment progress in real time.**



**Note:**

**The wrapper to be deployed in the production environment must be of the latest version, have been deployed in the development environment, and have passed the test. Otherwise, a message appears, indicating that the deployment in the production environment fails.**

**Click Complete. You can view and edit the created wrapper on the Wrappers page.**

### 2.4.5.15.3 Create a custom node type

**The Configure Custom Node Type page consists of three sections: Basic Information, Interaction, and Wrapper.**

- 1. On the DataStudio page, click Node Market in the top navigation bar. On the page that appears, click Custom Node Types in the left-side navigation pane.**

2. Click Create in the upper-right corner.
3. Specify the parameters in the Basic Information section.

| Parameter | Description                                                                                                                                                                                                                            |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name      | The name of the node type, which cannot be changed after being saved. Each node type has a unique name within the workspace. The name can be up to 20 characters in length, and can only contain letters, spaces, and underscores (_). |
| Icon      | The icon of the node type.                                                                                                                                                                                                             |
| Tabs      | The template of the node type. Currently, only Data Analytics is available.                                                                                                                                                            |
| Folder    | The folder where the node type belongs. You can select Data Integration or Data Analytics.                                                                                                                                             |

4. Specify the parameters in the Interaction section.

| Parameter     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shortcut Menu | <ul style="list-style-type: none"> <li>The options to appear in the shortcut menu. The following options are selected by default: Rename, Move, Clone, Steal Lock, View Versions , Locate in Operation Center, Delete, and Submit for Review.</li> <li>More options include Edit, Copy Resource Name , and Send to DataWorks Desktop (Shortcut).</li> </ul>                                                                                                                                                               |
| Tool Bar      | <ul style="list-style-type: none"> <li>The options to appear in the top navigation bar . The following options are selected by default : Save, Commit, Commit and Unlock, Steal Lock, Run, Show/Hide, Run with Arguments, Stop, Reload, Run Smoke Test in Development Environment, View Smoke Test Log in Development Environment, Run Smoke Test, View Smoke Test Log, Go to Operation Center of Development Environment, and Format.</li> <li>More options include Operation Center, Deploy, and Precompile.</li> </ul> |
| Editor Type   | The type of the editor. Currently, only Editor Only is available.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Parameter         | Description                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Right-Side Bar    | <ul style="list-style-type: none"><li>• The options to appear in the right-side bar. The following options are selected by default: Code Structure and Properties.</li><li>• More options include Version, Lineage, and Parameters.</li></ul>                                                                                 |
| Auto Parse Option | Specifies whether to display the Auto Parse option for this type of node. If you turn on this switch, the Auto Parse option is displayed on the Properties tab. Otherwise, it is not displayed. In an automatic parsing process, the system parses the input and output of a node based on the lineage specified in the code. |

5. Specify the parameters in the Wrapper section.

| Parameter                | Description                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Wrapper                  | The wrapper used for running the type of node. Select a wrapper that has been deployed.                                                                                 |
| Editor Language          | The language used for writing the code in the editor. Currently, only ODPS SQL is available.                                                                            |
| Use MaxCompute as Engine | Specifies whether to use MaxCompute as the compute engine. If your wrapper uses MaxCompute as the compute engine, select Yes. Otherwise, select No. Default value: Yes. |

6. Click Save and Exit. Then, go to the Data Analytics page to use the custom node type that is created.


## 2.4.6 Schedule

### 2.4.6.1 Basic attributes

On the Properties page of a node, you can set parameters of the node in the General, Schedule, Dependencies, and Parameters sections. The General section allows you to set basic attributes of the node.

In the left-side navigation pane, double-click a node to open it. Click Properties on the right side, and set parameters in the General section.



| Parameter   | Description                                                                                                                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node Name   | The name of the node that you specify when creating the node. To modify the name, right-click the node in the left-side navigation pane and select Rename.                                                                                                                                                                         |
| Node ID     | The unique ID of the node. The node ID is generated when the node is committed at the first time. The node ID cannot be modified.                                                                                                                                                                                                  |
| Node Type   | The type of the node that you specify when creating the node. The node type cannot be modified.                                                                                                                                                                                                                                    |
| Owner       | <p>The owner of the node. By default, the owner of a newly created node is the current logon user. You can change the owner.</p> <div> <b>Note:</b><br/>Only a member in the workspace where the node resides can be selected as the owner.</div> |
| Description | The description of the node about the business and usage .                                                                                                                                                                                                                                                                         |
| Arguments   | The parameter used to assign a value to a variable in the code during node scheduling. You can enter multiple parameters. Separate multiple parameters with spaces.                                                                                                                                                                |

Parameter value assignment formats for various node types

- **Format for ODPS SQL and ODPS MR nodes:** Variable name 1=Parameter 1  
Variable name 2=Parameter 2. Separate multiple parameters with spaces.
- **Format for shell nodes:** Parameter 1 Parameter 2. Separate multiple parameters with spaces.

For more information about built-in time parameters, see [Parameter configuration](#).

## 2.4.6.2 Parameter configuration

In common data R&D scenarios, the code of different types of nodes may be subject to change from time to time. You must dynamically modify the values of some parameters, such as date and time, based on the requirement changes and time changes.

The parameter configuration feature of DataWorks allows you to meet the requirements of each business scenario dynamically. After the parameters are set , the scheduled nodes with different recurrences can obtain required data through

automatic parsing. Parameters in DataWorks are classified into two categories: system parameters and custom parameters (recommended).

```
{
 "data": [
 {
 "beginRunningTime": "1564019679966",
 "beginWaitResTime": "1564019679966",
 "beginWaitTimeTime": "1564019679506",
 "bizdate": "1559318400000",
 "createTime": "1564019679464",
 "dagId": 332455685,
 "dagType": 5,
 "finishTime": "1564019679966",
 "instanceId": 2427622331,
 "modifyTime": "1564019679966",
 "nodeName": "vi", "status": 6
 }
],
 "errCode": "0",
 "errMsg": "",
 "requestId": "E17535-8C06-43F6-B1EA-6236FE9",
 "success": true
}
```

Auto-completion is supported when you specify a data type for a parameter.

Parameter types

| Parameter type                                                                           | Request method                                                                                                                                                                                                                                                                                    | Applicable to     | Example |
|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---------|
| <b>System parameters:</b><br>including bdp.system<br>.bizdate and bdp.<br>system.cyctime | To use the system<br>parameters in the<br>scheduling system,<br>directly reference \${<br>bdp.system.bizdate<br>} and \${bdp.system.<br>cyctime} in the code<br>without having<br>to set them in the<br>Arguments field. The<br>system automatically<br>replaces the values of<br>the parameters. | All node<br>types | None    |

| Parameter type                                         | Request method                                                                                                                                                                | Applicable to   | Example                                                                                                                                                                                                                                                         |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Non-system parameters: custom parameters (recommended) | Reference <code>\${key1}</code> and <code>\${key2}</code> in the code, and then set them in the Arguments field.<br>Example: <code>"key1=value1 key2=value2"</code> .         | Non-shell nodes | <ul style="list-style-type: none"> <li>• <b>Constant parameters:</b> <code>param1="abc"param2=1234.</code></li> <li>• <b>Variables:</b> <code>param1=\${yyyymmdd}</code>, which is calculated based on the value of <code>bdp.system.cyctime.</code></li> </ul> |
|                                                        | Reference <code>\$1</code> , <code>\$2</code> , and <code>\$3</code> in the code, and then set them in the Arguments field.<br>Example: <code>"value1 value2 value3"</code> . | Shell nodes     | <ul style="list-style-type: none"> <li>• <b>Constant parameters:</b> <code>"abc" 1234.</code></li> <li>• <b>Variables:</b> <code>\${yyyymmdd}</code>, which is calculated based on the value of <code>bdp.system.cyctime.</code></li> </ul>                     |

As described in the preceding table, the values of variables in custom parameters are calculated based on the values of system parameters. You can use custom variables to flexibly define the data to be obtained and the data format. For custom parameters, different parentheses are used as follows:

- Braces (`{}`): corresponding to the business time. For example, the variable `{yyyymmdd}` is calculated based on the value of `bdp.system.bizdate.`
- Brackets (`[]`): corresponding to the runtime. For example, the variable `[yyyymmddhh]` is calculated based on the value of `bdp.system.cyctime.`



#### Note:

Nodes can only be scheduled in the production environment. Therefore, the values of scheduling parameters are replaced only after scheduled nodes are run in the production environment.

You can use a smoke test to test whether the values of scheduling parameters can be replaced as expected during the scheduling. For more information, see [Smoke test in the development environment.](#)

Click Properties on the right side, and assign values to scheduling variables in the Arguments field in the General section. Note the following points when setting parameters:

- Do not add spaces on either side of an equation mark (=) in a parameter. For example, set `bizdate=$bizdate`.
- Separate multiple parameters (if any) with spaces. For example, set `bizdate=$bizdate datetime=${yyyymmdd}`.

### System parameters

DataWorks provides two system parameters, which are defined as follows:

- `${bdp.system.cyctime}`: the time to run an instance. Default format: `yyyymmddhh24miss`. It can be specified down to the hour and minute.
- `${bdp.system.bizdate}`: the timestamp of data to be analyzed. Default format: `yyyymmdd`. The default business date is one day before the scheduled runtime.

The formula for calculating the scheduled runtime and business date is as follows:

`Scheduled runtime = Business date + 1.`

To use the system parameters, directly reference `${bizdate}` in the code without having to set them in the Arguments field. The system automatically replaces the fields that reference the system parameters in the code.



#### Note:

The scheduling properties of a periodic node are configured to define the scheduling rules of the runtime. Therefore, you can calculate the business date based on the scheduled runtime of an instance and obtain the values of system parameters for the instance.

The scheduling parameter configuration of a PyODPS node is slightly different from that of a common node. For more information, see [PyODPS nodes](#).

### Example of system parameters

For example, to set an ODPS SQL node to be run once every hour from 00:00 to 23:59 every day, follow these steps if you want to use system parameters in the code:

#### 1. Reference system parameters in the code.

```
insert overwrite table tb1 partition(ds ='20150304') select
c1,c2,c3
```

```
from (
 select * from tb2
 where ds = '${bdp.system.cyctime}') t
full outer join(
 select * from tb3
 where ds = '${bdp.system.bizdate}') y
on t.c1 = y.c1;
```

2. After the preceding step, your node is partitioned by using the system parameters. Next, configure the scheduling properties and dependencies. For more information, see [Schedule](#) and [Dependencies](#). In this example, the node is scheduled by hour.
3. After setting the recurrence and dependencies, commit the node. You can check the node in [Operation Center](#). The node generates periodic instances during running from the second day. You can right-click an instance and select View Runtime Log to view the time when the system parameters are parsed.

For example, the scheduling system creates 24 running instances for the node on January 14, 2019. The business date is January 13, 2019 (the day before the execution date) for all instances. Therefore, `${bdp.system.bizdate}` is always displayed as 20190113. The runtime is the execution date plus the scheduled time. Therefore, `${bdp.system.cyctime}` is displayed as 20190114000000 plus the scheduled time of each instance.

Open the runtime logs of each instance and search for the replaced values of parameters in the code:

- The scheduled time for the first instance is 2019-01-14 00:00:00. Therefore, `bdp.system.bizdate` is replaced with 20190113, and `bdp.system.cyctime` is replaced with 20190114000000.
- The scheduled time for the second instance is 2019-01-14 01:00:00. Therefore, `bdp.system.bizdate` is replaced with 20190113, and `bdp.system.cyctime` is replaced with 20190114010000. The preceding figure shows the result after the replacement.
- Similarly, the scheduled time for the twenty-fourth instance is 2019-01-14 23:00:00. Therefore, `bdp.system.bizdate` is replaced with 20190113, and `bdp.system.cyctime` is replaced with 20190114230000.

Custom parameters for non-shell nodes

To set scheduling parameters for a non-shell node, add `${variable name}` in the code to reference the function, and then assign a value to the scheduling parameter.

**Note:**

The name of a variable in the SQL code can contain only lowercase letters (a-z), uppercase letters (A-Z), digits, and underscores (\_). If the variable name is date, the value of \$bizdate is automatically assigned to this variable. For more information, see the list of built-in scheduling parameters. You do not need to assign a value during scheduling parameter configuration. Even if another value is assigned, it is not used in the code because the value of \$bizdate is automatically assigned in the code by default.

**Example of custom parameters for non-shell nodes**

For example, to set an ODPS SQL node to be run once every hour from 00:00 to 23:59 every day, follow these steps if you want to use the hour-related custom variables **thishour** and **lasthour** in the code:

**1. Reference the parameters in the code.**

```
insert overwrite table tb1 partition(ds ='20150304') select
 c1,c2,c3
from (
 select * from tb2
 where ds ='${thishour}') t
full outer join(
 select * from tb3
 where ds = '${lasthour}') y
on t.c1 = y.c1;
```

**2. Click Properties on the right side, and assign values to the variables referenced in the code in the Arguments field in the General section.**

Configure the custom parameters as follows:

- **thishour**=\$[yyyy-mm-dd/hh24:mi:ss]
- **lasthour**=\$[yyyy-mm-dd/hh24:mi:ss-1/24]

**Note:**

The value of yyyy-mm-dd/hh24:mi:ss corresponds to that of cycetime. For more information, see the following Custom variables section.

You can enter **thishour**=\$[yyyy-mm-dd/hh24:mi:ss] **lasthour**=\$[yyyy-mm-dd/hh24:mi:ss-1/24] in the Arguments field.

**3. Set the node to be run once every hour.**

4. After setting the recurrence and dependencies, commit the node. You can check the node in [Operation Center](#). The node generates periodic instances during running from the second day. You can right-click an instance and select View Runtime Log to view the time when the custom parameters are parsed. The value of `cyctime` is 20190114010000. Therefore, the value of `thishour` is 2019-01-14/01:00:00, and the value of `lasthour`, which represents the last hour, is 2019-01-14/00:00:00.

#### Custom parameters for shell nodes

The parameter configuration procedure of a shell node is similar to that of a non-shell node, but the variable naming rules are different. Variable names for a shell node cannot be customized and must be named in the `$1,$2,$3...` format. For example, add the variable `$1` in the code of a shell node and configure the node parameter as `$xxx`, which is a built-in parameter. The value of `$xxx` replaces `$1` in the code.



#### Note:

For a shell node, when the number of parameters reaches 10, use `${10}` to declare the variable.

#### Example of custom parameters for shell nodes

Set a shell node to be run once at 01:00 each day. To use the custom constant parameter `myname` and the custom variable `ct` in the code, follow these steps:

1. Reference the parameters in the code.

```
echo "hello $1, two days ago is $2, the system param is ${bdp.system.cyctime}";
```

2. Click Properties on the right side, and assign values to the variables referenced in the code in the Arguments field in the General section. Value assignment rule: parameter 1 parameter 2 parameter 3... (The variables are parsed based on the parameter sequence. For example, `$1` is replaced with the value of parameter 1.) In this example, set `$1` and `$2` to `abcd` and `[$yyyy-mm-dd-2]`, respectively.
3. Set the node to be run once at 01:00 every day.
4. After setting the recurrence and dependencies, commit the node. You can check the node in Operation Center. The node generates periodic instances during running from the second day. Right-click an instance and select View Runtime Log. The logs show that `$1` in the code is replaced with constant

abcd, \$2 is replaced with 2019-01-12 (two days before the running date), and `${bdp.system.cyctime}` is replaced with 20190114010000.

#### Custom variables

A custom variable can be a constant parameter or a built-in scheduling parameter.

- Variable value being a constant value

For example, for an SQL node, add `${variable name}` in the code, and then configure the following parameter for the node: variable name=fixed value.

- **Code:** `select xxxxxx type='${type}'`
- **Value assigned to the scheduling variable:** type='aaa'. When the node is scheduled, the variable in the code is replaced with type='aaa'.

- Variable value being a variable

Variables are built-in scheduling parameters whose values depend on the system parameters `${bdp.system.bizdate}` and `${bdp.system.cyctime}`.

For example, for an SQL node, add `${variable name}` in the code, and then configure the following parameter for the node: variable name=scheduling parameter.

- **Code:** `select xxxxxx dt=${datetime}`
- **Value assigned to the scheduling variable:** datetime=\$bizdate

When the node is scheduled, if the current date is July 22, 2017, the variable in the code is replaced as follows: dt=20170721.

#### Variables

- \$bizdate
  - **Parameter description:** the business date in the format of yyyyymmdd. By default, this parameter is set to the day before the scheduled runtime.
  - **For example,** the code of an ODPS SQL node includes pt=\${datetime}, and the parameter configured for the node is datetime=\$bizdate. If the node is run on July 22, 2017, \$bizdate is replaced with pt=20170721.
- \$cyctime
  - **Parameter description:** the time at which the node is scheduled to run. If no scheduled time is configured for a daily-based node, cyctime is set to 00:00 of



the current day. The time is accurate to the second. This parameter is usually used for nodes scheduled by hour or minute.



**Note:**

:

- Pay attention to the difference between the time parameters configured by using `$[]` and `${}`. `$bizdate` indicates the business date, which is one day before the current day by default.
- `$cycetime` indicates the time at which the task is scheduled to run. If no scheduled time is configured for a daily task, `cycetime` is set to 00:00 of the current day. The time is accurate to the second. This parameter is usually used for tasks scheduled by hour or minute.

For example, if the node is scheduled to run at 00:30 on the current day, `$cycetime` is `yyyy-mm-dd 00:30:00`.

- If the time parameter is configured by using `${}`, `bizdate` is used as the benchmark for running nodes. For example, the time parameter is replaced by the business date selected for retroactive execution.
- If the time parameter is configured by using `$[]`, `cycetime` is used as the benchmark for running nodes, which is calculated in the same way as

the time in Oracle. The time parameter is replaced with the business date selected for retroactive execution plus one day.

For example, if the business date is set to 20140510 for retroactive execution, cyctime is replaced with 20140511.

- Examples (assume that \$cyctime=20140515103000):

■ `$(yyyy) = 2014`, `$(yy) = 14`, `$(mm) = 05`, `$(dd) = 15`, `$(yyyy-mm-dd) = 2014-05-15`, `$(hh24:mi:ss) = 10:30:00`, `$(yyyy-mm-dd hh24:mi:ss) = 2014-05-1510:30:00`

■ `$(hh24:mi:ss - 1/24) = 09:30:00`

■ `$(yyyy-mm-dd hh24:mi:ss -1/24/60) = 2014-05-1510:29:00`

■ `$(yyyy-mm-dd hh24:mi:ss -1/24) = 2014-05-15 09:30:00`

■ `$(add_months(yyyymmdd,-1)) = 20140415`

■ `$(add_months(yyyymmdd,-12*1)) = 20130515`

■ `$(hh24) =10`

■ `$(mi) =30`

- Method for testing the \$cyctime parameter:

After the instance starts to run, right-click the instance and select More.

Check whether the scheduled time is the time at which the instance is run periodically.

- \$jobid

- Parameter description: the ID of the workflow to which the node belongs.
- Example: `jobid=$jobid`.

- \$nodeid

- Parameter description: the node ID.
- Example: `nodeid=$nodeid`.

- \$taskid

- Parameter description: the instance ID of the node.
- Example: `taskid=$taskid`.

- \$bizmonth

- Parameter description: If the month of a business date (in the format of `yyyymm`) is the current month, the value of \$bizmonth is the month of the

business date minus 1. Otherwise, the value of \$bizmonth is the month of the business date.

- For example, the code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=$bizmonth`.

If the node is run on July 22, 2017, \$bizmonth is replaced as follows: `pt=201706`

.

- `${...}` custom parameters

- You can customize a time format based on the value of \$bizdate, where yyyy indicates the four-digit year, yy indicates the two-digit year, mm indicates the month, and dd indicates the day. You can use any combination of these parameters, for example, `${yyyy}`, `${yyyymm}`, `${yyyymmdd}`, and `${yyyy-mm-dd}`.
- \$bizdate is accurate to the day. Therefore, `${...}` can only represent the year, month, or day.
- The following table describes how to represent other intervals based on the system parameter \$bizdate.

| Interval        | Expression                    |
|-----------------|-------------------------------|
| N years later   | <code>\${yyyy+N}</code>       |
| N years before  | <code>\${yyyy-N}</code>       |
| N months later  | <code>\${yyyymm+N}</code>     |
| N months before | <code>\${yyyymm-N}</code>     |
| N weeks later   | <code>\${yyyymmdd+7*N}</code> |
| N weeks before  | <code>\${yyyymmdd-7*N}</code> |
| N days later    | <code>\${yyyymmdd+N}</code>   |
| N days before   | <code>\${yyyymmdd-N}</code>   |

- \$gmtdate

- **Parameter description:** the current date in the format of `yyyymmdd`. The value of this parameter is the current date by default. During retroactive execution, the input value is the business date plus one day.
- For example, the code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=$gmtdate`. If the node is run on July 22, 2017, \$gmtdate is replaced as follows: `pt=20170722`.

- `${yyyymmdd}`
- **Parameter description:** the business date in the format of `yyyymmdd`. The value of this parameter is the same as that of `$bizdate`, and it supports using multiple delimiters, for example, `yyyy-mm-dd`.

By default, this parameter is set to the day before the scheduled runtime. You can customize a time format for this parameter, for example, `yyyy-mm-dd` for `${yyyy-mm-dd}`.

- **Example:**
  - The code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=${yyyy-mm-dd}`. If the node is run on July 22, 2018, `${yyyy-mm-dd}` is replaced as follows: `pt=2018-07-21`.
  - The code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=${yyyymmdd-2}`. If the node is run on July 22, 2018, `${yyyymmdd-2}` is replaced as follows: `pt=20180719`.
  - The code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=${yyyymm-2}`. If the node is run on July 22, 2018, `${yyyymmdd-2}` is replaced as follows: `pt=201805`.
  - The code of an ODPS SQL node includes `pt=${datetime}`, and the parameter configured for the node is `datetime=${yyyy-2}`. If the node is run on July 22, 2018, `${yyyy-2}` is replaced as follows: `pt=2016`.
  - You can assign values to multiple parameters during ODPS SQL node configuration, for example, `startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}`.

## FAQ

- **Q:** The table partition format is `pt=yyyy-mm-dd hh24:mi:ss`, but spaces are not allowed in scheduling parameters. How can I configure the format of `[yyyy-mm-dd hh24:mi:ss]`?  
  
**A:** Use the user-defined parameters `datetime=[yyyy-mm-dd]` and `hour=[hh24:mi:ss]` to retrieve the date and time. Then, join them together to form `pt="${datetime} ${hour}"` in the code. The two parameters are separated with a space.
- **Q:** The table partition is `pt="${datetime} ${hour}"` in the code. To obtain the data for the last hour when the node is run, the custom variables `datetime=[yyyymmdd]` and `hour=[hh24-1/24]` can be used to obtain the date and time,

respectively. However, for an instance running at 00:00, the calculation result is 23:00 of the current day, instead of 23:00 of the previous day. What measures can I take in this case?

**A:** Modify the formula of datetime to `$(yyyymmdd-1/24)` and keep the formula of hour unchanged at `$(hh24-1/24)`. The calculation result is as follows:

- For an instance that is scheduled to run at 2015-10-27 00:00:00, the values of `$(yyyymmdd-1/24)` and `$(hh24-1/24)` are 20151026 and 23, respectively. This is because the scheduled time minus 1 hour is a time value that belongs to yesterday.
- For an instance that is scheduled to run at 2015-10-27 01:00:00, the values of `$(yyyymmdd-1/24)` and `$(hh24-1/24)` are 20151027 and 00, respectively. This is because the scheduled time minus 1 hour is a time value that belongs to the current day.

DataWorks offers four node execution modes.

- **Running in DataStudio:** You must assign a temporary value on the parameter configuration page to guarantee proper running. The configurations are not saved as node properties and do not take effect in the other three execution modes.
- **Automatic running at an interval:** No configuration is needed in the Arguments field. The scheduling system automatically replaces the parameters with the scheduled runtime of the current instance.
- **Run a test task or perform retroactive executions:** A business date must be specified for the execution. The scheduled runtime can be derived according to the formula described earlier in this section. Specify values for the business date and runtime parameters.

### 2.4.6.3 Schedule

This topic describes how to set the schedule properties of nodes, including the recurrence and whether to depend on last-cycle instances.

Click **Properties** on the right side and find the **Schedule** section.

Node statuses

- **Normal:** If you select this option, a node is scheduled and executed based on the recurrence. By default, this option is selected for a node.

- **Dry Run:** If you select this option, a node is scheduled based on the recurrence. However, once this node is scheduled, the system does not actually run the node but directly returns a success response.
- **Retry Upon Error:** If you select this check box, a node is rerun when it encounters an error. By default, a node can be automatically rerun for a maximum of three times at an interval of 2 minutes.
- **Skip Execution:** If you select this check box, a node is scheduled based on the recurrence. However, once this node is scheduled, the system does not actually run the node but directly returns a failure response. You can select this check box if you want to suspend a node and run it later.

#### Instance recurrence

After a node is committed, the scheduling system creates an instance every day from the next day based on the schedule properties of the node. Then the scheduling system runs the instance based on the execution results and schedule of its ancestor instances. If a node is committed after 23:30, the scheduling system creates instances for it two days later.



#### Note:

If you schedule a node to run on every Monday, the node is run only on Mondays. On the other days, once this node is scheduled, the system does not actually run the node but directly returns a success response. Therefore, you need to set the business date to one day earlier than the runtime for weekly scheduled nodes during testing or retroactive execution.

For a node that runs periodically, the priority of its dependencies is higher than that of its schedule properties. That is, when the scheduled time is reached, the node instance is not run immediately but first checks whether all the ancestor node instances have been run.

- The node instance is in the Not Running state if any ancestor instances have not been run when the scheduled runtime is reached.
- The node instance is in the Pending (Schedule) state if the scheduled runtime is not reached although all its ancestor instances have been run.
- The node instance is in the Pending (Resources) state if all its ancestor instances have been run and the scheduled runtime is reached.

## Cross-cycle dependencies

DataWorks supports the following three types of cross-cycle dependencies:

- **Dependency on instances of child nodes**
  - **Node dependency:** The current node depends on the last-cycle instances of its child nodes. For example, node A has three child nodes B, C, and D. If you select this node dependency, node A depends on the last-cycle instances of nodes B, C, and D.
  - **Business scenario:** The current node depends on the last-cycle instances of its child nodes to cleanse the output tables of the current node and check whether the final result is generated properly.
- **Dependency on instances of the current node**
  - **Node dependency:** The current node depends on its last-cycle instances.
  - **Business scenario:** The current node depends on the data output result of its last-cycle instances.
- **Dependency on instances of custom nodes:** You need to manually enter the IDs of the nodes on which the current node depends. You can specify multiple nodes and separate their IDs with a comma (,). Example: 12345,23456.
  - **Node dependency:** The current node depends on the last-cycle instances of custom nodes.
  - **Business scenario:** In the business logic, the current node depends on the proper output of other business data that is not processed by the current node
- 



### Note:

The difference between cross-cycle dependencies and dependencies in the current cycle lies in that cross-cycle dependencies are displayed as dotted lines in Operation Center.

Before bringing a node offline, you must delete all dependencies of the node so that other nodes can run properly.

## Schedule by day

Nodes scheduled by day are run automatically once every day. When you create a periodic node, the node is set to run at 00:00 every day by default. You can specify another runtime as needed.

- If you select the **Customize Runtime** check box, the node is run at the specified time every day. The time format is YYYY-MM-DD HH:MM:SS.

**Note:**

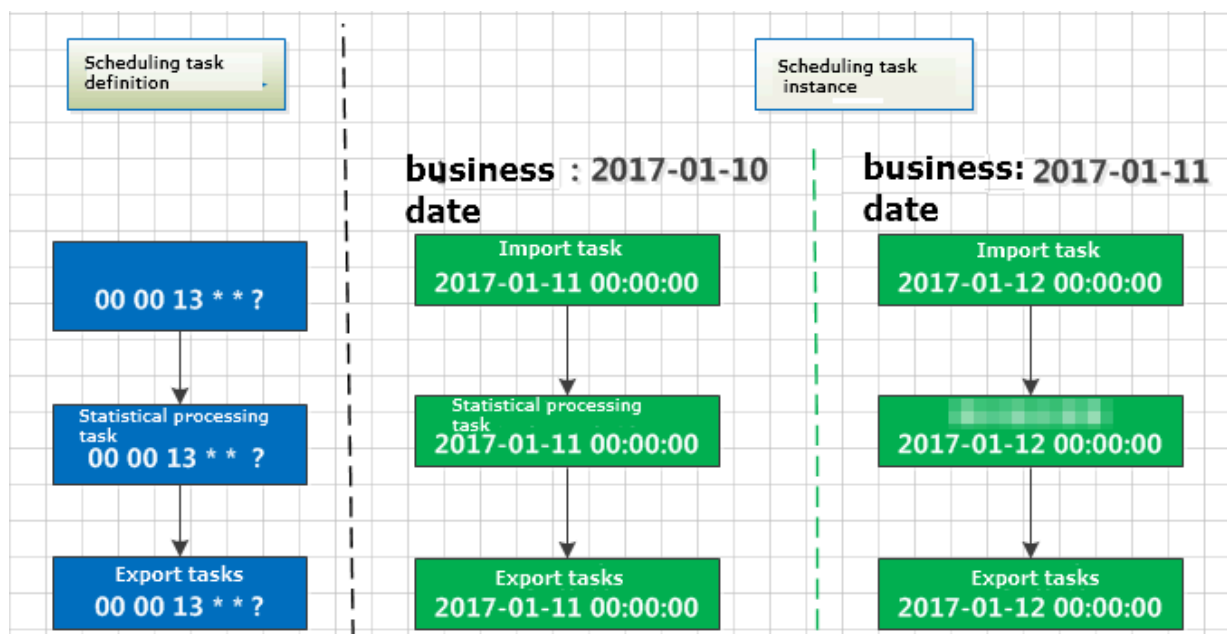
A node can be scheduled to run only when all its ancestor nodes have been run and the scheduled runtime is reached. Both prerequisites are indispensable and have no specific chronological order.

- If you do not select the **Customize Runtime** check box, the scheduled runtime of the daily node is randomly generated from 00:00 to 00:30.

**Scenarios:**

For example, you have created an import node, an analytics node, and an export node. They are all scheduled to run at 13:00 every day. The analytics node depends on the import node, and the export node depends on the analytics node.

Based on the preceding node schedules and dependencies, the scheduling system automatically generates instances for the nodes and runs them as follows.





## Schedule by week

Nodes scheduled by week are run at a specified time of specified days every week. On the other days, the system still generates instances to ensure the proper running of descendant instances. However, once a node instance is scheduled, the system does not actually run any logic or consume any resources but directly returns a success response.

Schedule ?

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Week

Plan Time : ☒

Specified Time : Monday x Friday x

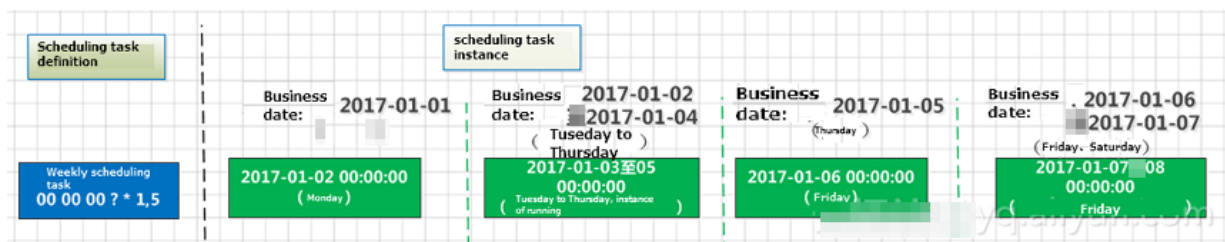
Planned Time : 13:00

CRON Expression : 00 00 13 ? \* 1,5

Depend on Last Interval : ☐

The configurations in the preceding figure mean that the system schedules instances created on Mondays and Fridays, but returns success responses without scheduling instances created on Tuesdays, Wednesdays, Thursdays, Saturdays, and Sundays.

Based on the preceding node schedules and dependencies, the scheduling system automatically generates instances for the nodes and runs them as follows.



## Schedule by month

Nodes scheduled by month are run at a specified time of specified days every month. On the other days, the system still generates instances to ensure the proper running of descendant instances. However, once a node instance is scheduled, the system does not actually run any logic or consume any resources but directly returns a success response.

**Schedule** ⓘ

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ⓘ

Validity Period : 1970-01-01 - 9999-01-01 ⓘ

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Month

Plan Time : ☒

Specified Time : Day 1 ×

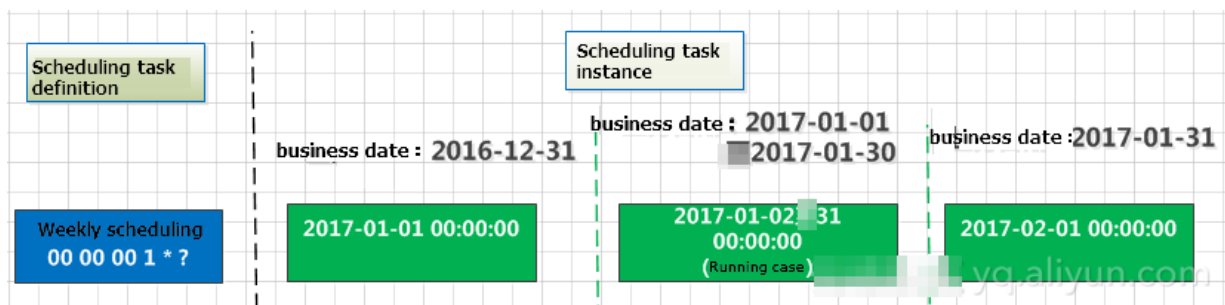
Planned Time : 00:00 ⓘ

CRON Expression : 00 00 00 1 \* ?

Depend on Last Interval : ☐

The configurations in the preceding figure mean that the system schedules instances created on the first day of each month, but returns success responses without scheduling instances created on the other days.

Based on the preceding node schedules and dependencies, the scheduling system automatically generates instances for the nodes and runs them as follows.



## Schedule by hour

**Nodes scheduled by hour are run once every N hours in a specific time period every day, for example, once every hour from 01:00 to 04:00 every day.**



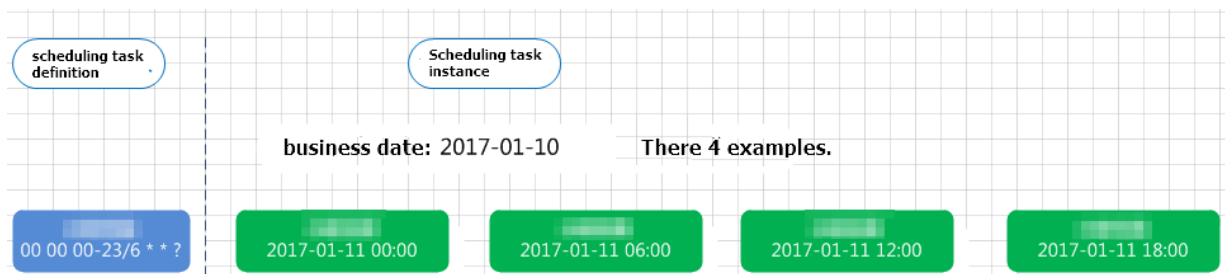
### Note:

The time period is a closed interval. For example, if a node is scheduled to run once every hour in the period from 00:00 to 03:00, the scheduling system creates four instances every day, which are run at 00:00, 01:00, 02:00, and 03:00, respectively.

The screenshot shows the 'Schedule by hour' configuration interface. It includes the following fields and options:

- Error Rate this product:** ☐ ?
- Validity Period:** 1970-01-01 - 9999-01-01 (with a calendar icon)
- Note:** The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.
- Pause Scheduling:** ☐
- Schedule Interval:** Hour (dropdown menu)
- Plan Time:** ☒
- Start Time:** 00:00 (with a clock icon) **Interval:** 1 h **End Time:** 23:59 (with a clock icon)
- Specified Time:** 0:00 (with a close icon and a dropdown arrow)
- CRON Expression:** 00 00 00-23/6 \*\* ?
- Depend on Last Interval:** ☐

The configurations in the preceding figure mean that the node is run automatically every six hours in the period from 00:00 to 23:59 every day. Therefore, the scheduling system automatically generates instances for the node and runs them as follows.



## Schedule by minute

**Nodes scheduled by minute are run once every N minutes in a specific time period every day.**

The configurations in the following figure mean that the node is run every 30 minutes in the period from 00:00 to 23:00 every day.

**Schedule** ?

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Minute

Plan Time : ☒

Start Time : 00:00

Interval : 30 min

End Time : 23:00

CRON Expression : 00 \*/30 00-23 \*\* ?

Currently, a minimum interval of 5 minutes is supported. The time expression is automatically generated based on the time you select and cannot be modified.

**Schedule** ?

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Minute

Plan Time : ☒

Start Time : 00:00

Interval : 30 min

End Time : 23:59

CRON Expression : 00 \*/30 00-23 \*\* ?

## FAQ

- **Q: Node A is scheduled by hour, and its descendant node B is scheduled by day. Is it feasible that node B is automatically run every day after all instances of node A are executed?**

**A:** A node can depend on any other node, and there are no limits on the scheduling type of the node. Therefore, a node scheduled by day can depend on a node scheduled by hour. To enable node B to be automatically run every day after all 24 instances of node A are run, do not specify the daily runtime for node B. Then, configure node A as an ancestor of node B. For more information, see the Dependencies topic. Therefore, nodes of different scheduling types can depend on each other. The recurrence of each node is specified in its schedule property settings.

- **Q: If node A is run once on the hour every hour every day and node B is run once every day, how do I enable node B to be run after node A is run for the first time every day?**

**A:** When configuring node A, select the Cross-Cycle Dependencies check box and select Instances of Current Node from the Depend On drop-down list. Set the scheduled time of node B to 00:00 everyday. In this way, instances of node B only depend on the instance of node A generated at 00:00 every day, that is, the first instance of node A.

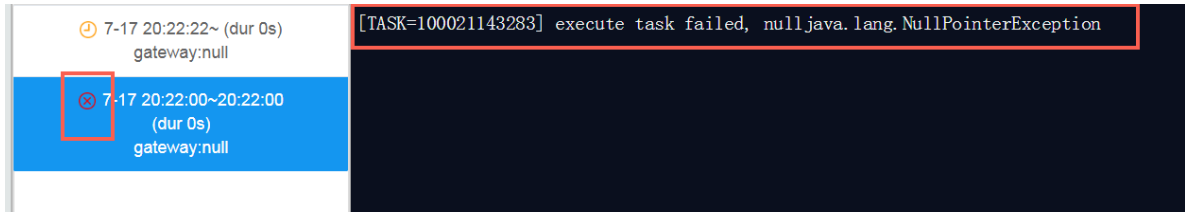
- **Q: Node A is run once every Monday and node B depends on node A. How do I enable node B to be run once every Monday?**

**A:** Set the schedule properties of node B to be the same as those of node A. That is, select Week as the instance recurrence and select Monday.

- **Q: How are the instances of a node affected when the node is deleted?**

**A:** When a node is deleted, its instances are remained because the scheduling system still generates one or more instances for the node based on the schedule properties. However, when the instances are initiated after the node is deleted,

an error message appears because the required code is unavailable, as shown in the following figure.



- **Q: Can I enable a node to process monthly data on the last day of each month?**

**A: No.** Currently, the system does not support setting the execution date to the last day of each month. If you enable a node to be run on the thirty-first day of each month, the scheduling system runs a node instance in each month that has 31 days and returns a success response without running the node instance in any other month.

We recommend that you configure a node to process the data of the past month on the first day of each month.

- **Q: If a node scheduled by day depends on a node scheduled by hour, how do I enable the node scheduled by day to be run at 00:00 every day?**

**A: The node scheduled by day does not need to depend on the data generated on the current day for the node scheduled by hour. Instead, the node scheduled by day only needs to depend on the data generated every hour on the day before for the node scheduled by hour. Otherwise, the instances of the node scheduled by day can be run only on the next day.**

In the Schedule section, select the Cross-Cycle Dependencies check box and select Instances of Custom Nodes from the Depend On drop-down list. Then, enter the ID of the ancestor node on which the node scheduled by day depends, and commit the node.

- **Q: What can I do if I do not know when the output data of the ancestor node is generated?**

**A: You can set the cross-cycle dependency for the current node to depend on the last-cycle instances of the ancestor node.**

- **Q: After a modified node is committed to the production environment, will the node instances that were originally faulty in the production environment be overwritten?**

**A: No. The updated code is used to run the node instances that have not been run. The node instances that have been generated will not be overwritten. If the scheduling configuration is modified, you need to generate and run a new instance.**

#### 2.4.6.4 Dependencies

Scheduling dependencies are the foundation for building orderly workflows. You need to configure correct dependencies between nodes to ensure that business data is produced effectively and in time. This helps standardize data R&D activities.

DataWorks allows you to automatically parse node dependencies from the code or manually customize node dependencies. You can guarantee the orderly production of business data by configuring correct relationships between ancestor and descendant nodes and monitoring the running status of nodes.

The purpose of configuring node dependencies is to check the data output time of the table queried by SQL and check whether data is properly produced from an ancestor node based on the node status.

You can set the output of an ancestor node as the input of a descendant node to configure a dependency between the two nodes.

Regardless of the dependency configuration mode, the overall scheduling logic is that descendant nodes can be scheduled only after ancestor nodes are run.

Therefore, each workflow node must have at least one parent node. The dependencies between the parent nodes and child nodes are the core of scheduling dependencies. The following sections describe the principle and configuration methods of scheduling dependencies in detail.

Differences between automatic parsing and custom dependencies

**In an automatic parsing process, the system parses the input and output of a node based on the lineage specified in the code.**

**Add a custom dependency if the lineage parsed from the code is inaccurate.**

**Ensure that the lineage in the code is the most accurate possible to reduce custom**

dependencies. The following section describes how to configure the node input and output.

**Auto Parse:** If you select Yes, node dependencies are automatically parsed from the code.

For example, the code of an ODPS SQL node is as follows:

```
insert overwrite table table_a as select * from project_b_name.table_b
;
```

From the code, the system determines that the current node depends on the node that generates table\_b and then the current node generates table\_a. Therefore, the output name of the parent node is project\_b\_name.table\_b and the output name of the current node is project\_name.table\_a.

- If you do not need to parse node dependencies from the code, select No.
- The code may contain many temporary tables with names starting with t\_. Temporary tables are not involved in the parsing of a scheduling dependency. You can specify the name prefix for temporary tables in application properties.
- If a table in the code is an output table that is being referenced (being depended on), it is parsed only as an output table.
- If a table in the code is referenced or exported for multiple times, only one scheduling dependency is parsed.



**Note:**

By default, a table with the name starting with t is recognized as a temporary table. Temporary tables are excluded from the automatic parsing process. If names starting with t do not indicate temporary tables, contact your application owner to specify the name prefix for temporary tables in application properties.

## Parent nodes

You must enter the output name of an ancestor node, rather than the ancestor node name. An ancestor node indicates the parent node on which the current node depends. Note that a node may contain multiple output names. Enter an output name as needed. You can search for an output name of the ancestor node to be added, or run the SQL statement for lineage analysis to parse the output name.



**Note:**



**You must use the output name or output table name of the depended ancestor node to search for it.**

**If you search for an output name of the ancestor node, the crawler searches for the output name among the output names of nodes that have been committed to the scheduling system.**

- **Search by entering the output name of the parent node**

**You can enter the output name of a node to search for it and configure the node as the ancestor node of the current node to create a dependency.**

- **Search by entering the output table name of the parent node**

**When using this method, make sure that one of the output names of the parent node is the table name following INSERT or CREATE in the SQL statement of the current node, such as `Project name.Table name` . Such output names can be automatically parsed.**

**Click Submit. The output name can be searched by other nodes through searching the table name.**

## Outputs

**This parameter specifies the output of the current node. You can view the current node output on the Properties page on the right.**

**The system assigns a default output name that ends with .out to each node. You can also customize the output name or obtain an output name through automatic parsing.**



### Note:

**The output name of each node must be globally unique.**

## FAQ

- **Q: After automatic parsing, the submission fails. The following error message appears on the page: The output workshop\_yanshi.tb\_2 of the parent node does**

not exist. Commit this node after committing the parent node. Why does this error occur?

**A:** The possible causes are as follows:

- The ancestor node is not committed. Commit the ancestor node and try again.
- The ancestor node is committed, but the output name of the ancestor node is not workshop\_yanshi.tb\_2.



**Note:**

Usually, the output names of the parent node and the current node are automatically parsed based on the table name following INSERT, CREATE, or FROM. Make sure that the configuration method is consistent with that described in the section Automatic dependency parsing.

- **Q:** In the output of the current node, the descendant node name and ID are empty and cannot be specified. Why does this happen?

**A:** If the current node does not have any descendant node, the descendant node name and ID are empty. After a descendant node is configured for the current node, the corresponding content is automatically parsed.

- **Q:** What is the output name of a node used for?

**A:** The output name of a node is used to establish dependencies with other nodes. If the output name of node A is ABC and node B takes ABC as its input, the dependency is established between nodes A and B.

- **Q:** Can a node have multiple output names?

**A:** Yes. If a descendant node references an output name of the current node as its parent node output name, a dependency is established between the descendant node and the current node.

- **Q:** Can multiple nodes have the same output name?

**A:** No. The output name of each node must be unique under your Alibaba Cloud account. If multiple nodes export data to the same MaxCompute table, we recommend that you use Table name\_Partition ID as the output name format of these nodes.

- **Q: How can I configure no parsing of intermediate tables during automatic dependency parsing?**

**A: Right-click the intermediate table name in the SQL code and select Delete Input or Delete Output, and then perform the automatic parsing of the input and output again.**

- **Q: How do I configure dependencies of the upmost node?**

**A: You can set it to depend on the root node of the current workspace.**

- **Q: Why do I find a non-existent output name of node B when searching for the ancestor node output name on node A?**

**A: This is because the search feature works based on the committed node information. After node B is committed, if you delete the output name of node B and does not commit node B to the scheduling system, the deleted output name of node B can still be found on node A.**

- **Q: How do I implement the node flow of A->B->C once an hour (run node B after node A is completed, and run node C after node B is completed)?**

**A: Set the output of node A as the input of node B and the output of node B as the input of node C, and set the recurrence of nodes A, B, and C to one hour.**

- **Q: An error is returned indicating that the parent node ID is not obtained through automatic parsing of the table. What does this error mean?**

**A: This error does not indicate that the table does not exist. Instead, it indicates that the table is not the output of a specific node. Therefore, the table cannot be used to locate the node that generates the table data. In this case, the dependency on the node cannot be created.**

**According to the principle of automatic parsing described above, a dependency is created after the output of an ancestor node is set as the input of a descendant node. If no ancestor node is found through the automatic parsing of the**

`xc_demo_partition` table in SQL statements, no node takes the `xc_demo_partition` table as its output.

You can resolve this problem in the following way:

1. Find the node that generates the table and view the node output.

If you do not know which is the target node, you can use the code search function to perform fuzzy search by keyword.

2. If the table is locally uploaded or you do not need to depend on the node, you can right-click the table name in the code area and select Delete Input.



**Note:**

Ensure that the lineage in the code is the most accurate possible to reduce custom dependencies.

## 2.4.6.5 Resource type

**Resource Group:** The group of server resources that is associated to the workspace. DataWorks provides a default resource group. You can add custom resource groups to a workspace if required.

## 2.4.7 Manage configurations

### 2.4.7.1 Configuration Center

The Configuration Center page allows you to configure the wizard of DataStudio. For example, you can change the code style or theme and add or remove tabs from the left-side navigation pane.

The following sections describes the five sections on the Configuration Center page.

- [Configuration Center](#)
- [Project Configuration](#)
- [Templates](#)
- [Theme Management](#)
- [Table Levels](#)

### 2.4.7.2 Configuration Center

The Configuration Center tab includes two sections: **Module Options** and **Editor Settings**.

## Module Options

You can select the services in the Module Options section. Selected services are displayed in the left-side navigation pane of DataStudio. You can select or deselect services by clicking corresponding tags, and sort them by dragging and dropping the tags between the Added Modules and Available Modules areas.

If you hover over a service tag in the Available Modules area, the tag turns into a blue Add button.

If you hover over a service tag in the Added Modules area, the tag turns into a red Remove button.



### Note:

Settings in the Module Options section take effect immediately in the current workspace. If you need to apply the settings to all workspaces, click **Apply to All Workspaces** in the Editor Settings section.

## Editor Settings

You can configure the code editor in the Editor Settings section. The settings take effect immediately in the current workspace without the need of updating the page.

- **Thumbnail View:** If you select this option, a minimap of your code is displayed on the right of the editor. The minimap is very useful for quick navigation and code understanding.
- **Error Check**

If you select this option, DataWorks marks potential syntax errors with a red squiggly line. When you see a syntax error, you can hover over the underlined code to view the error message.

- **Auto Save**

If you select this option, DataWorks automatically saves the code being edited at a certain interval. In this way, if the code editor of a node is closed unexpectedly, you can choose to apply the code saved on the server or the code saved in the local cache when you re-open the node.

- **Code Style**

You can select whether the code completed based on hints is uppercase or lowercase.

- **Code Font Size**

Valid values: 12 to 18. You can change the font size based on your habits and code size.

- **Code Hint**

When you are typing in the code editor, DataWorks provides a hint list. It automatically completes the term based on the top of the list if you press Enter.

- **Space:** Indicates whether to automatically add a space after each automatically completed term.
- **Keyword:** Indicates whether to enable keyword hints.
- **Syntax Template:** Indicates whether to enable syntax template hints.
- **Project:** Indicates whether to enable project name hints.
- **Table:** Indicates whether to enable table name hints.
- **Field:** Indicates whether to enable field name hints.

- **Theme**

You can choose from the black and white DataStudio themes.

- **Scope**

If you click Apply to All Workspaces, the settings on the Configuration Center tab will take effect in all existing workspaces.

### 2.4.7.3 Project Configuration

The Project Configuration page provides five parameters: Partition Date Format, Partition Field Name, Temporary Table Prefix, Upload Table (Import Table) Prefix, and Mask Data in Page Query Results.

Click the settings icon in the lower-left corner of the DataStudio page to open the Configuration Center page.

In the left-side navigation bar, click Project Configuration.

| Parameter              | Description                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Partition Date Format  | The default date format of partition field values. You can modify the format as required.                                     |
| Partition Field Name   | The default name of a partition field.                                                                                        |
| Temporary Table Prefix | The prefix of temporary table names. By default, tables with the prefix t_ in their names are identified as temporary tables. |

| Parameter                          | Description                                                                                                                  |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Upload Table (Import Table) Prefix | The prefix of the names of tables uploaded on the DataStudio page.                                                           |
| Mask Data in Page Query Results    | When the switch is turned on, the result returned for a temporary search task in the current workspace will be desensitized. |

Enable data desensitization for DataWorks workspaces

Data desensitization for DataWorks needs to be enabled in workspaces one by one. After data desensitization is enabled, the result returned for the temporary query task in the current workspace is desensitized. The underlying storage data is not affected because only dynamic desensitization is performed.



**Note:**

For example, data desensitization has been set for workspace A, but not workspace B. If you access a table in workspace A from workspace B, the plaintext result is displayed.

On the Project Configuration page, turn on the switch Mask Data in Page Query Results, and click Save to enable data desensitization for DataWorks workspaces.



**Note:**

By default, DataWorks does not allow you to download desensitized data or enable data desensitization.

Enter a query statement in the SQL node to check whether data desensitization is enabled for DataWorks workspaces.

## 2.4.7.4 Templates

The Templates page displays code template. Workspace administrators can change the templates.

Currently, templates are only available for ODPS SQL, ODPS MR, and shell nodes.

## 2.4.7.5 Theme Management

Each workspace can hold a great number of tables. For easy management, you can organize the tables in two levels of folders. Such folders are called topics.

Workspace administrators can create topics to categorize tables based on usage and names.

### 2.4.7.6 Table Levels

On the Table Levels page, you can manage table levels. If a table is incorrectly organized and cannot be located, the execution of tasks may fail.

The workspace owner and administrators can create table levels, and no default levels are available.

## 2.4.8 Deploy

### 2.4.8.1 Publish nodes

In a complete data analytics process, developers develop and debug code, configure dependencies, schedule tasks, and then submit the nodes to the production environment for execution.

Standard DataWorks workspaces can process data seamlessly from the development to production environments within a single workspace. We recommend that you use standard workspaces for data analytics, production, and publishing.

Publish a task in a standard workspace

Each standard DataWorks workspace are linked with two MaxCompute projects, one as the development environment and the other as the production environment . You can directly submit and publish nodes from the development environment to the production environment.

#### Procedure

1. Click Submit after the code is debugged and the node is scheduled. DataWorks automatically checks the dependencies between nodes.
2. When the submission is complete, click Publish.
3. Navigate to the node publishing page, find required nodes and click Add to Nodes to Publish in the Actions column. Then, the nodes appear on the list of Nodes to Publish.

You can search nodes by submitter, node type, change type, submission time, node name, and node ID. If you click Publish Selected Nodes, the nodes are published to the production environment for execution.



4. Choose **View Nodes to Publish > Publish All** to publish all nodes on the list to the production environment.



**Note:**

Standard workspaces protects tables in the production environment from being manipulated, and therefore the production environment is stable, secure, and reliable. We recommend that you use standard workspaces for data analytics, production, and publishing.

Clone nodes between basic workspaces

You cannot publish nodes in basic workspaces. If you need to separate development and production environments in basic workspaces, create two basic workspaces, one for development and the other for production. You can clone nodes from the development workspace to the production workspace.

You can clone nodes from the development workspace to the production workspace, and then submits the cloned nodes to the scheduler for execution.



**Note:**

- **Permissions:** Only workspace administrators and deployment experts can clone nodes.
- **Workspace type:** Only nodes in basic workspaces can be cloned, and those in standard workspaces cannot.
- **Prerequisites:** source workspace (basic workspace A) and destination workspace B (standard workspace B).

1. **Submit a node.**

After you edit a node, submit the node.

2. **Click Cross-project Cloning.**
3. **Select a destination workspace, select the submitted node, and click Add to Nodes to Clone.**
4. **Clone the node. Click View Nodes to Clone. Ensure the nodes to be clone, and click Clone All.**

Click Clone to complete the clone process.

## 5. View the cloned nodes.

You can view the successful clone tasks on the Release Package page.

Switch to the destination workspace. You can find the node cloned from the source workspace.

### 2.4.8.2 Clone nodes across workspaces

If you clone nodes across workspaces, DataWorks automatically modifies output names in the destination workspace to distinguish nodes in different workspaces of the same tenant account. This allows to successfully clone node dependencies.

#### Clone business flows

Assume that the output of the task\_A node in the Project\_1 workspace is Project\_1.task\_1\_out. If you clone a business flow of the task\_A node to the destination workspace Project\_2, then the node output name turns into Project\_2.task\_out in the destination workspace.

#### Clone single nodes

If you clone only one node task\_B to the destination workspace Project\_2, DataWorks automatically sets the root node of the destination workspace as the parent node and the output of the root node as the input for the task\_B node. This allows the task\_B node to automatically depend on the root node in the destination workspace.

#### Node dependencies

Assume that the task\_B node in the Project\_1 workspace is dependent on the task\_A node in the Project\_3 workspace. If you clone task\_B in Project\_1 to the destination workspace Project\_2, the dependency between task\_A and task\_B is also cloned. The task\_B node in Project\_2 also depends on the task\_A node in Project\_3.

## 2.4.9 Ad-hoc business flows

### 2.4.9.1 Description

In an ad-hoc business flow, all nodes can only be manually triggered, and cannot be automatically triggered by DataWorks. You do not need to configure the node dependencies or output for nodes in an ad-hoc business flow.

The following table describes icons and tabs on an ad-hoc business flow tab.

| No. | Icon or Tab             | Description                                                                                                                                                                            |
|-----|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Submit                  | Submit all nodes in the current ad-hoc business flow.                                                                                                                                  |
| 2   | Run                     | Run all nodes in the current ad-hoc business flow. Nodes in ad-hoc business flows do not have dependencies, and therefore they are initiated at the same time.                         |
| 3   | Stop                    | Stop all running nodes in the current ad-hoc business flow.                                                                                                                            |
| 4   | Publish                 | Redirect to the node publishing page. You can publish some or all nodes to the production environment.                                                                                 |
| 5   | Administration          | Redirect to the administration page.                                                                                                                                                   |
| 6   | Refresh                 | Refresh the current ad-hoc business flow tab.                                                                                                                                          |
| 7   | Auto Layout             | Sort the nodes in the current ad-hoc business flow.                                                                                                                                    |
| 8   | Zoom In                 | Zoom in the kanban of the current ad-hoc business flow.                                                                                                                                |
| 9   | Zoom Out                | Zoom out the kanban of the current ad-hoc business flow.                                                                                                                               |
| 10  | Search                  | Search for a node in the current ad-hoc business flow.                                                                                                                                 |
| 11  | Toggle Full Screen Mode | Toggle the full screen mode.                                                                                                                                                           |
| 12  | Parameters              | Configure flow parameters. Flow parameters have a higher priority than node parameters . This means that a flow parameter takes precedence if it has the same key as a node parameter. |
| 13  | Operation Records       | View the operation records of all nodes in the current ad-hoc business flow.                                                                                                           |
| 14  | Version                 | View the published versions of all nodes in the current ad-hoc business flow.                                                                                                          |

## 2.4.9.2 Functions

### Register functions

You can define MaxCompute functions in the DataWorks console. This feature is similar to the `add function` command of MaxCompute.

Currently, DataWorks supports Python and Java functions. To use a user-created function, *upload the function code as a resource* and then register the function.

### Procedure

1. Right-click the Ad-Hoc Business Flows folder on the Ad-Hoc Business Flows tab, and select Create Business Flow.
2. In a Java environment, complete the code for the function, and then package and publish the code.

Alternatively, in the DataWorks console, create a Python resource, edit the code in the resource file, and then save, submit, and publish the resource. For more information, see *Create resources*.

3. Choose Function > Create Function. In the dialog box that appears, specify the function name and click Submit.
4. Configure the function.

| Parameter   | Description                                                                                                                                                               |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Class Name  | The main class name of a Python function in the format of <code>pythonResourceName.className</code> . Do not include the <code>.py</code> extension in the resource name. |
| Resources   | The resource list. Separate the resource names by a comma (,).                                                                                                            |
| Description | (Optional) The description of the function.                                                                                                                               |

5. Submit the function.

After the function configuration is complete, click the Save in the tool bar and submit the function to the development environment. After the function is submitted, it is unlocked.

6. Publish the function.

For more information, see *Deploy*.

### 2.4.9.3 Resources

You need to upload files as MaxCompute resources if you need to use them in user-defined MaxCompute functions or ODPS MR nodes.

- **MaxCompute SQL functions:** You can upload compiled JAR packages to MaxCompute as JAR resources. Then, if you run a function that uses a JAR resource, MaxCompute automatically downloads the corresponding package.
- **ODPS MR nodes:** You can upload compiled JAR packages to MaxCompute as JAR resources. Then, if you run an ODPS MR node that uses a JAR resource, MapReduce automatically downloads the corresponding package.

You can also upload text files, MaxCompute tables, and various compressed packages (such as .zip, .tgz, .tar.gz, .tar, and .jar) to MaxCompute so that you can use them in user-created functions and ODPS MR nodes.

Available MaxCompute resource types are listed as follows:

- **File**
- **Archive:** DataWorks automatically identifies the file format based on the extension. Supported formats: .zip, .tgz, .tar.gz, .tar, and .jar.
- **JAR:** compiled Java JAR packages.

In wizard mode, you can upload resources of the JAR, Python, or file type. The method of creating resources differs as follows:

- **JAR resource:** You need to compile Java code in a Java environment, compress the code into a JAR package, and then upload the package as a JAR resource to MaxCompute.
- **File resource:** For a file that is smaller than or equal to 500 KB, you can create a file resource and edit it in the DataWorks console.
- **If you need to upload a local file, select Larger than 500 KB while creating the resource.**

Create a resource

1. Right-click the Ad-Hoc Business Flows folder on the Ad-Hoc Business Flows tab, and select **Create Business Flow**.
2. In the new business flow folder, right-click the Resource folder, and choose **Create Resource > JAR**.

3. The Create Resource dialog box appears. Specify a resource name according to the naming convention, set the resource type to JAR, select a local JAR package, and click Submit.

**Note:**

- If the selected JAR package has been uploaded from the ODPS client, deselect Upload to ODPS. Otherwise, an error will occur during the upload process.
- The resource name can be different from the name of the uploaded file.
- Convention for naming resources: A resource name can contain letters (case insensitive), numbers, underscores (\_), and periods (.). It must be 1 to 128 characters in length. A JAR resource name must end with .jar.

4. Click Submit to submit the resource to the development environment.
5. Publish the resource.

For more information, see [Deploy](#).

## 2.4.9.4 Tables

Create a table

1. Log on to the DataWorks console.
2. Create a manually triggered workflow.
  - a. On the DataStudio page, click Manually Triggered Workflows in the left-side navigation bar.
  - b. In the Create Workflow dialog box that appears, set Workflow Name and Description.
  - c. Click Create.
3. Expand the created workflow in the left-side navigation pane. Right-click Table, and select Create Table.
4. In the Create Table dialog box that appears, enter a name in Table Name and click Commit.
5. Configure the basic attributes of the table.

| Parameter      | Description                                                       |
|----------------|-------------------------------------------------------------------|
| Display Name   | The alias of the table.                                           |
| Level 1 Folder | The name of the level-1 target folder where the table is located. |

| Parameter      | Description                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------|
| Level 2 Folder | The name of the level-2 target folder where the table is located.                                                           |
| Create Folder  | Click Create Folder to redirect to the Theme Management page. On this page, you can create table folders of levels 1 and 2. |
| Description    | The description of the table.                                                                                               |

## 6. Create a table.

Use either of the following methods to create a table:

- Create a table in DDL mode.



Click Use DDL Statement in the tool bar. In the dialog box that appears, enter a standard statement for creating a table.

After you finish editing the statement, click Generate Table Schema. The Basic Information, Physical Model, and Schema sections are automatically filled.

- Create a table by using a wizard.


If the DDL mode is inappropriate for you to create a table, try using a wizard. The relevant parameters are described as follows.

| Category       | Parameter    | Description                                                                                                                                                                                                 |
|----------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Physical Model | Partitioning | Indicates whether a table is partitioned. Valid values: Partitioned Table and Non-Partitioned Table.                                                                                                        |
|                | Time-to-Live | The time-to-live of data in MaxCompute. The entered number indicates the number of days. Data in a table (or partition) that has not been updated in the specified number of days is automatically cleared. |
|                | Table Level  | Generally, tables are divided into three levels: DW, ODS, and RPT.                                                                                                                                          |

| Category | Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | Categories        | <p>Physically, tables are categorized into basic services, advanced services, and other services.</p> <p>If you need to create a table level, click <b>Create Level</b> to redirect to the Hierarchical Management page.</p> <div>  <b>Note:</b><br/>Physical categories are designed only for your management convenience and do not involve underlying implementation. </div> |
| Schema   | Field Name        | The name of a field. The value contains letters, digits, and underscores (_).                                                                                                                                                                                                                                                                                                                                                                                    |
|          | Display Name      | The alias of a field.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|          | Data Type         | The type of MaxCompute data. Currently, DataWorks supports only the data of STRING, BIGINT, DOUBLE, DATETIME, and BOOLEAN types.                                                                                                                                                                                                                                                                                                                                 |
|          | Description       | The detailed description of a field.                                                                                                                                                                                                                                                                                                                                                                                                                             |
|          | Primary Key Field | The field that serves as the primary key or part of a composite primary key.                                                                                                                                                                                                                                                                                                                                                                                     |
|          | Create Field      | Adds a field.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|          | Delete Field      | <p>Deletes a created field.</p> <div>  <b>Note:</b><br/>If you delete a field from a created table and then commit the table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment. </div>                                                                                          |



| Category | Parameter | Description                                                                                                                                                                                                                                                                |
|----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | Move Up   | Adjusts the field sequence of a table that has not been created. If you adjust the sequence of fields in a created table, DataStudio deletes the created table and creates a new table with the same name . This operation is not permitted in the production environment. |
|          | Move Down | The description is the same as that of the Move Up operation.                                                                                                                                                                                                              |
|          | Add       | Creates a partition for the current table. If you add a partition to a created table, DataStudio deletes the created table and creates a new table with the same name . This operation is not permitted in the production environment.                                     |
|          | Delete    | Deletes a partition. If you delete a partition from a created table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment.                                                       |

| Category                                                                                                                                                                                                                                  | Parameter                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Partition Field Design</b><br><br> <b>Note:</b><br>This parameter is available only when Table Type under Physical Model is set to Partitioned Table. | <b>Data Type</b>                             | We recommend that you use the STRING type globally.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                                                                                                                                                                                                                                           | <b>Partition Key Column Date Format</b>      | The format of a date partition. If the partition field is a date (although the data type may be STRING), select or enter a date format, such as yyymmdd or yyyy-mm-dd.                                                                                                                                                                                                                                                                                                                                                                             |
|                                                                                                                                                                                                                                           | <b>Partition Key Column Date Granularity</b> | The granularity of a date partition. The value can be second, minute, hour, day, month, quarter, or year. You can enter a partition granularity as required. If you need to specify multiple partition granularities, note that a greater granularity corresponds to a higher partition level by default. For example, there are three partitions whose granularities are day, hour, and month, respectively. The multi-level partition hierarchy is as follows: level-1 partition (month), level-2 partition (day), and level-3 partition (hour). |

Commit the table

After editing the schema of a table, you can commit the new table to the development environment and the production environment.

| Operation                                    | Description                                                                                                                                                                                                                                            |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Load from the development environment</b> | If the table has been committed to the development environment, the button is highlighted. After you click the button, the information about the table you create in the development environment overwrites the table information on the current page. |
| <b>Commit to the development environment</b> | The system first checks whether you have configured all the required items on the current editing page. If any item is missing, an alert is triggered and the table cannot be committed.                                                               |
| <b>Load from the production environment</b>  | The detailed information of the table committed to the production environment overwrites the table information on the current page.                                                                                                                    |

| Operation                            | Description                                         |
|--------------------------------------|-----------------------------------------------------|
| Commit to the production environment | The table is created in the production environment. |

## 2.4.10 Ad-hoc nodes

### 2.4.10.1 ODPS SQL nodes

ODPS SQL nodes adopt a syntax similar to SQL, and enables processing TB-level data in batches in distributed mode. It is an online analytical processing (OLAP) application designed to deal with large amounts of data. Each ODPS SQL node can process ten thousands of transactions although the process takes a long time from preparation to submission for each job.

#### 1. Create a business flow.

Click **Ad-Hoc Business Flows** in the left-side navigation pane, right click the **Ad-Hoc Business Flows** folder, and select **Create Business Flow**.

#### 2. Create an ODPS SQL node.

Right-click the **Data Analytics** and choose **Create Data Analytics Node > ODPS SQL**.

#### 3. Edit the code of the ODPS SQL node. The code must conform to the syntax.

#### 4. Schedule the node.

Click **Schedule** on the right of the code editor to open the **Schedule** tab. For more information, see [Schedule](#).

#### 5. Submit the node.

After the node schedule information is complete, click the **Save** in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

#### 6. Publish the node.

For more information, see [Deploy](#).

#### 7. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

## 2.4.10.2 PyODPS nodes

DataWorks is integrated with the MaxCompute SDK for Python. You can specify Python code to process data in MaxCompute.

Create a PyODPS node

**You can create a PyODPS node as follows:**

- 1. Create a business flow.**

**Right-click the Ad-Hoc Business Flows folder on the Ad-Hoc Business Flows tab, and select Create Business Flow.**

- 2. Create a PyODPS node.**

**In the new business flow folder, right-click the Data Analytics folder, and choose Create Data Analytics Node > PyODPS.**

- 3. Configure the PyODPS node.**

- a. Do not specify the entry point.**

**Each PyODPS node contain a global variable "odps" or "o", which is the entry point. Therefore, you do not need to manually specify the entry point.**

```
print(odps.exist_table('PYODPS_iris'))
```

- b. Run SQL statements.**

**You can query data by running MaxCompute SQL statements, and obtain the query results. You can use `execute_sql` and `run_sql` functions to create task instances.**



**Note:**

**You need to call certain functions to run statements that are not directly compatible with the MaxCompute console. For example, statements other than DDL and DML. To run GRANT and REVOKE statements, you need to call the `run_security_query` function. To run PAI commands, you need to call the `run_xflow` or `execute_xflow` function.**

```
o.execute_sql('select * from dual') # (Synchronous mode)
Blocked until the SQL statement is executed.
instance = o.run_sql('select * from dual') # Asynchronous mode.
print(instance.get_logview_address()) # Print the address of
LogView.
```

```
instance.wait_for_success() # Blocked until the SQL statement is executed.
```

#### c. Set runtime parameters.

You can use the `hints` parameter to set the runtime parameters. The type of the `hints` parameter is `DICT`.

```
o.execute_sql('select * from PYODPS_iris', hints={'odps.sql.mapper.split.size': 16})
```

If you set the `sql.settings` parameter, you need to set runtime parameters each time you run the code.

```
from odps import options
options.sql.settings = {'odps.sql.mapper.split.size': 16}
o.execute_sql('select * from PYODPS_iris') # The hints parameter is automatically set based on global settings.
```

#### d. Obtain SQL query results.

You can use the `open_reader` function to obtain query results if the SQL statement returns structured data.

```
with o.execute_sql('select * from dual').open_reader() as reader:
 for record in reader: # Process each record.
```

You can also use this function to obtain raw query results if a `DESC` statement is run.

```
with o.execute_sql('desc dual').open_reader() as reader:
 print(reader.raw)
```



#### Note:

If you use a custom time variable, you need to fix the variable to a time. PyODPS nodes does not support relative time variables.

#### 4. Schedule the node.

Click **Schedule** on the right of the code editor to open the **Schedule** tab. For more information, see [Schedule](#).

#### 5. Submit the node.

After the node schedule information is complete, click the **Save** in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

**6. Publish the node.**

For more information, see [Deploy](#).

**7. Test the node task in the production environment.**

For more information, see [Recurring tasks](#).

### 2.4.10.3 Ad-hoc data synchronization nodes

Currently, data synchronization nodes support the following data source types: MaxCompute, MySQL, PostgreSQL, Oracle, MongoDB, DB2, OTS, OTS Stream, OSS, FTP, HBase, LogHub, HDFS, and Stream. For more information, see [Supported data sources](#).

**1. Create a business flow.**

Right-click the Ad-Hoc Business Flows folder on the Ad-Hoc Business Flows tab, and select Create Business Flow.

**2. Create a data synchronization node.**

In the new business flow folder, right-click the Data Integration folder, and choose Create Data Integration Node > Data Sync.

**3. Configure the data synchronization node.**

You only need to specify the name of the source table and that of the destination table to complete the configuration of a most basic node.

When you enter a table name, the drop-down list displays all matched tables. Currently, only exact match is supported. Ensure that you specify a complete table name. Tables are labeled with Unsupported if they are not supported by data synchronization nodes. If you hover over a table on the list, the details of the table is prompted, including the database, IP address, and owner. After you

select a table, column information is automatically filled in. You can create, move, and delete columns from tables except for MaxCompute tables.

a. Configure synchronization tables.

b. Modify the source table.

In most cases, you do not need to modify the source table. If required, you can modify the source table in the Mappings section. Modification is not supported for MaxCompute source tables.

- Click Add to create a column.
- Click the Delete icon to delete a column.

c. Modify the destination table.

In most cases, you do not need to modify the destination table. If required, you can modify the destination table in the Mappings section. For example, you can modify the destination table if only some of the columns need to be synchronized.



**Note:**

Modification is not supported for MaxCompute source tables. Columns in the source table are mapped to those in the destination table according to the mappings you have configured, instead of based on column names.

d. Configure incremental or full synchronization.

- Format of the Partition value if incremental synchronization is used: `ds=${bizdate}`
- Format of the Partition value if full synchronization is used: `ds=*`



**Note:**

To synchronize multiple shards, you can use simple regular expressions.

- Alternatively, specify the Partition parameter as `ds=20180312 | ds=20180313 | ds=20180314;`
- Data synchronization nodes support synchronizing a range of partitions. To do so, start the Partition parameter value with `/query/`. For example, `/query*/ds>=20180313 and ds<20180315;`
- You need to define the `bizdate` variable as follows: `-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour"`. You also need to define custom

variable if exists. For example, to use `pt=${selfVar}`, define the custom variable `selfVar` as follows: `-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour -DselfVar=xxxx.`

e. Configure field mappings.

Fields are mapped according to the mappings you configure, instead of based on names or types.



**Note:**

You can add columns only if the source table is not a MaxCompute table.

f. Configure the channel.

The Channel section allows you to control the speed and error rate of the data synchronization node.

| Parameter                  | Description                                                                                                                                                                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                        | A data migration unit (DMU) represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.                                                                                                            |
| Concurrent Jobs            | You can specify a maximum number of concurrent threads to read and write data to data storage within a single data synchronization task.                                                                                                                                                 |
| Bandwidth Throttling       | If bandwidth throttling is enabled, you need to specify a maximum transmission rate.                                                                                                                                                                                                     |
| Dirty Data Records Allowed | The maximum number of dirty data records allowed . Dirty data is usually caused by infeasible data type conversion. If you set the value to 0, the task ends when a dirty data record is found. If the value is empty, the task continues as normal no matter whether dirty data exists. |
| Task Resource Group        | The resource on which tasks are run. You can manage resource groups by using the Data Integration service.                                                                                                                                                                               |

4. Schedule the node.

Click **Schedule** on the right of the code editor to open the Schedule tab. For more information, see [Schedule](#).



#### 5. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

#### 6. Publish the node.

For more information, see [Deploy](#).

#### 7. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

### 2.4.10.4 ODPS MR nodes

MaxCompute is interfaced with MapReduce. You can create and run ODPS MR nodes in DataWorks. You can also call MapReduce Java API operations to develop MapReduce programs for processing data in MaxCompute.

You need to upload, submit, and publish required resources before creating ODPS MR nodes.

Create a resource

#### 1. Create a business flow.

Right-click the Ad-Hoc Business Flows folder on the Ad-Hoc Business Flows tab, and select Create Business Flow.

#### 2. In the new business flow folder, right-click the Resource folder, and choose Create Resource > JAR.

#### 3. The Create Resource dialog box appears. Specify a resource name according to the naming convention, set the resource type to JAR, select a local JAR package, and click Submit.



**Note:**

- If the selected JAR package has been uploaded from the ODPS client, deselect Upload to ODPS. Otherwise, an error will occur during the upload process.
- The resource name can be different from the name of the uploaded file.
- Convention for naming resources: A resource name can contain letters (case insensitive), numbers, underscores (\_), and periods (.). It must be 1 to 128 characters in length. A JAR resource name must end with .jar, and a Python resource name must end with .py.

4. Click **Submit** to submit the resource to the development environment.
5. Publish the resource.

For more information, see [Deploy](#).

Create an ODPS MR node

1. Create a business flow.

Right-click the **Ad-Hoc Business Flows** folder on the **Ad-Hoc Business Flows** tab, and select **Create Business Flow**.

2. Create an ODPS MR node.

In the new business flow folder, right-click the **Data Analytics** folder, and choose **Create Data Analytics Node > ODPS MR**.

3. Edit the code of the ODPS MR node. Double-click the new ODPS MR node. The editor of the ODPS MR node appears.

Sample code:

```
jar -resources base_test.jar -classpath ./base_test.jar com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll
```

Code description:

- `-resources base_test.jar`: the name of the JAR resource.
- `-classpath`: the path of the JAR resource. You can right-click the resource and select **Insert Resource** to insert the path of the JAR package into the code.



**Note:**

Ensure that the configuration tab of the ODPS MR node is active when you attempt to insert a JAR package path.

- `com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll`: the main class in the JAR package. It must be consistent as specified in the JAR package.

If you use multiple JAR resources in a single ODPS MR node, separate each resource path with a comma (,) as follows: `-classpath ./xxxx1.jar,./xxxx2.jar`.

4. Schedule the node.

Click **Schedule** on the right of the code editor to open the **Schedule** tab. For more information, see [Schedule](#).

**5. Submit the node.**

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

**6. Publish the node.**

For more information, see [Deploy](#).

**7. Test the node task in the production environment.**

For more information, see [Recurring tasks](#).

## 2.4.10.5 SQL script template

This topic describes how to create and configure SQL script templates in a manually triggered workflow.

### Procedure

**1. Log on to the DataWorks console.**

**2. Create a manually triggered workflow.**

- a. On the DataStudio page, click Manually Triggered Workflows in the left-side navigation bar.
- b. In the Create Workflow dialog box that appears, set Workflow Name and Description.
- c. Click Create.

**3. Create an SQL script template.**

- a. Expand the created workflow in the left-side navigation pane. Right-click Data Analytics, and choose Create Data Analytics Node > SQL Snippet.
- b. In the Create Node dialog box that appears, enter a name in Node Name and click Commit.

To improve development efficiency, you can create data analytics nodes by using the script templates provided by workspace members and tenants.

- The script templates provided by members of this workspace are available on the Workspace-Specific tab.
- The script templates provided by tenants are available on the Public tab.

Specify parameters for the selected script template.

#### 4. Configure the node schedule.

Click **Properties** on the right side, and set the relevant parameters. For more information, see [Properties](#).

#### 5. Commit the node.

After finishing the schedule configuration, click **Save** in the upper-left corner and commit the node to the development environment. After you commit the node, it is unlocked.

#### 6. Deploy the node.

For more information, see [Deploy](#).

#### 7. Test the node in the production environment.

For more information, see [Recurring tasks](#).

### Upgrade the version of an SQL script template

When a new version is deployed for a script template, you can decide whether to upgrade the version of the script template used in your nodes to the latest version.

The script template upgrade mechanism allows developers to continuously upgrade script template versions. This mechanism enhances the process execution efficiency and optimizes the business performance. For example:



User A uses the version V1.0 of a script template that belongs to user B. User B then commits a new version for the script template V2.0. User A receives a notification of the new version. After comparing the code of the two versions, user A can decide whether to upgrade the script template version to the latest version.

SQL script templates are easy to upgrade because they are developed based on a template. To upgrade the SQL script template version, properly set parameters for the SQL script template of the new version according to the version description. Then, save the node and commit it for deployment.

### GUI features

The following table describes the GUI features.

| No. | Feature | Description                                                           |
|-----|---------|-----------------------------------------------------------------------|
| 1   | Save    | Click the button to save the settings of the current script template. |

| No. | Feature             | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2   | Submit              | Click the button to commit the script template to the development environment.                                                                                                                                                                                                                                                                                                                                                   |
| 3   | Submit and Unlock   | Click the button to commit the node and unlock the node to edit the code.                                                                                                                                                                                                                                                                                                                                                        |
| 4   | Steal Lock          | Click the button to steal the lock of the node and then edit it if you are not the owner of the script template.                                                                                                                                                                                                                                                                                                                 |
| 5   | Run                 | Click the button to run the script template in the local development environment.                                                                                                                                                                                                                                                                                                                                                |
| 6   | Run with Parameters | <p>Click the button to run the code of the node using the parameters configured for the node.</p> <div>  <b>Note:</b><br/>This feature is not available on a shell node. </div>                                                                                                                                                                 |
| 7   | Stop                | Click the button to stop a script template from running.                                                                                                                                                                                                                                                                                                                                                                         |
| 8   | Reload              | <p>Click the button to refresh the page and restore the last saved status. Unsaved content will be lost.</p> <div>  <b>Note:</b><br/>If you have enabled the cache in the configuration center, you are notified of the code that is cached but not saved after the page is refreshed. In this case, select the version that you need. </div> |
| 9   | Run Smoke Test      | Click the button to conduct a smoke test on the current node.                                                                                                                                                                                                                                                                                                                                                                    |
| 10  | View Smoke Test Log | Click the button to view the runtime logs of the current node.                                                                                                                                                                                                                                                                                                                                                                   |
| 11  | Parameters          | Click the button to configure script template information, request parameters, and response parameters.                                                                                                                                                                                                                                                                                                                          |
| 12  | Properties          | Click the button to set the owner, description, parameters, and resource group of a node.                                                                                                                                                                                                                                                                                                                                        |
| 13  | Lineage             | Click the button to view the map of lineage and dependencies between the SQL script templates.                                                                                                                                                                                                                                                                                                                                   |

| No. | Feature  | Description                                                            |
|-----|----------|------------------------------------------------------------------------|
| 14  | Versions | Click the button to show the deployed versions of the script template. |

### 2.4.10.6 Zero-load node

A zero-load node is a control node, which only supports dry-run scheduling and does not generate any data. It usually serves as the root node of a workflow.

Create a zero-load node

1. Log on to the DataWorks console.
2. Create a manually triggered workflow.
  - a. On the DataStudio page, click Manually Triggered Workflows in the left-side navigation bar.
  - b. In the Create Workflow dialog box that appears, set Workflow Name and Description.
  - c. Click Create.
3. Expand the created workflow in the left-side navigation pane. Right-click Data Analytics, and choose Create Data Analytics Node > Zero-Load Node.
4. In the Create Node dialog box that appears, set the relevant parameters and click Commit.
5. Configure the node schedule.

You do not need to edit the code of the zero-load node. Click Properties on the right side, and set the relevant parameters. For more information, see [Properties](#).

6. Commit the node.

After finishing the schedule configuration, click Save in the upper-left corner and commit the node to the development environment. After you commit the node, it is unlocked.

7. Deploy the node.

For more information, see [Deploy](#).

8. Test the node in the production environment.

For more information, see [Recurring tasks](#).

### 2.4.10.7 Shell nodes

Shell nodes support standard shell syntax but not interactive shell syntax. Shell nodes can be run on the default resource group. If your shell nodes need to access an IP address or a domain name, add the IP address or domain name to the sandbox whitelist on the Project Management tab.

#### Procedure

1. Create a business flow.

Click Ad-Hoc Business Flows folder in the left-side pane, and select Create Business Flow.

2. Create a shell node.

Right-click the Data Analytics folder and choose Create Data Analytics Node > Shell.

3. Set Node Type to Shell, specify the node name, select the target folder, and click Submit.

4. Edit the code of the shell node.

Edit code in the editor of the shell node.

If you need to use relative time parameters, use the following statement:

```
echo "$1 $2 $3"
```



**Note:**

Separate parameters with a space character. For more information about relative time parameters, see [Parameter configuration](#).

5. Submit the node.

After the node schedule information is complete, click the Save in the tool bar and submit the node to the development environment. After the node is submitted, it is unlocked.

6. Publish the node.

For more information, see [Deploy](#).

7. Test the node task in the production environment.

For more information, see [Recurring tasks](#).

## Scenarios

If you use shell nodes to connect to databases:

- If the database is hosted on Alibaba Cloud and the region is China (Shanghai), whitelist the following IP addresses for the database:

10.152.69.0/24, 10.153.136.0/24, 10.143.32.0/24, 120.27.160.26, 10.46.67.156, 120.27.160.81, 10.46.64.81, 121.43.110.160, 10.117.39.238, 121.43.112.137, 10.117.28.203, 118.178.84.74, 10.27.63.41, 118.178.56.228, 10.27.63.60, 118.178.59.233, 10.27.63.38, 118.178.142.154, 10.27.63.15, and 100.64.0.0/8

**Note:**

If the database is hosted on Apsara Stack but the region is not China (Shanghai), we recommend that you connect to the database over a public network.

Alternatively, create an ECS instance in the same region as the database, add the ECS instance to DataWorks as a custom resource group, and run the shell node on the custom resource group.

- If the database is hosted on the premises, we recommend that you connect to the database over a public network and whitelist the preceding IP address for the database.

**Note:**


If the shell node is run on a custom resource group, you also need to whitelist the server IP address of the custom resource group for the database.

## 2.4.11 Configure parameters for ad-hoc tasks

### 2.4.11.1 Basic Information

| Parameter | Description                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Node Name | The name of a node that you specify when creating the node. To modify the name, right-click the node in the left-side pane and select Rename. |
| Node ID   | The unique ID of a node generated when a task is submitted at the first time. The node ID cannot be modified.                                 |
| Node Type | The name of a node that you specify when creating the node. The node type cannot be modified.                                                 |



| Parameter   | Description                                                                                                                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Owner       | <p>The owner of a node, which defaults to the user who created the node. You can select a new owner from the drop-down list.</p> <div> <b>Note:</b><br/>You can only select a member in the workspace where the node resides.</div> |
| Description | The description of the node about the business and usage.                                                                                                                                                                                                                                                            |
| Parameters  | <p>The parameters assigned to the variables used in the code.</p> <p>For example, if you specify <code>pt=\${datetime}</code> to indicate time, then specify the Parameters as <code>datetime=\$bizdate</code>.</p>                                                                                                  |

Format of the Parameters value

- **Format for ODPS SQL, ODPS PL, and ODPS MR nodes:** `Variable name 1=Parameter 1 Variable name 2=Parameter 2....` Separate multiple parameters by a space.
- **Format for shell nodes:** `Parameter 1 Parameter 2....` Separate multiple parameters by a space.
- **Data synchronization nodes:** No template is automatically provided. You can specify the Parameters in the format of `-p " "`. Add `-Dvariable_name=value` between the double quotes.



**Note:**

**Template:**

```
-p "-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour -
Dvariable_name=${time_expression}"
```

Replace `variable_name` with a variable name, and replace `time_expression` with a time expression in the format of `YYYYMMDD`.

For more information about built-in time parameters, see [Parameter configuration](#).

## 2.4.11.2 Configure parameters for ad-hoc nodes

DataWorks provides the parameter configuration feature to ensure that ad-hoc tasks can adapt to environment changes. Note the following two points when you configure parameters:

- Do not add spaces on either side of an equation mark (=). Example: bizdate=\$bizdate.
- Separate multiple parameters by spaces.

System variables

DataWorks provides two system variables.

- `${bdp.system.cyctime}`: the time to run an instance. Default format: YYYYMMDDHH24miss.
- `${bdp.system.bizdate}`: the timestamp of data to be analyzed. Default format: YYYYMMDD. By default, the task is run before the timestamp date.

**Formula:** timestamp of data to process = time to run an instance - 1.

You can use system variables such as `${bizdate}` in your code without additional settings.



**Note:**

The execution time is configured for each recurring task. You can calculate the timestamp based on the execution time.

Example

The following example sets an ODPS\_SQL task. You can use system variables in the following statement:

```
insert overwrite table tb1 partition(ds ='20180606') select
c1,c2,c3
from (
select * from tb2
where ds='${bizdate}');
```

Configure relative time parameters in non-shell nodes



**Note:**

**A variable name in SQL code can contain letters, numbers, and underscores (\_).  
The value of a variable named date is fixed to \$bizdate even if you change the value in your code.**

**For a node whose type is not shell, you must first add \${variableName} in the code, and then assign a value to the variable on the Schedule tab.**

**For example, add \${variableName} in the code of an ODPS SQL node and set the variable in the format of variableName=systemVariable on the Schedule tab.**

**Parameters must be assigned to variables used in the code.**

Configure relative time parameters in non-shell nodes

**The parameter configuration procedure of a shell node is similar to that of a non-shell node, but the variable naming rules are different. Variable names for a shell node cannot be customized and must be named in the format of '\$1,\$2,\$3'.**

**For example, add the variable \$1 in the code of a shell node and configure the node parameter as \$xxx, which is a built-in parameter. The value of \$xxx replaces \$1 in the code.**

**Parameters must be assigned to variables used in the code.**



**Note:**

**For a shell node, if more than 10 parameters are referenced, you must use \${10} to declare variables.**

Configure a variable with a fixed value

**Take an SQL node as an example. Add \${variable name} in the code of the SQL node and configure the parameter for the node in the format of variable name=fixed value.**

**Code: select xxxxxx type='\${type}'**

**Value assigned to the variable: type='aaa'**

**When the node is scheduled, the variable in the code is replaced by type='aaa'.**

Configure a variable with a built-in parameter

**Take an SQL node for example. Add \${variable name} in the code of the SQL node and configure the parameter for the node in the format of variable name=parameter.**

**Code:** select xxxxxx dt=\${datetime}

**Value assigned to the variable:** datetime=\$bizdate

When the node is scheduled, the variable in the code is replaced by dt=20170721 if the current date is July 22, 2017.

Built-in parameters

**\$bizdate:** business date in the format of yyyyymmdd.



**Note:**

The parameter value is by default the date that is one day before the execution date.

For example, in the code of an ODPS SQL node, pt=\${datetime}. The parameter for the node is configured as datetime=\$bizdate. If the node is executed on July 22, 2017, \$bizdate is replaced by 20170721.

**Example:** In the code of an ODPS SQL node, pt=\${datetime}. The parameter for the node is configured as datetime=\$gmtdate. If the node is executed on July 22, 2017, \$gmtdate is replaced by 20170722.

For example, in the code of an ODPS SQL node, pt=\${datetime}. The parameter for the node is configured as datetime=\$bizdate. If the node is executed on July 22, 2017, \$bizdate is replaced by 20170721.

For example, in the code of an ODPS SQL node, pt=\${datetime}. The parameter for the node is configured as datetime=\$gmtdate. Today is July 1, 2017. When the node is executed today, \$gmtdate is replaced by pt=20170701.

**\$cyctime:** The time at which the task is scheduled to run. If no scheduled time is configured for a daily task, cyctime is set to 00:00 of the current day. The time is accurate to the second. This parameter is usually used for tasks scheduled by hour or minute. Example: cyctime=\$cyctime.



**Note:**

Pay attention to the difference between the time parameters configured by using \$[] and \${}. \$bizdate: the business date, which is one day before the current time by default. \$cyctime: the scheduled time of the task. If no scheduled time is configured for a daily task, cyctime is set to 00:00 of the current day. The time is accurate to the second. This parameter is usually used for tasks scheduled by hour

or minute. For example, if a task is scheduled to run on 00:30, the scheduled time is yyyy-mm-dd 00:30:00. If the time parameter is configured by using \${}, bizdate is used as the benchmark for running tasks. For example, the time parameter is replaced by the business date selected for retroactive execution. If the time parameter is configured using [], cyctime is used as the benchmark for running tasks. For more information about how to use the parameter, see the following descriptions. The parameter value is replaced with the date that is one day after the business date selected for the retroactive execution. For example, if the date of 20140510 is selected as the business date, the cyctime parameter value is 20140511.

**\$jobid:** the ID of the workflow to which a task belongs. Example: jobid=\$jobid.

**\$nodeid:** the ID of the node. Example: nodeid=\$nodeid.

**\$taskid:** the ID of the task, which is the ID of the node instance. Example: taskid=\$taskid.

**\$bizmonth:** the month of the business date in the format of yyymm.

- **Note:** If the month of a business date is the current month, \$bizmonth = the month of the business date - 1. If the month of a business date is not the current month, \$bizmonth = the month of the business date.
- **Example:** In the code of the ODPS SQL node, pt=\${datetime}. The parameter for the node is configured as datetime=\$bizmonth. If the node is executed on July 22, 2017, \$bizmonth is replaced by 201706.

**\$gmtdate:** the current date in the format of yyymmdd.

The value of this parameter is the current date by default. For retroactive executions, specify the parameter value as the date that is one day after the business date.

**User-defined parameter \${...} description**

- Customize a time format as the value of \$bizdate, where yyyy indicates a 4-digit year, yy indicates a 2-digit year, mm indicates the month, and dd indicates the day. You can specify the parameter in any combination. For example, \${yyyy}, \${yyyymm}, \${yyyymmdd}, \${yyyy-mm-dd}.
- \$bizdate is accurate to day. Therefore, the user-defined parameter \${...} can only represent the year, month, or day.

- You can specify a time as follows:

Next N years: `${yyyy+N}`

Previous N years: `${yyyy-N}`

Next N months: `${yyyymm+N}`

Previous N months: `${yyyymm-N}`

Next N weeks: `${yyyymmdd+7*N}`

Previous N weeks: `${yyyymmdd-7*N}`

Next N days: `${yyyymmdd+N}`

Previous N days: `${yyyymmdd-N}`

`${yyyymmdd}`: the business date in the format of `yyyymmdd`. The value is the same as that of `$bizdate`.

- **Note:** The parameter value is by default the date that is one day before the execution date. The format of this parameter can be customized. For example, the date format of `${yyyy-mm-dd}` is `yyyy-mm-dd`.
- **Example:** In the code of an ODPS SQL node, `pt=${datetime}`. The parameter for the node is configured as `datetime=${yyyymmdd}`. If the node is executed on 22 July, 2013, `${yyyymmdd}` is replaced by 20130721.

`${yyyymmdd-/ +N}`: N days before or after `yyyymmdd`.

`${yyyymm-/ +N}`: N months before or after `yyyymm`.

`${yyyy-/ +N}`: N years before or after the year (yyyy).

`${yy-/ +N}`: N years before or after the year (yy).



**Note:**

`yyyymmdd` indicates the business date. Any separator can be used in the date format, such as `yyyy-mm-dd`. The above parameters are based on the days, months and years of the business date.

**Example:**

- In the code of an ODPS SQL node, `pt=${datetime}`. The parameter for the node is configured as `datetime=${yyyy-mm-dd}`. If the node is executed on July 22, 2018, `${yyyy-mm-dd}` is replaced by 2018-07-21.

- In the code of an ODPS SQL node, `pt=${datetime}`. The parameter for the node is configured as `datetime=${yyyymmdd-2}`. If the node is executed on July 22, 2018, `${yyyymmdd-2}` is replaced by 20180719.
- In the code of an ODPS SQL node, `pt=${datetime}`. The parameter for the node is configured as `datetime=${yyyymm-2}`. If the node is executed on July 22, 2018, `${yyyymm-2}` is replaced by 201805.
- In the code of an ODPS SQL node, `pt=${datetime}`. The parameter for the node is configured as `datetime=${yyyy-2}`. If the node is executed on July 22, 2018, `${yyyy-2}` is replaced by 2016.

You can configure multiple parameters for an ODPS SQL node. For example, `startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}`.

Example: (Assume that `$cyctime=20140515103000`)

- `[$yyyy] = 2014`, `[$yy] = 14`, `[$mm] = 05`, `[$dd] = 15`, `[$yyyy-mm-dd] = 2014-05-15`, `[$hh24:mi:ss] = 10:30:00`, `[$yyyy-mm-dd hh24:mi:ss] = 2014-05-1510:30:00`
- `[$hh24:mi:ss - 1/24] = 09:30:00`
- `[$yyyy-mm-dd hh24:mi:ss - 1/24/60] = 2014-05-1510:29:00`
- `[$yyyy-mm-dd hh24:mi:ss - 1/24] = 2014-05-1509:30:00`
- `[$add_months(yyyymmdd,-1)] = 2014-04-15`
- `[$add_months(yyyymmdd,-12*1)] = 2013-05-15`
- `[$hh24] = 10`
- `[$mi] = 30`

Method for checking the `$cyctime` parameter:

Jump to the Operation Center page. Right-click the DAG of the cycle instance and select More. Check whether the scheduled time displayed on the Attributes tab page is the time at which the instance runs periodically.

The Execution Parameter value contains a time that is one hour earlier than the scheduled time.

**Note**

## 2.4.12 Components

### 2.4.12.1 Create components

#### Description

**A component is an SQL code template with multiple input and output parameters. Each component converts one or more input tables into an output table by filtering, joining, and aggregating.**

#### Benefits

**Many SQL statements process input and output table with the same or compatible schema. To reuse SQL code in this case, you can develop code templates as SQL components with input and output parameters to specify input and output tables.**

**After you have created SQL components, you can create SQL component nodes to process input and output tables without reproducing code, which significantly improves development efficiency. The method of scheduling SQL component nodes is the same as that of scheduling SQL nodes.**

#### Composition

**Each component consists of input parameters, output parameters, and a body.**

#### Input parameters

**The attributes of an input parameter includes its name, type, description, and schema. The type can be table or string.**

- **A table-type parameter specifies a table to be used in a component.**
- **A string-type parameter controls a variable in a component. For example, if you need an output table to include N cities with highest sales volume in each region, then you can set a string-type parameter to control the number N.**

**If you need an output table to include the sales volume in only one province, then you can set a string-type parameter to control the province.**

- **The description indicates the meaning of a parameter.**



- The schema indicates the schema of the table, which is specified in text. This attribute is required only for table-type parameters. The component works properly only if you input a table that has the same or compatible schema.

The input table must contain fields that match the specified names and types with no limitation on the sequence. The input table can also have other fields. The specified schema is for reference only.

- We recommend that you set the table schema in the following format:

```
Name of field 1 Type of field 1 Description of field 1
Name of field 2 Type of field 2 Description of field 2
Name of field n Type of field n Description of field n
```

**Example:**

```
area_id string 'Region ID'
city_id string 'City ID'
order_amt double 'Order amount'
```

Output parameters

- The attributes of an output parameter includes its name, type, description, and schema. The type can only be table.
- A table-type parameter specifies a table that is output by a component.
- The description indicates the meaning of a parameter.
- The schema indicates the schema of the table, which is specified in text.

You must specify an output table with the same number of fields and with a compatible schema. Otherwise, an error occurs. The field names in the output table do not need to be consistent with those defined by the table parameter. The specified schema is for reference only.

- We recommend that you set the table schema in the following format:

```
Name of field 1 Type of field 1 Description of field 1
Name of field 2 Type of field 2 Description of field 2
Name of field n Type of field n Description of field n
```

**Example:**

```
area_id string 'Region ID'
city_id string 'City ID'
order_amt double 'Order amount'
rank bigint 'Ranking'
```

Body

To use a parameter, specify as follows: @@{Parameter name}.

The body is a group of SQL statements that converts an input table into a meaningful output table based on other input parameters.

You need to ensure that component SQL statements are correct and executable no matter how the input and output parameters are specified.

Example

You can create a component after specifying the component name and description in the Create component dialog box.

Example: Input table schema

The following table describes the schema of the input MySQL table.

| Column name      | Data type | Description         |
|------------------|-----------|---------------------|
| order_id         | varchar   | Order ID            |
| report_date      | datetime  | Order date          |
| customer_name    | varchar   | Customer name       |
| order_level      | varchar   | Order level         |
| order_number     | double    | Order quantity      |
| order_amt        | double    | Order amount        |
| back_point       | double    | Discount            |
| shipping_type    | varchar   | Shipping method     |
| profit_amt       | double    | Profit amount       |
| price            | double    | Unit price          |
| shipping_cost    | double    | Shipping cost       |
| area             | varchar   | Region              |
| province         | varchar   | Province            |
| city             | varchar   | City                |
| product_type     | varchar   | Product type        |
| product_sub_type | varchar   | Product sub-type    |
| product_name     | varchar   | Product name        |
| product_box      | varchar   | Product packing box |
| shipping_date    | datetime  | Transportation date |

Example: Feature

**Component name:** get\_top\_n

**Description:** This component outputs the ranking of a specified number (data type: string) of cities with the highest sales amount based on a table (data type: table).

Example: Parameters

**Input parameter 1:**

**Name:** myinputtable, **type:** table

**Input parameter 2:**

**Name:** topn, **type:** string

**Input parameter 3:**

**Name:** myoutput, **type:** table

**Parameter definition:**

**area\_id** string

**city\_id** string

**order\_amt** double

**rank** bigint

**CREATE TABLE statement:**

```
CREATE TABLE IF NOT EXISTS company_sales_top_n
(
 area STRING COMMENT 'Region',
 city STRING COMMENT 'City',
 sales_amount DOUBLE COMMENT 'Sales volume',
 rank BIGINT COMMENT 'Ranking'
)
COMMENT 'Sales amount ranking within the company'
PARTITIONED BY (pt STRING COMMENT '')
LIFECYCLE 365;
```

Example: Body

```
INSERT OVERWRITE TABLE @@{myoutput} PARTITION (pt='${bizdate}')
 SELECT r3.area_id,
 r3.city_id,
 r3.order_amt,
 r3.rank
 from (
 SELECT
 area_id,
 city_id,
 rank,
```

```

 order_amt_1505468133993_sum as order_amt ,
 order_number_1505468133991_sum,
 profit_amt_1505468134000_sum
FROM
 (SELECT
 area_id,
 city_id,
 ROW_NUMBER() OVER (PARTITION BY r1.area_id ORDER BY r1.order_amt_
1505468133993_sum DESC)
 AS rank,
 order_amt_1505468133993_sum,
 order_number_1505468133991_sum,
 profit_amt_1505468134000_sum
 FROM
 (SELECT area AS area_id,
 city AS city_id,
 SUM(order_amt) AS order_amt_1505468133993_sum,
 SUM(order_number) AS order_number_1505468133991_sum,
 SUM(profit_amt) AS profit_amt_1505468134000_sum
 FROM
 @@{myinputtable}
 WHERE
 SUBSTR(pt, 1, 8) IN ('${bizdate}')
 GROUP BY
 area,
 city)
 r1) r2
 WHERE
 r2.rank >= 1 AND r2.rank <= @@{topn}
 ORDER BY
 area_id,
 rank limit 10000) r3;

```

Share components

**Components are categorized into two types: workspace-specific and public.**

**After a component is submitted, all members in the current workspace can use this component. You can click Publish Component to make a component available for all workspaces of the current tenant.**

Use components

**For more information about how to use components, see [Use components](#).**

Usage records

**You can click Reference Records on the right of the component editor to view usage records.**

## 2.4.12.2 Use components

**To improve development efficiency, you can create data analytics nodes based on components that are created by workspace and organization members.**

- The **Workspace-Specific** tab lists the components created by workspace members.
- The **Public** tab lists the components created by organization members.

For more information, see [SQL component nodes](#).

#### Component configuration wizard

The component configuration wizard is described as follows:

| No. | Icon or Tab          | Description                                                                                                            |
|-----|----------------------|------------------------------------------------------------------------------------------------------------------------|
| 1   | Save                 | Save the settings of the component.                                                                                    |
| 2   | Steal Lock           | Change a component that is not owned by you.                                                                           |
| 3   | Submit               | Submit the component to the development environment.                                                                   |
| 4   | Publish Component    | Publish a component to the entire organization so that all members in the organization can view and use the component. |
| 5   | Parse I/O Parameters | Parse input and output parameters from the code.                                                                       |
| 6   | Precompile           | Edit custom and system parameters for the component.                                                                   |
| 7   | Run                  | Run the component in the development environment.                                                                      |
| 8   | Stop                 | Stop running the component.                                                                                            |
| 9   | Format               | Format the code based on keywords.                                                                                     |
| 10  | Parameters           | View and configure the component information, input parameters, and output parameters.                                 |
| 11  | Version              | View the published versions of the component.                                                                          |
| 12  | Reference Records    | View the use records of the component.                                                                                 |

### 2.4.13 Query

The query tab lets you check your code for errors and check whether your code works as expected in the development environment. Query nodes does not support submitting, publishing, or scheduling. If you need nodes schedulable, create nodes on the Data Analytics tab.

## Create a folder

1. Click Query in the left-side navigation pane. Hover over the Create icon in the tool bar of the left-side pane, and select Folder from the drop-down list.
2. Specify a folder name, select a destination folder, and click Submit.



**Note:**

Folder hierarchy is supported. You can place the folder in another folder that has been created.

## Create a node

Only shell nodes and ODPS SQL nodes are supported on the Query tab.

The following section describes how to create an ODPS SQL node as an example. Right-click a folder name and choose Create Node > ODPS SQL.

| No. | Feature             | Description                                                                                                                                                                                                                                                                                                                                                                     |
|-----|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Save                | Save the code for the node.                                                                                                                                                                                                                                                                                                                                                     |
| 2   | Steal Lock          | Edit a node that is not owned by you.                                                                                                                                                                                                                                                                                                                                           |
| 3   | Run                 | Run the code in the development environment.                                                                                                                                                                                                                                                                                                                                    |
| 4   | Run with Parameters | Run the code after specifying parameters.<br> <b>Note:</b><br>This feature is unavailable for shell nodes.                                                                                                                                                                                   |
| 5   | Stop                | Stop running the code.                                                                                                                                                                                                                                                                                                                                                          |
| 6   | Reload              | Reload the code. The code will be restored to the version last saved, and unsaved changes will be lost.<br> <b>Note:</b><br>If you have enabled caching in the Configuration Center, a dialog box appears, which lets you to choose from the cached code version and the saved code version. |
| 7   | Format              | Format the code based on keywords to avoid excessively long code in single lines.                                                                                                                                                                                                                                                                                               |

## 2.4.14 Runtime Log

The Runtime Log pane lists all logs of tasks run with DataStudio in the past three days. You can click a log to view the runtime log and filter the log by task status.



**Note:**

Runtime logs are retained for three days.

View runtime logs

1. Click Runtime Log in the left-side navigation pane. By default, the pane that appears lists all tasks regardless of task status.
2. Click the All drop-down list and select a status.
3. Click a log to open the runtime log tab. The tab shows the runtime log and the corresponding code.

Save the log to a temporary file

If you need to save the related SQL statements, click the Save as Query File icon in the tool bar.

Specify a file name and a destination folder, and click Submit.

## 2.4.15 Public Tables

On the Public Tables tab, you can view tables in all workspaces of the current tenant account.

- **Workspace:** the workspace name. A prefix of odps is added to each workspace name. For example, if a workspace name is test, then the value in the Workspace column is odps.test.
- **Table Name:** the name of the table in the workspace.

Column information, partition information, and data preview are available in the lower part of the left-side pane after you click a table name.

- **Column:** displays the name, data type, and description for each column.
- **Partition:** displays partition information. A maximum of 60,000 partitions are supported. If you have specified the time-to-live for partitions, the number of partitions depends on the time-to-live.
- **Preview:** displays a preview on data.

Switch between the development environment and the production environment

**Public tables are available either in the development or production environment. A rectangular in blue indicates the current environment. If you click the other rectangular, the environment is switched.**

## 2.4.16 Tables

Create tables

1. Click Tables in the left-side navigation pane.
2. Click the Create icon to create a table.
3. Specify a table name and click Submit.



**Note:**

**Currently, only MaxCompute tables are supported.**

4. Specify basic information.

| Parameter or command | Description                                                                             |
|----------------------|-----------------------------------------------------------------------------------------|
| Display Name         | The display name of the table.                                                          |
| Level 1 Topic        | The name of the level 1 folder where the table is located.                              |
| Level 2 Topic        | The name of the level 2 folder where the table is located.                              |
| Description          | The description of the table.                                                           |
| Create Topic         | Jump to the Config Center page. On this page, you can create folders of levels 1 and 2. |

5. Create a table in DDL mode.

Click DDL Mode in the tool bar. In the dialog box that appears, enter standard CREATE TABLE statements.

After you enter the statements, click Generate Table Structure and the Basic Information, Physical Model, and Table Structure sections are automatically completed.



## 6. Create a table in wizard mode.

You can create a table in wizard mode. The parameters are described as follows:


- **Physical Model**

| Parameter      | Description                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Partitioning   | Indicates whether the table is partitioned. Valid values: Partitioned Table and Non-Partitioned Table.                                      |
| Time-to-Live   | The time-to-live of data in MaxCompute. Data in a table or partition that has not been updated for the specified number of days is deleted. |
| Table Level    | The level of the table. Valid values: DW, ODS, and RPT .                                                                                    |
| Table Category | The category of the table. If you need to create a table level, click Create Level.                                                         |
| Table Type     | This parameter defaults to Internal Table.                                                                                                  |

- **Table Structure**

- **Configure fields in the table.**

| Parameter or command | Description                                                                     |
|----------------------|---------------------------------------------------------------------------------|
| Field English Name   | The name of the field, which can contain letters, numbers, and underscores (_). |
| Display Name         | The display name of the field.                                                  |
| Field Type           | The MaxCompute data type.                                                       |
| Length or Settings   | The length of the field.                                                        |
| Description          | The description of the field.                                                   |
| Primary Key          | The field that serves as the primary key or part of a composite primary key.    |

| Parameter or command | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actions              | <p>Two buttons are available: Save and Delete.</p> <ul style="list-style-type: none"> <li>■ Click the Save icon to save the settings for a field.</li> <li>■ Click the Delete icon to delete a field.</li> </ul> <div>  <p><b>Note:</b><br/>If you delete a field from a created table and then submit the table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment.</p> </div> |
| Move Up              | Adjust the sequence of fields in the table. If you adjust the sequence of fields in a created table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment.                                                                                                                                                                                                                                                                                                          |
| Move Down            | Adjust the sequence of fields in the table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

- Add Partition


Add a partition to the table. If you add a partition to a created table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment.



**Note:**

This section is available only if you specify Partitioning as Partitioned Table in the Physical Model section.

| Parameter or command  | Description                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------|
| Field English Name    | The name of the field.                                                                  |
| Field Type            | We recommend that you use the String type for all fields.                               |
| Length                | The length of the field.                                                                |
| Description           | The description of the field.                                                           |
| Partition Date Format | If a partition field indicates a date, select or enter a date format, such as YYYYMMDD. |

| Parameter or command  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Partition Granularity | <p>Valid values: Second, Minute, Hour, Day, Month, Quarter, and Year.</p> <p>If multiple partition granularities are required , then the larger the granularity, the higher the partition level. For example, if three partition granularities (hour, day, and month) exist, then data is partitioned by month at level 1, by day at level 2, and by hour at level 3.</p>                                                                                                |
| Actions               | <ul style="list-style-type: none"><li>■ Click the Save icon to save the settings for a field.</li><li>■ Click the Delete icon to delete a field.</li></ul> <div> <b>Note:</b><br/>If you delete a partition from a created table, DataStudio deletes the created table and creates a new table with the same name. This operation is not permitted in the production environment.</div> |

#### Submit a table

After editing the schema of a table, submit table to the development or production environment.

- If the table has been submitted to the development environment, you can click the Load from Development Environment button to update the current tab to the table information saved in the development environment.
- After you click the Submit to Development Environment button, DataStudio checks whether all required parameters on the current tab are specified. If any parameter is unspecified, an alert is prompted and the submission fails.
- After you click the Load from Production Environment button, the current page is updated to the table information saved in the production environment.
- After you click the Submit to Production Environment button, DataStudio creates the table in the MaxCompute project of the production environment.

## Query tables by environment

**In the Tables pane, you can select Development Environment or Production Environment to query tables. Query results are organized in folders.**

- **If you select Development Environment, the Tables pane displays a list that only includes tables saved in the development environment.**
- **If you select Production Environment, the Tables pane displays a list that only includes tables saved in the production environment. Use caution while operating tables in the production environment.**

## 2.4.17 Functions

**The functions tab lists all available system functions by category. It also provides the instructions and parameter descriptions for each function.**

**The functions tab categorizes the functions into six groups: string functions, mathematical functions, date functions, analytic functions, aggregate functions, and other functions. After you click a function, you can view the instructions and parameter descriptions of this function in the Description area.**

## 2.4.18 Recycle bin

**The Recycle Bin pane appears if you click Recycle Bin in the left-side navigation pane.**

**The Recycle Bin tab displays a list of nodes that have been deleted from the current workspace. After you right-click a node, you can select to restore or destroy the node.**

**Click the My Files icon in the tool bar of the Recycle Bin pane to view all your nodes that have been deleted.**



### **Note:**

**A node cannot be restored after it is deleted from the recycle bin.**

## 2.4.19 Editor keyboard shortcuts

**This section describes keyboard shortcuts available for the code editor.**

### Google Chrome in Windows OS

**Ctrl + S: Save changes to a node.**

**Ctrl + Z: Undo an action.**

**Ctrl + Y: Redo an action.**

**Ctrl + D: Select occurrences.**

**Ctrl + X: Cut a line.**

**Ctrl + Shift + K: Delete a line.**

**Ctrl + C: Copy a line.**

**Ctrl + I : Select a line.**

**Alt + Shift + Drag: Select a block.**

**Alt + Click: Insert an additional cursor.**

**Ctrl + Shift + L: Select all occurrences.**

**Ctrl + F: Search for text in a node.**

**Ctrl + H: Replace text in a node.**

**Ctrl + G: Locate a line.**

**Alt + Enter: Select all matched strings.**

**Alt + Up or down arrow: Move a line up or down.**

**Alt + Shift + Up or down arrow: Duplicate a line.**

**Ctrl + Shift + K: Delete a line.**

**Ctrl + Enter or Ctrl + Shift + Enter: Insert a line break downwards or upwards.**

**Ctrl + Shift + Back slash (\): Jump to the parenthesis, bracket, or brace that matches the adjacent one.**

**Ctrl + Left bracket ([]) or right bracket ([]): Increase or decrease the indent of a line.**

**Home or End: Move the cursor to the beginning or end of a line.**

**Ctrl + Home or End: Move the cursor to the top or bottom of a node.**

**Ctrl + Left or Right arrow: Move the cursor one word to left or right.**

Ctrl + Shift + Left bracket (]) or right bracket ([): **Hide or show a block.**

Ctrl + K + Left bracket (]) or right bracket ([): **Hide or show sub-blocks in a block.**

Ctrl + K + 0 or J: **Hide or show all blocks.**

Ctrl + Slash (/): **Comment out or uncomment the selected lines or blocks.**

#### Google Chrome in Mac OS

Command-S: **Save changes to a node.**

Command-Z: **Undo an action.**

Command-Y: **Redo an action.**

Command-D: **Select occurrences.**

Command-X: **Cut a line.**

Shift-Command-K: **Delete a line.**

Command-C: **Copy a line.**

Command-I: **Select a line.**

Command-F: **Search for text in a node.**

Option-Command-F: **Replace text in a node.**

Option-Up or down arrow: **Move a line up or down.**

Option-Shift-Up or down arrow: **Duplicate a line.**

Shift-Command-K: **Delete a line.**

Command-Enter or Shift-Command-Enter: **Insert a line break downwards or upwards.**

Shift-Command-Back slash (\): **Jump to the parenthesis, bracket, or brace that matches the adjacent one.**

Command-Left bracket (]) or right bracket ([): **Increase or decrease the indent of a line.**

Command-Left or right arrow: **Move the cursor to the beginning or end of a line.**

Command-Up or down arrow: **Move the cursor to the top or bottom of a node.**

**Option-Left or right arrow: Move the cursor one word to left or right.**

**Option-Command-Left bracket (]) or right bracket ([): Hide or show a block.**

**Command-K-Left bracket (]) or right bracket ([): Hide or show sub-blocks in a block.**

**Command-K-0 or J: Hide or show all blocks.**

**Command-Slash (/): Comment out or uncomment the selected lines or blocks.**

Insert multiple cursors and select multiple occurrences or lines

**Option-Click: Insert an additional cursor.**

**Option-Command-Up or down arrow: Insert an additional cursor to the previous or next line.**

**Command-U: Undo a cursor-related operation.**

**Option-Shift-I: Insert a cursor at the end of each selected line.**

**Command-G or Shift-Command-G: Select the next or previous matched string.**

**Command-F2: Select the nearest character of each cursor.**

**Shift-Command-L: Select the nearest word of each cursor.**

**Option-Enter: Select all the matched strings.**

**Option-Shift-Drag: Multi-select lines**

**Option-Shift-Command-Up or down arrow: Extend a selection one line up or down.**

**Option-Shift-Command-Left or right arrow: Extend a selection one character to the left or right.**

## 2.4.20 Use EMR in DataWorks

Bind an EMR project and a DataWorks workspace



### Note:

**Before binding them, you need to obtain information about the EMR project.**

- 1. Log on to the DataWorks console.**
- 2. Click the Project Manage icon in the upper-right corner. The Project Management page appears.**

3. On the Project Management page, find the Compute Engine section. Click Add Compute Engine, and select Add EMR Cluster.
4. In the Add EMR Cluster dialog box that appears, set the relevant parameters.

| Parameter                | Description                                                                      |
|--------------------------|----------------------------------------------------------------------------------|
| Cluster Name             | The name of the EMR cluster. The value must be globally unique.                  |
| Access ID and Access Key | The AccessKey of the account that has been authorized to access the EMR cluster. |
| emrClusterID             | The ID of the EMR cluster. You can obtain the value from EMR.                    |
| emrUserId                | The user ID of the EMR cluster. You can obtain the value from EMR.               |
| emrProjectID             | The project ID of the EMR cluster. You can obtain the value from EMR.            |
| emrResource QueueName    | The name of a resource queue. You can obtain the value from EMR.                 |
| emrEndpoint              | The endpoint of the EMR cluster. You can obtain the value from EMR.              |

5. Click OK to complete the binding. Then, you can open the DataStudio page to create an EMR node.

**Note:**

If the binding fails, check whether the failure is caused by one of the following reasons:

- The EMR user ID has been bound to another tenant account.
- The specified cluster name has been used.

Create an EMR node

EMR nodes are categorized into four types: EMR\_HIVE, EMR\_SPARK\_SQL, EMR\_SPARK, and EMR\_MR4.

1. Log on to the DataWorks console.



**2. Create a workflow.**

- a. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
- b. In the Create Workflow dialog box that appears, set Workflow Name and Description.
- c. Click Create.

**3. Create an EMR node.**

- a. Expand the created workflow in the left-side navigation pane. Right-click Data Analytics, and choose Create Data Analytics Node > EMR HIVE.
- b. In the Create Node dialog box that appears, enter a name in Node Name and click Commit.

**4. Edit the code on the edit page.**

**5. Commit the node.**

After finishing the schedule configuration, click Save in the upper-left corner and commit the node to the development environment. After you commit the node, it is unlocked.

**6. Deploy the node.**

For more information, see [Publish nodes](#).

**7. Test the node in the production environment.**

For more information, see [Recurring tasks](#).

Reference resource files

EMR resource files are categorized into two resource types: EMR JAR and EMR File.

To reference EMR resource files, do as follows:

- For EMR\_HIVE and EMR\_MR resources: Place `--@resource_reference{"Resource name"}` at the first line.
- For EMR\_SPARK resources: Place `##@resource_reference {"Resource name "}` at the first line.

Manage data

DataWorks allows you to query EMR metadata and synchronize the data to the development environment for editing.

## 2.5 Administration

### 2.5.1 Overview

Generally, developers need to test workflows and nodes on the Operation Center page.

As a key tool for routine O&M, Operation Center enables you to manage and maintain the workflows and nodes that you have committed. The Operation Center service consists of four modules: Dashboard, Nodes, Node Instances, and Monitor.

- **Dashboard:** enables you to view and manage all global nodes of DataWorks. It displays various information, including Instances, Instances Run Today, Node Runtime, Instances Run in the Last Month, Nodes with Errors in the Last Month, and Node Types of the current workspace.
- **Nodes:** provides Recurring and Manually Triggered.
- **Node Instances:** provides Recurring, Manually Triggered, Smoke Test and Retroactive. You can manage them in a list view or DAG.
  - The list view displays the running status of nodes in a list. You can add multiple alerts at a time, change owners, and add nodes to baselines.
  - In the DAG, you can maintain and manage the running status of nodes and their dependencies on ancestor and descendant nodes. You can also perform operations, such as retroactive data generation and rerun, for a single node.
- **Monitor:** provides Baseline Instances, Baselines, Events, Alert Triggers, and Alerts.

### 2.5.2 Permissions

#### 2.5.2.1 Role permissions

The Administration service is only available to workspace administrators, developers, and administration experts.

The following table lists the permissions of workspace administrators, developers, and administration experts.

| Parent permission   | Permission                                                                | Workspace administrator | Developer | Administration expert |
|---------------------|---------------------------------------------------------------------------|-------------------------|-----------|-----------------------|
| Task administration | View the DAG of a recurring task                                          | Supported               | Supported | Supported             |
|                     | Jump to the DataStudio page to edit code for a recurring task             | Supported               | Supported | Not supported         |
|                     | View the DAG of an instance                                               | Supported               | Supported | Supported             |
|                     | View ancestor and descendant tasks in the DAG of a recurring task         | Supported               | Supported | Supported             |
|                     | View the list of tasks                                                    | Supported               | Supported | Supported             |
|                     | View operational logs of the business flow where a recurring task resides | Supported               | Supported | Supported             |
|                     | Perform smoke tests on a recurring task                                   | Supported               | Supported | Supported             |
|                     | Retroactively run a recurring task                                        | Supported               | Supported | Supported             |
|                     | Change the owner of a recurring task                                      | Supported               | Supported | Supported             |
|                     | View details of a recurring task                                          | Supported               | Supported | Supported             |

| Parent permission   | Permission                                                       | Workspace administrator | Developer     | Administration expert |
|---------------------|------------------------------------------------------------------|-------------------------|---------------|-----------------------|
|                     | View ancestor and descendant instances in the DAG of an instance | Supported               | Supported     | Supported             |
|                     | Pause an instance                                                | Supported               | Supported     | Supported             |
|                     | Restore an instance                                              | Supported               | Supported     | Supported             |
|                     | Terminate a single instance                                      | Supported               | Supported     | Supported             |
|                     | Terminate multiple instances at a time                           | Supported               | Not supported | Supported             |
|                     | View the list of instances                                       | Supported               | Supported     | Supported             |
|                     | View runtime logs                                                | Supported               | Supported     | Supported             |
|                     | Rerun a single instance                                          | Supported               | Supported     | Supported             |
|                     | Rerun multiple instances at a time                               | Supported               | Not supported | Supported             |
|                     | Search for an instance                                           | Supported               | Supported     | Supported             |
|                     | Set the status of an instance to Successful                      | Supported               | Supported     | Supported             |
| Node administration | Modify the baseline for a single recurring task                  | Supported               | Supported     | Supported             |

| Parent permission | Permission                                                       | Workspace administrator | Developer     | Administration expert |
|-------------------|------------------------------------------------------------------|-------------------------|---------------|-----------------------|
|                   | Modify the baseline for multiple recurring tasks at a time       | Supported               | Not supported | Supported             |
|                   | View the code of a recurring task                                | Supported               | Supported     | Supported             |
|                   | Change the owner of a single recurring task                      | Supported               | Supported     | Supported             |
|                   | Change the owner of multiple recurring tasks at a time           | Supported               | Not supported | Supported             |
|                   | Change the resource group for a single recurring task            | Supported               | Supported     | Supported             |
|                   | Change the resource group for multiple recurring tasks at a time | Supported               | Not supported | Supported             |
|                   | Perform smoke tests on a recurring task                          | Supported               | Supported     | Supported             |
|                   | Retroactively run a recurring task                               | Supported               | Supported     | Supported             |
|                   | Delete a dependency from an instance                             | Supported               | Not supported | Supported             |
|                   | Pause an instance                                                | Supported               | Supported     | Supported             |

| Parent permission       | Permission                                         | Workspace administrator | Developer     | Administration expert |
|-------------------------|----------------------------------------------------|-------------------------|---------------|-----------------------|
|                         | Restore an instance                                | Supported               | Supported     | Supported             |
|                         | Terminate a single instance                        | Supported               | Supported     | Supported             |
|                         | Terminate multiple instance at a time              | Supported               | Not supported | Supported             |
|                         | Modify the priority of an instance                 | Supported               | Supported     | Supported             |
|                         | Refresh the dependencies of an instance            | Supported               | Supported     | Supported             |
|                         | Rerun a single instance                            | Supported               | Supported     | Supported             |
|                         | Rerun multiple instances at a time                 | Supported               | Not supported | Supported             |
|                         | Set the status of an instance to Successful        | Supported               | Supported     | Supported             |
| Administration overview | View the number of overtime task instances         | Supported               | Supported     | Supported             |
|                         | Remove a record from the overview page             | Supported               | Supported     | Supported             |
|                         | View the sorting of tasks by errors within 30 days | Supported               | Supported     | Supported             |

| Parent permission | Permission                                    | Workspace administrator | Developer     | Administration expert |
|-------------------|-----------------------------------------------|-------------------------|---------------|-----------------------|
|                   | View the trend of task instances run each day | Supported               | Supported     | Supported             |
|                   | View the distribution of tasks by status      | Supported               | Supported     | Supported             |
|                   | View the trend of task instances run today    | Supported               | Supported     | Supported             |
|                   | View the sorting of tasks by duration         | Supported               | Supported     | Supported             |
|                   | View the distribution of tasks by type        | Supported               | Supported     | Supported             |
|                   | View the list of notification messages        | Supported               | Supported     | Supported             |
| Monitoring        | Disable a single alert                        | Supported               | Supported     | Supported             |
|                   | Disable multiple alerts at a time             | Supported               | Not supported | Supported             |
|                   | Enable or disable notification by phone       | Supported               | Supported     | Supported             |
|                   | Create custom notification rules              | Supported               | Supported     | Supported             |
|                   | Delete custom notification rules              | Supported               | Supported     | Supported             |

| Parent permission | Permission                       | Workspace administrator | Developer | Administration expert |
|-------------------|----------------------------------|-------------------------|-----------|-----------------------|
|                   | Edit custom notification rules   | Supported               | Supported | Supported             |
|                   | View custom notification rules   | Supported               | Supported | Supported             |
|                   | View the list of events          | Supported               | Supported | Supported             |
|                   | View details of an event         | Supported               | Supported | Supported             |
|                   | View details of a personal event | Supported               | Supported | Supported             |

## 2.5.2.2 Developers

Common scenarios

**Test run and manage the workflows submitted in the DataStudio service.**

**Perform operations only on workflow tasks and node tasks of workspaces in the development environment. Access to the production environment through authorization is not allowed.**

Permissions in the development environment

- **Test, pause, rerun, and perform retroactive executions on a workflow or node task.**
- **Modify the attributes of multiple workflows or node tasks. Terminate and rerun multiple tasks. Configure alerts.**

## 2.5.2.3 Deployment expert

**The Administration service is only available for developers, administration experts, and workspace administrators. The job of deployment experts is to publish nodes.**

## 2.5.2.4 Administration expert

Scenarios

- **Create tasks to publish nodes.**



- Handle task exceptions.

Perform administration for tasks in development and production environments after being authorized by an administrator.

For example, test or pause a business flow or node task, rerun a task, retroactively run a task, modify the attributes of multiple business flows or node tasks, terminate or rerun multiple tasks, and configure the alerts.

### 2.5.2.5 Workspace administrator

You are authorized to use the Administration service to manage the workspace.

## 2.5.3 O&M Overview

Log on to the DataWorks console and choose Operation Center > O&M Overview.

The tab that appears consists of Instances, Tasks Running, Durations, Errors in Last 30 Days, and other sections.

## 2.5.4 Task List

### 2.5.4.1 Recurring tasks

Ad-hoc tasks are automatically run as scheduled after being submitted to the scheduling system.

1. [Log on to the DataWorks console](#).
2. Choose Operation Center > Task List > Cycle Task.
3. Find the target task and click DAG.
4. The following table describes commands available on the list that appears if you right-click a task in the DAG.

| Command                                | Description                                                                                                          |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| Show Parent Nodes and Show Child Nodes | Show ancestor and descendant tasks. By default, a DAG displays only the current task and its parent and child tasks. |
| View Code                              | View the SQL or MapReduce code of a task.                                                                            |
| Edit Node                              | Jump to the DataStudio page and edit the code for the corresponding node.                                            |
| View Instances                         | View the instances created from a recurring task.                                                                    |
| View Lineage                           | View the lineage of a task.                                                                                          |
| Test                                   | Create a test instance to run a task.                                                                                |

| Command                      | Description                                                                                                                              |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Patch Data                   | Specify a date-based timestamp for the data to be processed, and create a retroactive task instance to process the specified data range. |
| Pause                        | Pause the execution of a task.                                                                                                           |
| Recover                      | Restore the execution of a task.                                                                                                         |
| Configure Quality Monitoring | Set data quality monitoring rules for a task.                                                                                            |

### 2.5.4.2 Ad-hoc tasks

Ad-hoc tasks are created from ad-hoc nodes that have been submitted to the scheduling system.



**Note:**

Ad-hoc tasks can only be manually run, and cannot be automatically run.

1. [Log on to the DataWorks console](#).
2. Choose Operation Center > Task List > Manual Task.
3. Find the target ad-hoc task, and click Run in the Actions column. Confirm the date-based timestamp of the data to be processed. Then, an ad-hoc task instance appears on the Manual Instance tab.

### 2.5.5 Task O&M

#### Recurring task instances

A recurring task instance is a snapshot of a task taken at a specified time.

DataWorks creates an instance every time a task is initiated. Task instances include recurring task instances, test instances, and retroactive task instances. You can manage task instances, such as terminating, rerunning, and restoring tasks.

DataWorks creates a task instance at 22:30 every day and determines whether the task instance will be run based on your schedule. Nodes and dependencies submitted after 22:30 takes effect on the next day.

#### Test instances

A test instance is created from a selected task and run as scheduled. Each test instance processes data with a timestamp of the previous day.

You can test an entire business flow with one test instance. That is, a test instance is a snapshot of all tasks in a business flow.

You can also test one or more selected tasks with one test instance.

**Note:**

Smoke tests can modify involved tables and files.



#### Ad-hoc task instances



DataWorks creates ad-hoc task instances from ad-hoc tasks. They do not have task dependencies, and must be initiated manually.

#### Retroactive task instances

If running recurring task instances fail, you can create retroactive task instances for substitution. This ensures the completion of business data. If you create a retroactive task instance from a recurring task instance, only the selected task instance is retroactively rerun, and its descendant task instances are not rerun.

The following table describes commands available on the list that appears if you right-click a task instance in a DAG.

| Command           | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Terminate         | <p>Terminate a task instance, and set its status to Failed.</p> <div> <b>Note:</b><br/>The status of the task instance must be Waiting for Scheduled Time, Waiting for Resources, or Running.</div>                                                                                                                           |
| Rerun             | <p>Rerun a task instance. After the task instance is successfully rerun, its descendant task instances will be run as scheduled. You can use this command if a task instance fails or it is not run as scheduled.</p> <div> <b>Note:</b><br/>The status of the task instance must be Not Running, Succeeded, or Failed.</div> |
| Rerun Child Nodes | Rerun descendant task instances of an instance.                                                                                                                                                                                                                                                                                                                                                                  |

| Command                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Set to Successful                      | <p>Set the status of a task instance to <b>Successful</b>, and runs its descendant task instances as scheduled. You can use this command if a task instance fails.</p> <div> <b>Note:</b><br/>The status of the task instance must be <b>Failed</b>.</div>                                                                                                                                                                 |
| Show Parent Nodes and Show Child Nodes | Show ancestor and descendant task instances. By default, a DAG displays only the current task instance and its parent and child task instances.                                                                                                                                                                                                                                                                                                                                                             |
| Pause                                  | <p>Pause the execution of a task instance in the current cycle.</p> <div> <b>Note:</b><br/>Task instances in subsequent cycles will be run as scheduled. If you need to pause the execution of a node in all cycles, navigate as follows: <b>Node &gt; Schedule &gt; Scheduling Mode &gt; Pause Scheduling</b>. Then, select <b>Pause Scheduling</b>, and save and publish the node to make the setting take effect.</div> |
| Recover                                | Restore the execution of a task instance, which is the opposite operation of <b>Pause</b> .                                                                                                                                                                                                                                                                                                                                                                                                                 |
| View Code                              | View the SQL or MapReduce code of a task instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Edit Node                              | Jump to the DataStudio page and edit the code for the corresponding node.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| View Nodes Affected                    | View the baseline to which the status of a task instance is related.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| More                                   | View the attributes, runtime log, and code of a task instance.                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## 2.5.6 Monitor

### 2.5.6.1 Overview

The Monitor module is a node monitoring and analysis system of DataWorks. Based on monitoring rules and node running status, the Monitor module determines whether, when, and how to trigger an alert, and whom an alert is sent to. It automatically selects the most appropriate alerting time, notification methods, and recipients.

The Monitor module provides you with the following benefits:

- Improves your efficiency on configuring monitoring rules.
- Prevents invalid alerts from bothering you.
- Automatically covers all important nodes for you.

General monitoring systems cannot meet the requirement of DataWorks. The reasons are as follows:

- DataWorks has numerous nodes, so it is difficult for you to find out the nodes to be monitored. Some DataWorks businesses have a large number of nodes, and dependencies between the nodes are complex. Even if you know the most important node, it is difficult to find all ancestor nodes of the node and monitor them all. In this case, if you simply monitor all nodes, a large number of invalid alerts may be generated. In consequence, you may miss those useful alerts.
- The alerting method varies with nodes. For example, some monitoring tasks require the relevant nodes to run for more than one hour before triggering alerts, while other monitoring tasks require the relevant nodes to run for more than two hours. It is extremely complex to set a monitoring node for each node, and it is difficult to predict the alert threshold value for each node.
- The alerting time varies with nodes. For example, an alert for an unimportant node can be sent after you start working in the morning. An alert for an important node needs to be sent immediately when an error occurs. General monitoring systems cannot tell the importance of each node.
- Different alerts require different operations to turn off.

The Monitor module provides comprehensive monitoring and alerting logic. You only need to provide the node name of your business. Then, the Monitor module automatically monitors the entire process of your node and creates standard alert triggers for the node. In addition, you can customize alerting triggers by completing basic settings.

Currently, the Monitor module has been used for monitoring all important businesses of Alibaba Group. Its full-path monitoring function guarantees the overall data output of all important businesses of Alibaba Group. In addition, it supports analyzing ancestor and descendant node paths to promptly detect risks and provide O&M advice for business departments. These functions of the Monitor module have guaranteed the long-term high stability of businesses in Alibaba Group.

## 2.5.6.2 Feature description

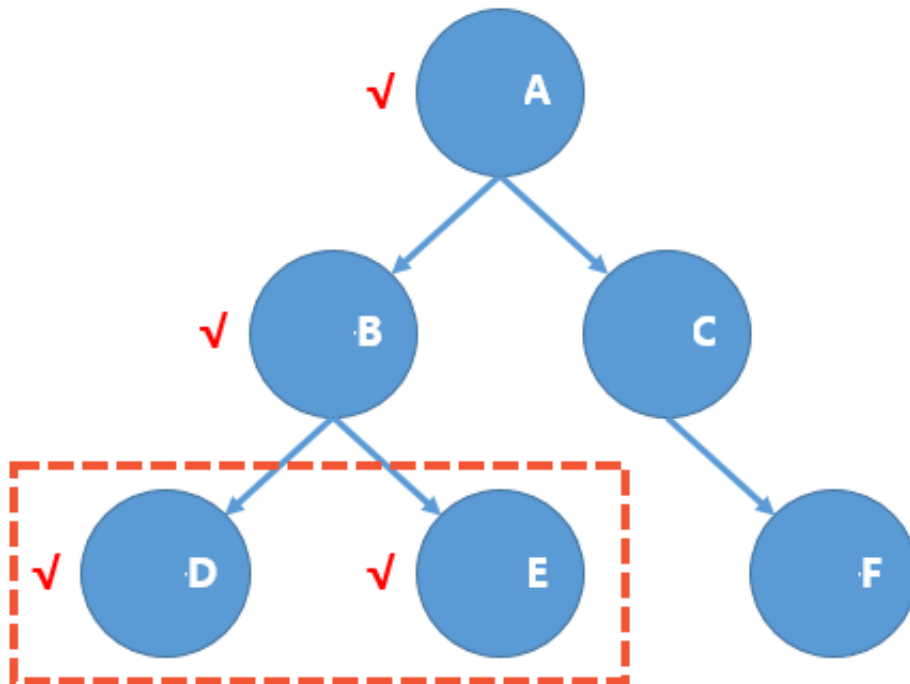
### 2.5.6.2.1 Baseline alert and event alert

This topic describes the functional logic of baseline alerts and event alerts from the aspects of monitoring scope, node capturing, alert object judgment, alerting time judgment, notification methods, and alert escalation.

#### Monitoring scope

A baseline is a management unit of a group of nodes, that is, a node group. You can specify nodes to monitor in a baseline.

After a baseline is monitored, all nodes of the baseline and its ancestor nodes are monitored. The Monitor module does not monitor all nodes by default. A node is monitored only when it has descendant nodes that are added to a monitoring baseline. If no descendant nodes are added to a monitoring baseline, the Monitor module does not report any alert even if the node fails.



As shown in the preceding figure, assume that DataWorks has only six nodes, and nodes D and E belong to a monitoring baseline. Nodes D and E and all their ancestor nodes are monitored by the Monitor module. That is, any error or slowdown on node A, B, D, or E will be detected by the Monitor module. However, nodes C and F are not monitored by the Monitor module.

## Node capturing

After the nodes to be monitored are specified, if a monitored node incurs an exception, the Monitor module generates an event. All alert decisions are based on the analysis of this event. Two types of node exceptions are available. You can choose Events > Event Type to view them.

- **Error:** indicates that a node fails to run.
- **Slow:** indicates that the running time of a node is significantly longer than the average running time of the node in the past periods.



### Note:

If a node times out and then encounters an error, two events are generated.

## Alerting time judgment

**Buffer**, an important concept in the Monitor module, refers to the maximum time period that a node can be delayed. The latest start time of a node is obtained by subtracting the average uptime from the baseline time.

The baseline time of baseline A is 05:00, you must set the latest start time of node E to 04:10. This time is calculated by subtracting the average uptime of node F (20 minutes) and node E (30 minutes) from the baseline time 05:00. This time is also the latest completion time of node B in baseline A.

To ensure that the baseline time of baseline B is 06:00, you must set the latest completion time of node B to 04:00. This time, which is earlier than 04:10, is calculated by subtracting the average uptime of node D (2 hours) from the baseline time 06:00. To meet the baseline time of both baseline A and baseline B, you must set the latest completion time of node B to 04:00.

The latest completion time of node A is 02:00, which is calculated by subtracting the average uptime of node B (2 hours) from 04:00. The latest start time of node A is 01:50, which is calculated by subtracting the average uptime of node A (10 minutes) from 02:00. If node A fails to run before 01:50, it is probable that baseline A is broken.

If node A fails to run at 01:00, its buffer is 50 minutes, which is the difference between 01:00 and 01:50. As demonstrated in this example, buffer reflects the degree of caution for a node exception.

## Baseline alert

**Baseline alerting is an additional feature developed for baselines that are enabled . Each baseline must provide an alert buffer and committed time. Baseline alerting is the action of notifying the preset alert recipient three times at the interval of 30 minutes when the baseline completion time estimated by the Monitor module exceeds the alert buffer.**

## Notification method

**Currently, baseline alerts are sent to the baseline owner by default. On the Alert Triggers page, you can find Global Baseline Alert Trigger, click View Details, and change the alert trigger method and the alerting action.**

## Gantt chart function

**The Gantt chart function reflects the key path of a node. The function is provided by the Baseline Instances module of Monitor.**



### Note:

**The key path is the slowest upstream link that causes the node to be completed at this time point.**

## 2.5.6.2.2 Custom alert trigger

**Alert trigger customization is a lightweight monitoring function of the Monitor module.**

**You can customize all monitoring alert triggers by setting the following parameters :**

- **Objects:** You can specify nodes, baselines, and workspaces as objects.
- **Trigger Condition:** Valid values include Completed, Uncompleted, Error, Uncompleted Cycle, and Overtime.
- **Notification Method:** Valid values include SMS and Email.
- **Maximum Alerts:** This parameter indicates the maximum number of alert reporting times. If the number of alerting times exceeds the preset threshold, no alerts are generated.
- **Minimum Alert Interval:** This parameter indicates the minimum time interval at which DataWorks reports alerts.



- **Quiet Hours:** This parameter indicates the specified period during which no alerts are reported.
- **Recipient:** This parameter indicates the person who receives alerts. You can set this parameter to the node owner or another recipient.

A monitoring rule uses the following five alert trigger conditions: Completed, Uncompleted, Error, Uncompleted Cycle, and Overtime.

- **Completed**

A completion alert can be set for nodes, baselines, and workspaces. Once all nodes of the preset objects are completed, the completion alert is reported. If you set a completion alert for a baseline, the alert is reported when all nodes of the baseline are completed.

- **Uncompleted**

You can set alerts for nodes, baselines, or workspaces that are not completed at a certain time point. For example, if you require that a baseline be completed at 10:00, an alert containing a list of uncompleted nodes is reported once a node in the baseline is not completed at the specified time.

- **Error**

An error alert can be set for nodes, baselines, and workspaces. Once a node has an error, an alert containing detailed node error information is sent to the recipient.

- **Uncompleted Cycle**

For the monitoring rules of hourly scheduled nodes, you can separately specify the uncompleted time points in different periods.

- **Overtime**

An overtime alert can be set for nodes, baselines, and workspaces. Once a monitored node of the preset object is not completed within the specified time, an alert is reported.

## 2.5.6.3 Instructions

### 2.5.6.3.1 Baseline instances

On the Baseline Instances page, you can view the relevant information about a baseline.

After creating a baseline, you need to enable the baseline so that baseline instances can be generated. On the Baseline Instances page, you can search for baseline instances by business date, owner, event ID, workspace, or baseline name. You can also click View Details, Handle, and View Gantt Chart in the Actions column as required.

A baseline can be in any of the following four statuses:

- **Normal:** indicates that all nodes in the baseline are completed before the alerting time.
- **Alerting:** indicates that one or more nodes in the baseline are not completed after the alerting time but the committed time has not arrived.
- **Overtime:** indicates that one or more nodes in the baseline are still not completed when the committed time expires.
- **Others:** indicates that all nodes in the baseline are paused or the baseline is not associated with any node.

You can click View Details, Handle, and View Gantt Chart in the Actions column as required.

- **View Details:** Click this button to open the Baseline Instance Details page and view the details of a baseline instance.

On the Baseline Instance Details page, you can view the Basic Info, Critical Path, Baseline Instance Info, Historical Completion Curve, and Related Events.



**Note:**

The business date is one day before the system date. You need to specify periods only for hourly baselines.

- **Handle:** The baseline that reports an alert stops alerting while the alert is being handled.
- **View Gantt Chart:** Click this button to view the critical paths of nodes.

### 2.5.6.3.2 Events

On the Events page, you can view all events related to slowdown or errors.

On the Events page, you can search for events by conditions, such as Owner, Detection Time, Event Status, Event Type, and the name or ID of a node or node instance.

Each row in the search results represents an event (that is, each row is associated with an abnormal node). The worst baseline is the baseline with the minimum buffer among the baselines affected by the event.

- Click View Details in the Actions column of the relevant event. You can view the event occurrence time, alerting time, clearance time, historical runtime logs of the node, and detailed node logs.

You can assign an alert recipient. After you click Alert Information, the alert details page corresponding to the event appears. The baseline impact shows all descendant baselines affected by the node corresponding to an event. You can determine the specific cause of the event by observing the corresponding descendant baselines, baseline break status, and node logs.

- After you click Handle, the handle operation on the event is recorded and the event is not reported during the operation.
- After you click Ignore, the ignore operation on the event is recorded and the event is not reported permanently.

### 2.5.6.3.3 Alert triggers

This topic describes how to customize alert triggers on the Alert Triggers page.

1. In the left-side navigation bar, choose Monitor > Alert Triggers. The Alert Triggers page appears.
2. Click Create Custom Trigger in the upper-right corner.
3. In the Create Custom Trigger dialog box that appears, set the relevant parameters.

| Parameter    | Description                                                             |
|--------------|-------------------------------------------------------------------------|
| Trigger Name | The name of the new custom trigger.                                     |
| Object Type  | The granularity of monitoring objects. Valid values: Node and Workflow. |

| Parameter              | Description                                                                                                                         |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Objects                | The monitored object. Enter the name or ID of a node or workflow, and click the icon on the right to add the object.                |
| Trigger Condition      | The conditions for triggering alerts. Valid values: Completed, Uncompleted, Error, Uncompleted Cycle, and Overtime.                 |
| Maximum Alerts         | The maximum number of alert reporting times. If the number of alerting times exceeds the preset threshold, no alerts are generated. |
| Minimum Alert Interval | The minimum time interval at which DataWorks reports alerts.                                                                        |
| Quiet Hours            | The specified period during which no alerts are reported.                                                                           |
| Notification Method    | The method of reporting alerts. Valid values: Email and SMS.                                                                        |
| Recipient              | The person who receives alerts. You can set this parameter to the node owner or another recipient.                                  |
| DingTalk Chatbot       | The DingTalk chatbot to receive alerts.                                                                                             |

4. Click OK to create the trigger.

On the Alert Triggers page, click View Details next to a trigger to view the specific content of the trigger.

### 2.5.6.3.4 Alerts

You can view all alerts in the Monitor module.

Choose Monitor > Alerts. On the page that appears, you can search for alerts by Trigger ID/Name, Recipient, Alert Time, Notification Method, or Trigger Type.

You can also view alert information such as Notification Method and Status. Click View Details in the Actions column to view more information about each alert.

### 2.5.6.4 FAQ related to the Monitor module

#### 2.5.6.4.1 Why was my alert reported to someone else?

- For custom alerts, confirm the alert triggers with their creators.

- For alerts generated for a baseline after the baseline is enabled, you can view the specific event page. The reason for alert assignment is presented at the bottom of the event page.

#### 2.5.6.4.2 What can I do if I do not want to receive alerts for unimportant nodes?

Click View Details on the Events page to view the descendant baselines affected by the node. If any problem occurs within the scope of these baselines, alerts may be triggered. In this case, contact the corresponding baseline owner.

#### 2.5.6.4.3 Why is no alert reported for a baseline break?

Baseline monitoring is controlled by the baseline switch and enabled for nodes. If all nodes are running normally, no alert will be triggered even in case of a baseline break. This is because all the nodes are running normally and DataWorks cannot determine which node has an error. Baseline break is a baseline status, indicating that a node is still not completed after the committed time.

The reasons why the baseline is still broken when all nodes are normal are as follows:

- The baseline time is set improperly.
- The node dependency is incorrect.

#### 2.5.6.4.4 Can I disable DataWorks from reporting an alert for a node that slows down?

DataWorks reports a node slowdown alert only when a node meets both the following conditions:

- The node is an ancestor node of an important baseline.
- Compared with its historical performance, the node does become slowdown.

If the node slowdown has a minor impact, you can ignore the alert. Confirm the impact with the party whose monitoring baseline uses your node as an ancestor node. You can go to the Events page to view the descendant baseline information. If you are responsible for the party, maintain the node properly.

#### 2.5.6.4.5 Why did I fail to receive an alert for an error node?

DataWorks reports an alert only for specified nodes when an error occurs. An alert is reported for an error node only when the node meets any of the following conditions:

- The node is an ancestor node of a baseline that has been enabled.
- An alert trigger has been customized.

#### 2.5.6.4.6 What can I do if I receive an alert at night?

If you receive an alert at night, go to the event page to disable the event alert. However, the alert can only be disabled for a period of time. After receiving an alert, you need to handle it in a timely manner.

## 2.6 Organization management

### 2.6.1 Project management

#### 2.6.1.1 Description

On the Project Management page, you can create, change, enable, and disable workspaces.

#### 2.6.1.2 Create a workspace

This topic describes how to create a workspace in the Project Management service.

##### Prerequisites

Before creating a workspace, you must create a compute engine to initialize the MaxCompute project, and then bind the compute engine in the Create Workspace dialog box.

##### Overview

DataWorks provides various preset templates for a workspace administrator to select when creating one or more workspaces in working environments, including development, test, staging, and production. It can also automatically generate associations between workspaces. A one-to-multiple relationship exists between departments and workspaces. That is, multiple workspaces can be created under one department.

You can create a workspace in either of the following modes:

- **Standard Mode (Development and Production Environments):** One DataWorks workspace corresponds to one MaxCompute project in the development environment and one MaxCompute project in the production environment, respectively.
- **Basic Mode (Production Environment Only):** One DataWorks workspace corresponds to only one MaxCompute project.



**Note:**

For more information about the two workspace modes, see [Workspace mode overview](#).

#### Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move your pointer over the DataWorks icon in the upper-left corner, and click **Project Management**.
3. On the Workspaces page that appears, click **Create Workspace** in the upper-right corner.
4. Set parameters in the Create Workspace dialog box, and select the corresponding MaxCompute project.



**Note:**

If the standard mode is selected, you must select two MaxCompute projects to map the workspace.

5. Set parameters in the Advanced Settings section. You can select whether to enable periodic scheduling and whether to allow downloading SELECT query results. You also need to associate the workspace with MaxCompute projects.
6. Click **OK**.

## 2.6.2 Member management

1. [Log on to the DataWorks console](#) as a workspace administrator.
2. Move your pointer over the DataWorks icon in the upper-left corner, and click **Project Management**.
3. Click **Member Management** in the left-side navigation bar. The **Members** page appears.

4. You can enter a member name or logon name in the search box to search for a member to be added or removed from the current organization.

- **Assign a role**

To assign a role to a member, click the Roles drop-down list next to the member, and select the role to be assigned.

To unassign a role from a member, click x next to the role.

- **Remove a member from an organization**

Click Delete next to the member, and click OK in the Remove from Tenant dialog box that appears.

## 2.6.3 Resource groups

### 2.6.3.1 About scheduling resources

You can use the Scheduling Resource page to create, configure, and edit a scheduling resource.

A scheduling resource is an object within an organization. A dedicated scheduling resource may contain multiple physical machines or ECS instances that are used to implement a specific task.

1. Log on to the [DataWorks](#) console.
2. In the left-side navigation pane, choose Organization Management > Scheduling Resources.



**Note:**

On the Scheduling Resources page, the tenant administrator can create a dedicated scheduling resource, and edit an existing scheduling resource.

### 2.6.3.2 Create a scheduling resource

A tenant administrator can create a dedicated scheduling resource and allocate the resource to a workspace to implement a specific task, which may be a data synchronization task, shell script, MaxCompute SQL script, MaxCompute MapReduce, or machine learning. When no dedicated scheduling resource is specified, all tasks within a workspace are implemented by using the resources of the cluster that hosts the system.



## Example

A scheduling resource can be created to address the following issues:

- Many tasks of the current workspace are waiting for resources, and the number of tasks has reached a threshold. Existing gateways can no longer meet service requirements, and more gateways are required.
- Some special tasks, such as shell scripts of the workspace must be implemented on a specific server. You must apply for a separate gateway to implement this task.

To meet the requirements of data development, a tenant administrator needs to create a dedicated scheduling resource and allocate it to the workspace to implement a specific data synchronization task.

## Procedure

Follow these steps to create a scheduling resource:

1. Log on to the [DataWorks](#) console as a tenant administrator.
2. In the left-side navigation pane, choose **Organization Management** > **Scheduling Resources**.
3. Click **Add Scheduling Resource** in the upper right corner.
4. In the **Add Scheduling Resource** dialog box, specify the required fields.

| Field         | Description                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource Name | The resource name. The specified name can contain Chinese characters, letters, underscores, and digits. It can be up to 60 characters in length. |
| Workspace     | The workspace to which the resource group allocates.                                                                                             |

5. Click **OK**.

### 2.6.3.3 Change the workspace of scheduling resources

You can change the workspace of dedicated scheduling resources that have been created and configured.

## Procedure

To change the workspace of dedicated scheduling resources, the tenant administrator performs the following operations:

1. Log on to the [DataWorks](#) console as a tenant administrator.

2. Choose Organization Management > Scheduling Resources.
3. On the page that appears, enter a scheduling resource name for a fuzzy search to find the target scheduling resource.
4. Click Change Workspace.
5. Click OK.

### 2.6.3.4 Manage servers

After successfully creating a scheduling resource, the tenant administrator must bind the resource to servers to complete resource configuration. A dedicated scheduling resource can be bound to a maximum of 1,000 servers. The servers must be selected from the ECS instances of the tenant. One ECS instance can be bound to only one dedicated scheduling resource.

#### Example

**Scenario:** The tenant administrator has added a dedicated scheduling resource for the data synchronization task. However, the dedicated scheduling resource has not been bound to a server ([Create a scheduling resource](#)). Now the administrator needs to select and add a server from the ECS instances.

Existing scheduling resources cannot meet the production needs during the daily scheduling process. As a result, many data synchronization tasks are waiting for resources. Now the tenant administrator needs to add servers for the dedicated scheduling resource that is used for the data synchronization task.

#### Procedure

1. Click Server Management.



**Note:**

You can find name of the server from your instance.

2. Click Add Server, and specify the required fields in the dialog box that appears.



**Note:**

You can find name of the server from your instance.

3. Click Initialization and perform operations as prompted.

## 2.6.4 Compute engine

### 2.6.4.1 Configure the compute engine

Currently, DataWorks only supports MaxCompute as its compute engine. All business flows and nodes in a workspace are run on the MaxCompute project associated to the workspace.

Example



**Note:**

Tenant administrators can modify the settings for MaxCompute projects. The following settings are changeable: the project description, whether to use the MaxCompute project owner account to run MaxCompute jobs, the account used for running MaxCompute jobs, and the AccessKey of the account.

Assume that the account used for running MaxCompute jobs is no longer available, for example, because the account owner has resigned. If Run MaxCompute Task Using MaxCompute Owner Account is not selected, the tenant administrator needs to immediately modify the account used for running MaxCompute jobs and its AccessKey so that tasks can properly run in the workspace that uses the corresponding MaxCompute project.

Procedure

You can modify the account used for running MaxCompute jobs and its AccessKey as follows:

1. Log on to the DataWorks console as a tenant administrator. For more information, see [Log on to the DataWorks console](#).
2. Choose Project Management > Compute Engine.
3. In the search box on the Project Management > Compute Engine page, enter the compute engine name. Fuzzy search is supported.
4. Find the target compute engine, and click Configure in the Actions column.
5. In the Configure Compute Engine dialog box, specify the Alibaba Cloud Account and the AccessKey.



**Note:**

You can also select Run MaxCompute Task Using MaxCompute Owner Account or create a new Alibaba Cloud account.

---

**6. Click Submit.**

## 2.7 Project Management

### 2.7.1 Configure a workspace

#### 2.7.1.1 Basic property settings

You can configure basic properties by specifying the following fields as required: **Workspace ID, Created At, Workspace Name, Mode, Display Name, Allow SELECT Result Download, and Enable Periodic Scheduling.**

#### Example

To meet workspace requirements, a workspace administrator may need to check the basic properties of a workspace and modify the default scheduling resource, workspace description, and release target.

#### Procedure

To modify the default scheduling resource, workspace description, and release target, follow these steps:

1. Log on to the DataWorks console as an administrator.
2. In the left-side navigation pane, choose **Workspace Management > Settings**.
3. On the Settings page, you can view the workspace name, owner, and mode.

| Field                        | Description                                                                                                                                                                                                       |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Display name                 | The name of the workspace that is displayed on the page.                                                                                                                                                          |
| Description                  | The description of the workspace.                                                                                                                                                                                 |
| Allow SELECT Result Download | Specifies whether to allow you to download the result that is returned by the SELECT statement. After enabling this feature, you can download the running result on the SQL page to your on-premises environment. |
| Enable Periodic Scheduling   | Specifies whether to enable periodic scheduling. After enabling this feature, the task can be submitted to the scheduling system for running.                                                                     |


### 2.7.1.2 Compute engine

- If your DataWorks workspace is in standard mode, you can view the name of the MaxCompute production project and the name of the MaxCompute development project for the current workspace.
- If your DataWorks workspace is in basic mode, you can view the only one MaxCompute project name for the workspace.

### 2.7.2 Member management

You can use the Workspace Management page in the administration console to manage the workspace members.

In the left-side navigation pane, choose **Workspace Management > Member Management**.

| Field                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Member                      | The username that is used to log on to the Alibaba Cloud console.                                                                                                                                                                                                                                                                                                                                                                 |
| Apsara Stack Tenant Account | The Apsara Stack tenant account of the user.                                                                                                                                                                                                                                                                                                                                                                                      |
| Role                        | <p>The role of the member in the workspace: administrator, developer, administration expert, deployment expert, visitor, and security expert.</p> <div> <b>Note:</b><br/>A project administrator can assign roles to RAM users as required. To view more information about the permissions of different roles, click <b>Permissions</b>.</div> |
| Joined At                   | The time when the member joined the workspace.                                                                                                                                                                                                                                                                                                                                                                                    |
| Actions                     | The allowed actions by the current member. Currently , only one action is available: removing a member from the workspace. Only a workspace administrator can perform this action.                                                                                                                                                                                                                                                |

To add a member to the workspace, click **Add Member** in the upper-right corner. The system automatically shows all the RAM users under the Apsara Stack tenant account, and allows you to search for and filter RAM users. You can select one or more matched RAM users and batch assign roles for the users. Then, you can add

selected users to the workspace. These users can perform operations on the data and features of the current workspace.

### 2.7.3 Permission management

DataWorks provides seven roles: workspace owner (not authorizable), workspace administrator, developer, O&M engineer, deployment engineer, guest, and security administrator. This section describes the permissions of these roles.

#### Data management

| Permission                                              | Workspace owner | Workspace administrator | Developer | O&M engineer | Deployment engineer | Guest | Security administrator |
|---------------------------------------------------------|-----------------|-------------------------|-----------|--------------|---------------------|-------|------------------------|
| Delete a self-created table                             | √               | √                       | √         | None         | None                | None  | None                   |
| Specify a category for a self-created table             | √               | √                       | √         | None         | None                | None  | None                   |
| View favorite tables                                    | √               | √                       | √         | None         | None                | None  | None                   |
| Create a table                                          | √               | √                       | √         | None         | None                | None  | None                   |
| Unhide a self-created table                             | √               | √                       | √         | None         | None                | None  | None                   |
| Modify the schema of a self-created table               | √               | √                       | √         | None         | None                | None  | None                   |
| View self-created tables                                | √               | √                       | √         | None         | None                | None  | None                   |
| View the content of a self-submitted request            | √               | √                       | √         | None         | None                | None  | None                   |
| Hide a self-created table                               | √               | √                       | √         | None         | None                | None  | None                   |
| Specify the time-to-live for a self-created table       | √               | √                       | √         | None         | None                | None  | None                   |
| Request permissions for a table created by another user | √               | √                       | √         | None         | None                | None  | None                   |
| Delete a table                                          | None            | √                       | √         | None         | None                | None  | None                   |

| Permission                                | Workspace owner | Workspace administrator | Developer | O&M engine | Deployment engine | Guest | Security administrator |
|-------------------------------------------|-----------------|-------------------------|-----------|------------|-------------------|-------|------------------------|
| Update a table                            | None            | √                       | √         | None       | None              | None  | None                   |
| Preview data                              | √               | √                       | √         | √          | √                 | √     | √                      |
| Preview table data of other organizations | √               | √                       | None      | None       | None              | None  | None                   |

#### Deployment management

| Permission                           | Workspace owner | Workspace administrator | Developer | O&M engine | Deployment engine | Guest | Security administrator |
|--------------------------------------|-----------------|-------------------------|-----------|------------|-------------------|-------|------------------------|
| Create deployment tasks              | √               | √                       | √         | √          | None              | None  | None                   |
| View the list of deployment tasks    | √               | √                       | √         | √          | √                 | √     | None                   |
| Delete deployment tasks              | √               | √                       | √         | √          | None              | None  | None                   |
| Deploy                               | √               | √                       | None      | √          | √                 | None  | None                   |
| View the content of deployment tasks | √               | √                       | √         | √          | √                 | √     | None                   |

#### Button control

| Permission      | Workspace owner | Workspace administrator | Developer | O&M engine | Deployment engine | Guest | Security administrator |
|-----------------|-----------------|-------------------------|-----------|------------|-------------------|-------|------------------------|
| Button: Stop    | √               | √                       | √         | None       | None              | None  | None                   |
| Button: Format  | √               | √                       | √         | None       | None              | None  | None                   |
| Button: Edit    | √               | √                       | √         | None       | None              | None  | None                   |
| Button: Run     | √               | √                       | √         | None       | None              | None  | None                   |
| Button: Zoom In | √               | √                       | √         | None       | None              | None  | None                   |

| Permission        | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|-------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Button: Save      | √               | √                       | √         | None       | None          | None  | None                   |
| Button: Show/Hide | √               | √                       | √         | None       | None          | None  | None                   |
| Button: Delete    | √               | √                       | √         | None       | None          | None  | None                   |

#### Code development

| Permission             | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Save and commit nodes  | √               | √                       | √         | None       | None          | None  | None                   |
| View the code of nodes | √               | √                       | √         | √          | √             | √     | None                   |
| Create nodes           | √               | √                       | √         | None       | None          | None  | None                   |
| Delete nodes           | √               | √                       | √         | None       | None          | None  | None                   |
| View the node list     | √               | √                       | √         | √          | √             | √     | None                   |
| Run nodes              | √               | √                       | √         | None       | None          | None  | None                   |
| Edit the code of nodes | √               | √                       | √         | None       | None          | None  | None                   |
| Download files         | √               | √                       | √         | None       | None          | None  | None                   |
| Upload local files     | √               | √                       | √         | None       | None          | None  | None                   |

#### Function development

| Permission            | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|-----------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| View function details | √               | √                       | √         | √          | √             | √     | None                   |
| Create functions      | √               | √                       | √         | None       | None          | None  | None                   |



| Permission       | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Query functions  | √               | √                       | √         | √          | √             | √     | None                   |
| Delete functions | √               | √                       | √         | None       | None          | None  | None                   |

#### Node types

| Permission                   | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|------------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Node type: Machine Learning  | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: ODPS MR           | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: Data Sync         | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: ODPS SQL          | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: XLIB              | √               | √                       | √         | √          | √             | √     | None                   |
| Node type: Shell             | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: Zero-Load Node    | √               | √                       | √         | √          | √             | √     | None                   |
| Node type: dtboost_recommand | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: dtboost_analytic  | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: PyODPS            | √               | √                       | √         | None       | None          | None  | None                   |
| Node type: script_seahawks   | √               | √                       | √         | None       | None          | None  | None                   |

## Resource management

| Permission                             | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|----------------------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| View the resource list                 | √               | √                       | √         | √          | √             | √     | None                   |
| Delete resources                       | √               | √                       | √         | None       | None          | None  | None                   |
| Create resources                       | √               | √                       | √         | None       | None          | None  | None                   |
| Upload Python files                    | √               | √                       | √         | None       | None          | None  | None                   |
| Upload JAR files                       | √               | √                       | √         | None       | None          | None  | None                   |
| Upload TXT files                       | √               | √                       | √         | None       | None          | None  | None                   |
| Upload files as archive-type resources | √               | √                       | √         | None       | None          | None  | None                   |

## Workflow development

| Permission                | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|---------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Run or stop workflows     | √               | √                       | √         | None       | None          | None  | None                   |
| Save workflows            | √               | √                       | √         | None       | None          | None  | None                   |
| View workflows            | √               | √                       | √         | √          | √             | √     | None                   |
| Commit the code of nodes  | √               | √                       | √         | None       | None          | None  | None                   |
| Modify workflows          | √               | √                       | √         | None       | None          | None  | None                   |
| View the workflow list    | √               | √                       | √         | √          | √             | √     | None                   |
| Change the workflow owner | √               | √                       | None      | None       | None          | None  | None                   |
| Open the code of nodes    | √               | √                       | √         | None       | None          | None  | None                   |
| Delete workflows          | √               | √                       | √         | None       | None          | None  | None                   |

| Permission              | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|-------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Create workflows        | √               | √                       | √         | None       | None          | None  | None                   |
| Migrate database tables | √               | √                       | √         | None       | None          | None  | None                   |
| Create folders          | √               | √                       | √         | None       | None          | None  | None                   |
| Delete folders          | √               | √                       | √         | None       | None          | None  | None                   |
| Modify folders          | √               | √                       | √         | None       | None          | None  | None                   |
| Export workflows        | None            | √                       | √         | √          | None          | None  | None                   |

#### Data integration

| Permission                           | Workspace owner | Workspace administrator | Developer | O&M engine | Deploy engine | Guest | Security administrator |
|--------------------------------------|-----------------|-------------------------|-----------|------------|---------------|-------|------------------------|
| Resource consumption monitoring menu | √               | √                       | None      | None       | None          | None  | None                   |
| Data Integration: edit nodes         | √               | √                       | √         | None       | None          | None  | None                   |
| Data Integration: view nodes         | √               | √                       | √         | None       | None          | None  | None                   |
| Resource consumption monitoring      | √               | √                       | None      | None       | None          | None  | None                   |
| Data Integration: delete nodes       | √               | √                       | √         | None       | None          | None  | None                   |
| Migrate database tables              | √               | √                       | None      | None       | None          | None  | None                   |

## 2.7.4 MaxCompute management

### 2.7.4.1 Basic settings

In the project management module of the Alibaba Cloud DTplus platform, you can manage and configure MaxCompute properties for the current project.

You can configure commonly used permissions for MaxCompute in basic settings.

1. Log on to the [DataWorks](#) console.
2. Choose Workspace Settings > MaxCompute Management.
3. On the Basic Settings tab, specify MaxCompute settings.

The following table lists the available permissions.

| Permission                                                | Description                                                                                                                                                                                                           |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enable ACL-based authorization                            | You can enable or disable the option. It is equivalent to the following operation: An owner account sets CheckPermissionUsingACL to true or false in a MaxCompute project. The option is enabled by default.          |
| Allow object creators to access objects                   | You can enable or disable the option. It is equivalent to the following operation: An owner account sets ObjectCreatorHasAccessPermission to true or false in a MaxCompute project. The option is enabled by default. |
| Allow object creators to grant object-related permissions | You can enable or disable the option. It is equivalent to the following operation: An owner account sets ObjectCreatorHasGrantPermission to true or false in a MaxCompute project. The option is enabled by default.  |
| Protect workspace data                                    | You can enable or disable the option. It is equivalent to the following operation: An owner account sets ProjectProtection to true or false in a MaxCompute project. The option is disabled by default.               |
| RAM user service                                          | You can enable or disable the option to allow or disallow RAM users to access the MaxCompute project. The option is enabled by default.                                                                               |
| Enable policy-based authorization                         | You can enable or disable the option. It is equivalent to the following operation: An owner account sets CheckPermissionUsingPolicy to true or false in a MaxCompute project. The option is enabled by default.       |

| Permission                         | Description                                                                                                                                                                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enable column-level access control | You can enable or disable the option. It is equivalent to the following operation: An owner account sets LabelSecurity to true or false in a MaxCompute project . The option is disabled by default, and an owner can enable it if necessary. |

### 2.7.4.2 Customize user roles

You can go to the MaxCompute Management page to manage and configure the roles of the current workspace on the Custom User Roles tab.

1. Log on to the DataWorks console.
2. Click the Project Manage icon in the upper-right corner. The Project Management page appears.
3. In the left-side navigation bar, click MaxCompute Management, and then click Custom User Roles.

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Role Name   | A role name in a MaxCompute project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Actions     | <ul style="list-style-type: none"><li>• <b>View Details:</b> Click this button to view the list of members that are assigned the current role and the permissions of the role on tables and projects.</li><li>• <b>Members:</b> Click this button to add a member to or remove a member from the current role.</li><li>• <b>Authorizations:</b> Click this button to manage the permissions of the current role on tables or projects.</li><li>• <b>Delete:</b> Click this button to delete the current role.</li></ul> |
| Create Role | Click Create Role in the upper-right corner. In the Create Role dialog box that appears, enter a name in Role Name. Select the member account to be added, click > to move it to the list of added accounts, and then click OK.                                                                                                                                                                                                                                                                                         |

## 2.8 Data Integration

### 2.8.1 Data Integration

#### 2.8.1.1 Overview

Data Integration is a stable and efficient data synchronization service provided by Alibaba Group. You can add data stores to and remove them from DataWorks by

using this service. Data Integration is designed to implement fast and stable data transmission and synchronization between various heterogeneous data stores in complex networks.

#### Batch data synchronization

The Data Integration service facilitates data transmission between diverse structured and semi-structured data stores. It provides readers and writers for the supported data stores and defines a channel based on the data stores and datasets of a reader and a writer. This service applies a simplified data type system.

#### Supported data store types

Data Integration provides extensive options for data stores listed as follows:

- Text storage, such as File Transfer Protocol (FTP) and SSH File Transfer Protocol (SFTP) servers, Object Storage Service (OSS), and multimedia files
- Relational databases, such as Relational Database Service (RDS) , MySQL, and PostgreSQL
- NoSQL databases, such as Memcache, Redis, MongoDB, and HBase
- Big data products, such as MaxCompute, and Hadoop Distributed File System (HDFS)
- Massively parallel processor (MPP) databases

For more information, see [Supported data sources](#).



#### Note:

The parameter settings vary with data stores. Ensure that settings of data stores and data synchronization nodes are consistent with the site conditions.

#### Synchronization node configuration modes

You can configure data synchronization nodes by using the codeless UI or code editor.

- **Codeless UI:** enables you to configure data synchronization nodes by using the codeless UI. This mode provides step-by-step instructions to help you quickly complete the configuration of a data synchronization node. This mode is easy to use but provides only limited features.
- **Code editor:** allows you to write a JSON script for each data synchronization node. The code editor provides advanced features to facilitate flexible

configuration. This mode is suitable for experienced users and increases the cost of learning.

**Note:**

- The code generated for a data configuration node on the codeless UI can be converted to a script. This conversion is irreversible. After the conversion is complete, the configuration node cannot be switched back.
- You must configure data stores and create a destination table before editing the code of a data synchronization node.

### Network types

A data store can be located in the classic network, a VPC, or a user-created data center network.

| Network type                     | Description                                                                                                                                                                                                                                                                                      |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Classic network                  | A network deployed by Alibaba Cloud, which is shared with other tenants. Networks of this type are easy to use.                                                                                                                                                                                  |
| VPC                              | A Virtual Private Cloud (VPC) network created on Apsara Stack, which is isolated to only one Apsara Stack tenant account. You have full control over your VPC, including customizing the IP address range, dividing your VPC into multiple subnets, and configuring routing tables and gateways. |
| User-created data center network | A data center network deployed by yourself, which can be connected to DataWorks.                                                                                                                                                                                                                 |

**Note:**

- Specify the network type as classic network for public network connections. Note the public network bandwidth and other relevant billing. We recommend that you do not use public network connections.
- If you need to configure a data synchronization node on an on-premises data center network, use the code editor and run the node on a local resource group. You can also configure the node by specifying the reader and writer in a shell node.
- VPC is an isolated network environment, which allows you to customize the IP address range, subnets, and gateways. With its continuous security

enhancement, VPC has become more widely used. In this context, Data Integration provides RDS-MySQL, and RDS-PostgreSQL. You do not need to create an ECS instance in a VPC. Instead, DataWorks automatically detects an ECS instance through a reverse proxy to provide network connectivity.

Data Integration also supports other Apsara Stack and Alibaba Cloud databases, such as PPAS, OceanBase, Redis, MongoDB, Memcache, Table Store, and HBase. To establish a connection to a non-RDS data store in a VPC for configuring data synchronization nodes, you need to create an ECS instance in the VPC.

#### Constraints and limits

- Data Integration supports only the synchronization of structured, semi-structured, and unstructured data. Structured databases include RDS. Unstructured data, such as OSS objects and text files, must be capable of being converted to structured data. Data Integration allows you to synchronize logical two-dimensional tables that are converted from source data. However, it cannot synchronize other unstructured data, such as MP3 files stored in OSS, to MaxCompute.
- Data Integration supports data synchronization and exchange in one region or between multiple regions.

In some regions, data can be transmitted over the classic network, but this cannot be guaranteed. We recommend that you use a public network connection if the classic network is required but cannot be connected.

- Data Integration supports only data synchronization but not data stream consumption.

### 2.8.1.2 Basic concepts

#### DMU

A data migration unit (DMU) represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.

#### Concurrency

You can specify a maximum number of concurrent threads to read and write data to data storage within a single data synchronization task.



## Bandwidth throttling

If bandwidth throttling is enabled, you need to specify a maximum transmission rate.

## Dirty data

Dirty data indicates meaningless data and data that does not match the specified data type. For example, data is dirty if its type in the source table is VARCHAR and it is to be written to an INT-type field of the destination table. The writing will fail due to the infeasible type conversion.

## Data source

A data source in DataWorks is a database or a data warehouse. DataWorks supports various data sources, and supports data synchronization between data sources of different types.

## 2.8.2 Data sources

### 2.8.2.1 Supported data sources

Data Integration is a stable, efficient, and scalable data synchronization platform provided by Alibaba Cloud. It supports transmission of data in batches for Alibaba Cloud services, such as MaxCompute, and OSS.

The following table lists data source types supported by Data Integration.

| Data source category | Data source type | Reader    | Writer    | Configuration methods | Hosted on Apsara Stack or on the premises |
|----------------------|------------------|-----------|-----------|-----------------------|-------------------------------------------|
| Relational database  | MySQL            | Supported | Supported | Wizard or script      | Hosted on Apsara Stack or on the premises |
| Relational database  | PostgreSQL       | Supported | Supported | Wizard or script      | Hosted on Apsara Stack or on the premises |

| <b>Data source category</b>      | <b>Data source type</b>                    | <b>Reader</b>        | <b>Writer</b>    | <b>Configuration methods</b> | <b>Hosted on Apsara Stack or on the premises</b> |
|----------------------------------|--------------------------------------------|----------------------|------------------|------------------------------|--------------------------------------------------|
| <b>Relational database</b>       | <b>Oracle</b>                              | <b>Supported</b>     | <b>Supported</b> | <b>Wizard or script</b>      | <b>Hosted on the premises</b>                    |
| <b>Relational database</b>       | <b>Db2</b>                                 | <b>Supported</b>     | <b>Supported</b> | <b>Script</b>                | <b>Hosted on the premises</b>                    |
| <b>Relational database</b>       | <b>DM</b>                                  | <b>Supported</b>     | <b>Supported</b> | <b>script</b>                | <b>Hosted on the premises</b>                    |
| <b>Relational database</b>       | <b>RDS for PPAS</b>                        | <b>Supported</b>     | <b>Supported</b> | <b>Script</b>                | <b>Hosted on Apsara Stack</b>                    |
| <b>Big data storage</b>          | <b>MaxCompute (formerly known as ODPS)</b> | <b>Supported</b>     | <b>Supported</b> | <b>Wizard or script</b>      | <b>Hosted on Apsara Stack</b>                    |
| <b>Big data storage</b>          | <b>DataHub</b>                             | <b>Not supported</b> | <b>Supported</b> | <b>Script</b>                | <b>Hosted on Apsara Stack</b>                    |
| <b>Big data storage</b>          | <b>Elasticsearch</b>                       | <b>Not supported</b> | <b>Supported</b> | <b>Script</b>                | <b>Hosted on Apsara Stack</b>                    |
| <b>Unstructured data storage</b> | <b>OSS</b>                                 | <b>Supported</b>     | <b>Supported</b> | <b>Wizard or script</b>      | <b>Hosted on Apsara Stack</b>                    |
| <b>Unstructured data storage</b> | <b>HDFS</b>                                | <b>Supported</b>     | <b>Supported</b> | <b>Script</b>                | <b>Hosted on the premises</b>                    |
| <b>Unstructured data storage</b> | <b>FTP</b>                                 | <b>Supported</b>     | <b>Supported</b> | <b>Wizard or script</b>      | <b>Hosted on the premises</b>                    |

| <b>Data source category</b> | <b>Data source type</b>  | <b>Reader</b>        | <b>Writer</b>        | <b>Configuration methods</b> | <b>Hosted on Apsara Stack or on the premises</b> |
|-----------------------------|--------------------------|----------------------|----------------------|------------------------------|--------------------------------------------------|
| <b>Message queue</b>        | <b>LogHub</b>            | <b>Supported</b>     | <b>Not supported</b> | <b>Wizard or script</b>      | <b>Hosted on Apsara Stack</b>                    |
| <b>NoSQL database</b>       | <b>HBase</b>             | <b>Supported</b>     | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack or on the premises</b> |
| <b>NoSQL database</b>       | <b>MongoDB</b>           | <b>Supported</b>     | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack or on the premises</b> |
| <b>NoSQL database</b>       | <b>Memcached</b>         | <b>Not supported</b> | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack or on the premises</b> |
| <b>NoSQL database</b>       | <b>Table Store (OTS)</b> | <b>Supported</b>     | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack</b>                    |
| <b>NoSQL database</b>       | <b>OpenSearch</b>        | <b>Not supported</b> | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack</b>                    |
| <b>NoSQL database</b>       | <b>Redis</b>             | <b>Not supported</b> | <b>Supported</b>     | <b>Script</b>                | <b>Hosted on Apsara Stack or on the premises</b> |
| <b>Performance testing</b>  | <b>Stream</b>            | <b>Supported</b>     | <b>Supported</b>     | <b>Script</b>                | <b>N/A</b>                                       |

### 2.8.2.2 Data transmission

The Sync Data Monitoring page displays the total number of instances for different data store wrappers and the instance details based on the selected workspace and time range.

The cut-off time of data to be displayed is 0 minutes 0 seconds of the current hour. For example, if the current time is 2019-04-04 10:10:00, the data generated before 2019-04-04 10:00:00 is displayed on the page.

1. Log on to the DataWorks console.
2. Move your pointer over the DataWorks icon in the upper-left corner, and click Data Integration. The Data Integration page appears.
3. In the left-side navigation bar, click Sync Data Monitoring, and view the total number of instances for different data store wrappers and the instance details.

- View summary data by wrapper type

The summary data of the source end is displayed on the left, and that of the destination end is displayed on the right. Take the source end as an example . If the page shows a MaxCompute node with the value 1, it indicates that the source end is a MaxCompute node, which has been run once in the selected time range.

- View instance details

The Sync Instances section displays the details of all instances running in the selected time range.

- Click the corresponding node in the Node Name column to redirect to the node configuration page.
- In the Sync Instances section, you can search by conditions, such as ID , submitter, node name, data type of the source end, and data type of the destination end. You can also sort search results by the number of synchronized data records or synchronized data size.

### 2.8.2.3 Test data store connectivity

This topic describes the data store types that support connectivity testing and provides sample FAQs related to data store connectivity testing.

| Data store | Data store type                                                        | Network type    | Connectivity testing | Require custom resource groups |
|------------|------------------------------------------------------------------------|-----------------|----------------------|--------------------------------|
| MySQL      | ApsaraDB                                                               | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Supported            | -                              |
|            | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on premises without a public IP address |                 | Not supported        | Yes                            |
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
| PostgreSQL | ApsaraDB                                                               | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Supported            | -                              |
|            | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on premises without a public IP address |                 | Not supported        | Yes                            |
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
| Oracle     | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on premises without a public IP address |                 | Not supported        | Yes                            |

| Data store | Data store type                                                        | Network type    | Connectivity testing | Require custom resource groups |
|------------|------------------------------------------------------------------------|-----------------|----------------------|--------------------------------|
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
| MaxCompute | ApsaraDB                                                               | Classic network | Supported            | -                              |
| OSS        | ApsaraDB                                                               | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Supported            | -                              |
| HDFS       | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
| FTP        | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on premises without a public IP address |                 | Not supported        | Yes                            |
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
| MongoDB    | ApsaraDB                                                               | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Coming soon          | Yes                            |
|            | User-created data store hosted on premises with a public IP address    |                 | Supported            | -                              |
|            | User-created data store hosted on ECS                                  | Classic network | Supported            | -                              |
|            |                                                                        | VPC             | Not supported        | Yes                            |
|            |                                                                        |                 |                      |                                |

| Data store  | Data store type                                                     | Network type    | Connectivity testing | Require custom resource groups |
|-------------|---------------------------------------------------------------------|-----------------|----------------------|--------------------------------|
| Memcache    | ApsaraDB                                                            | Classic network | Supported            | -                              |
|             |                                                                     | VPC             | Coming soon          | Yes                            |
| Redis       | ApsaraDB                                                            | Classic network | Supported            | -                              |
|             |                                                                     | VPC             | Coming soon          | Yes                            |
|             | User-created data store hosted on premises with a public IP address |                 | Supported            | -                              |
|             | User-created data store hosted on ECS                               | Classic network | Supported            | -                              |
|             |                                                                     | VPC             | Not supported        | Yes                            |
|             |                                                                     |                 |                      |                                |
| Table Store | ApsaraDB                                                            | Classic network | Supported            | -                              |
|             |                                                                     | VPC             | Coming soon          | Yes                            |
| DataHub     | ApsaraDB                                                            | Classic network | Supported            | -                              |
|             |                                                                     | VPC             | Not supported        | -                              |

#### Note

In this table, a hyphen (-) means that custom resource groups are not supported for data stores of the corresponding type. "Not supported" means that connectivity testing is not supported for data stores of the corresponding type, though you can still configure data synchronization nodes for these data stores after you add custom resource groups.

- Data stores in VPC
  - RDS data stores in VPC have supported connectivity testing.
  - Other types of data stores in VPC are being planned to support connectivity testing.
  - Currently, DataWorks does not support connectivity testing in Finance Cloud networks.

- **User-created data stores hosted on ECS**
  - **DataWorks allows you to conduct connectivity testing on ECS-hosted data stores in the classic network by setting up a Java Database Connectivity (JDBC ) connection. Generally, connectivity testing is conducted over the public network.**
  - **Currently, DataWorks does not support connectivity testing on ECS-hosted data stores in VPC.**
  - **Currently, DataWorks does not support connectivity testing across regions.**
  - **Currently, DataWorks does not support connectivity testing in Finance Cloud networks.**

**When configuring a security group for an ECS-hosted data store, add the IP address of the scheduling cluster to the inbound and outbound rules of the security group. You must perform this operation regardless of whether the data store is deployed in a VPC or the classic network. If the security group is not properly configured, data synchronization fails due to a network connection failure.**

**Use an API, instead of the console, to set a wide port range for a security group rule.**

- **User-created data stores hosted on-premises or ECS without public IP addresses**
  - **Connectivity testing is not supported.**
  - **You must add a custom resource group before configuring data synchronization nodes.**
- **User-created data stores hosted on-premises or ECS with public IP addresses**

**Conduct connectivity testing on the data stores by setting up a JDBC connection over the public network. If a connectivity test fails, check the constraints configured for your local network and database.**



**Note:**

**Currently, Data Integration in DataWorks is free of charge while you need to pay for products involved in data synchronization nodes as required. Take data synchronization from ApsaraDB RDS for MySQL to MaxCompute as an example.**

**Data synchronization from or to MaxCompute is free of charge. However, you need to pay for the public network address of the MaxCompute tunnel if you have**



configured one in the code editor. The public network address of the MaxCompute tunnel is not available in the template generated in the code editor.

## Issues

When a connectivity test fails, verify that the region, network type, RDS whitelist, database name, and username are properly configured for the data store. Examples of common errors are described as follows:

- The database password is incorrect.
- The network connection fails.
- A network error occurs during synchronization.

Check the log and determine which resource group incurs the issue. Check whether the problematic resource group is a custom one.

If a custom resource group is used, check whether its IP address has been added to the whitelist of the data store such as RDS. Do the same check if the data store type is MongoDB.

Check whether the connectivity test between the two data stores has succeeded and whether you have added all server IP addresses to the whitelist of each data store. If the IP address of a server is not added to one of the data store whitelists, the data synchronization node fails when it runs on this server. However, the node succeeds when it runs on another server whose IP address has been added to the whitelists.

- The result shows that a node is run successfully but the log contains a disconnection error 8000.

This issue occurs when a custom resource group is used and no inbound rule is configured for the IP address 10.116.134.123 and port 8000 in the security group. To resolve the issue, add the IP address and port to the inbound rule of the security group and run the node again.

## Examples of connectivity test failures

### Example 1

- Symptom

A data store failed the connectivity test. Database URL: jdbc:mysql://xx.xx.xx.x:xxxx/t\_uoer\_bradeb. Username: xxxx\_test. Error message: Access denied for user 'xxxx\_test'@'%' to database 'yyyy\_demo'.

- **Troubleshooting**

1. Check whether the configurations are correct.
2. Check whether the password is correct, the whitelist is properly configured, and your account has permission to access the database. You can grant the required permissions in the RDS console.

- **Example 2**

- **Symptom**

A data store failed the connectivity test. The error message is provided as follows:

```
error message: Timed out after 5000 ms while waiting for a server
that matches ReadPreferenceServerSelector{readPreference=primary
}. Client view of cluster state is {type=UNKNOWN, servers=[(
xxxxxxxxxx), type=UNKNOWN, state=CONNECTING, exception={com.
mongodb.MongoSocketReadException: Prematurely reached end of
stream}}}]
```

- **Troubleshooting**

Before testing the connectivity to a MongoDB data store that is not deployed in a VPC, you must add related IP addresses to the whitelist of the data store.

## 2.8.2.4 Add a DataHub connection

DataHub provides a comprehensive data import solution to support fast computing for large amounts of data. You can write data from other data stores into DataHub by using the DataHub writer.

### Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move the pointer over the DataWorks icon in the upper-left corner, and then select Data Integration.
3. Choose Sync Resources > Connections in the left-side navigation pane, and then click Add Connection.
4. In the Add Connection dialog box that appears, select DataHub.
5. Set the parameters for the DataHub connection.

| Parameter       | Description                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------|
| Connection Name | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter. |

| Parameter                         | Description                                                                                                                   |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Description                       | The description of the connection. The description can be up to 80 characters in length.                                      |
| Datahub Endpoint                  | The endpoint of the DataHub project. This parameter is read-only, which is automatically obtained from system configurations. |
| DataHub Project                   | The ID of the DataHub project.                                                                                                |
| AccessKey ID and AccessKey Secret | The logon credentials, which are the account name and the password.                                                           |

6. Click Test Connection.

7. After the connectivity test is passed, click Complete.

The connectivity test checks whether the entered information is correct.

Subsequent steps

Now you have learned how to configure the DataHub connection. You can proceed with the next tutorial, such as configuring the DataHub writer. For more information, see [Configure the DataHub writer](#).

### 2.8.2.5 Add a DM connection

DataWorks provides the DM reader and writer for you to read data from and write data to DM data stores. You can configure data synchronization nodes for DM data stores by using the codeless UI or code editor.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move the pointer over the DataWorks icon in the upper-left corner, and then select Data Integration.
3. Choose Sync Resources > Connections in the left-side navigation pane, and then click Add Connection.
4. In the Add Connection dialog box that appears, select DM.

## 5. Set the parameters for the DM connection.

The DM connection type can be set to User-Created Data Store with Public IP Addresses or User-Created Data Store without Public IP Addresses.

- User-Created Data Store with Public IP Addresses

| Parameter             | Description                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------|
| Connect To            | The connection type. In this example, the DM connection type is User-Created Data Store with Public IP Addresses.   |
| Connection Name       | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter. |
| Description           | The description of the connection. The description can be up to 80 characters in length.                            |
| JDBC URL              | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                 |
| Username and Password | The username and password used to connect to the database.                                                          |

- User-Created Data Store without Public IP Addresses

| Parameter       | Description                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connect To      | The connection type. In this example, the DM connection type is User-Created Data Store without Public IP Addresses. You need to run data synchronization nodes that involve this type of connection on custom resource groups. For more information, click here in the wizard. |
| Connection Name | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter.                                                                                                                                                             |
| Description     | The description of the connection. The description can be up to 80 characters in length.                                                                                                                                                                                        |
| Resource Groups | The resource group on which data synchronization nodes that involve this type of connection are run. Each resource group consists of one or more servers.                                                                                                                       |
| JDBC URL        | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                                                                                                                                                                             |

| Parameter             | Description                                                |
|-----------------------|------------------------------------------------------------|
| Username and Password | The username and password used to connect to the database. |

6. Click Test Connection.

7. After the connectivity test is passed, click Complete.

The connectivity test checks whether the entered information is correct.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the JDBC URL is correct and whether the username and password are valid.
- Connectivity testing is not supported for data stores in VPCs and data stores hosted on the premises without public IP addresses. You can click Complete without testing the connectivity.

### 2.8.2.6 Add FTP data sources

DataWorks supports reading and writing data to FTP data sources with the FTP reader and writer. You can configure data synchronization tasks for FTP data sources either in wizard or script mode.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select FTP.
5. Set the parameters for the FTP data source.

Two options are available for the Data Source Type of the FTP data source.

- Public IP Address Available

| Parameter        | Description                                                                     |
|------------------|---------------------------------------------------------------------------------|
| Data Source Type | The data source type. In this example, Public IP Address Available is selected. |

| Parameter             | Description                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                    |
| Description           | The description of the data source, which cannot exceed 80 characters in length.                                                      |
| Protocol              | The protocol adopted by the FTP host. Currently, only FTP and SFTP are supported.                                                     |
| Host                  | The IP address of the FTP host.                                                                                                       |
| Port                  | The port of the FTP host. This parameter defaults to 21 if you select SFTP as the adopted protocol, defaults to 22 if you select FTP. |
| Username and Password | The account and password used to access the FTP service.                                                                              |

- **Public IP Address Unavailable**

| Parameter             | Description                                                                                                                                                                                                                                          |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Type      | The data source type. In this example, Public IP Address Unavailable is selected. Data synchronization tasks that involve data sources of this type must run custom resource groups. For more information, click <a href="#">here</a> in the wizard. |
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                                                                                                                                   |
| Description           | The description of the data source, which cannot exceed 80 characters in length.                                                                                                                                                                     |
| Resource Group        | The resource group used to run synchronization tasks. You can add multiple servers to a resource group when you create the resource group.                                                                                                           |
| Protocol              | Currently, only FTP and SFTP protocols are supported.                                                                                                                                                                                                |
| Host                  | The IP address of the FTP host.                                                                                                                                                                                                                      |
| Port                  | The port of the FTP host. This parameter defaults to 21 if you select SFTP as the adopted protocol, defaults to 22 if you select FTP.                                                                                                                |
| Username and Password | The account and password used to access the FTP service.                                                                                                                                                                                             |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

The connectivity test checks whether the specified information is correct.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the host IP address, port, username, and password are correct.
- Connectivity testing are not supported for data sources in VPC networks. You can click Complete without testing the connectivity.

Subsequent steps

For more information about how to configure the FTP writer, see [Configure FTP Reader](#) and [Configure the FTP writer](#).

### 2.8.2.7 Add HDFS data sources

DataWorks supports reading and writing data to Hadoop Distributed File System (HDFS) data sources with the HDFS reader and writer. You can configure data synchronization tasks for HDFS data sources in script mode.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select HDFS.
5. Set the parameters for the HDFS data source.

| Parameter        | Description                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Name | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description      | The description of the data source, which cannot exceed 80 characters in length.                                   |
| DefaultFS        | The address of the HDFS node, in the format of <code>hdfs://ServerIP:Port</code> .                                 |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

The connectivity test checks whether the specified information is correct.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the JDBC URL, username, and password are correct.
- Connectivity testing are not supported for data sources in VPC networks. You can click Complete without testing the connectivity.

Subsequent steps

For more information about how to configure the HDFS writer, see [Configure the HDFS reader](#) and [Configure the HDFS writer](#).

## 2.8.2.8 Add LogHub data sources

DataWorks supports reading and writing data to LogHub data sources with the LogHub reader and writer.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select LogHub.
5. Set the parameters for the LogHub data source.

| Parameter        | Description                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Name | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description      | The description of the data source, which cannot exceed 80 characters in length.                                   |
| LogHub Endpoint  | The endpoint of the LogHub project, in the format of http://cn-shanghai.log.aliyun.com.                            |
| Project          | The name of the LogHub project.                                                                                    |



| Parameter                         | Description                                                                |
|-----------------------------------|----------------------------------------------------------------------------|
| AccessKey ID and AccessKey Secret | The logon credential, which consists of AccessKey ID and AccessKey Secret. |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

The connectivity test checks whether the specified project name and AccessKey are correct.

Subsequent steps

For more information, see [Configure the LogHub reader](#) and [Configure LogHub Writer](#).

### 2.8.2.9 Add MaxCompute data sources

MaxCompute (formerly known as ODPS) provides a comprehensive data import solution that allows quicker massive data computing. DataWorks supports reading and writing data to MaxCompute data sources with the MaxCompute reader and writer.



**Note:**

DataWorks automatically creates a data source with a name of odps\_first for each workspace from the MaxCompute project that serves as the compute engine.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select MaxCompute (ODPS).
5. Set the parameters for the MaxCompute data source.

| Parameter        | Description                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Name | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |

| Parameter                         | Description                                                                                                                      |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Description                       | The description of the data source, which cannot exceed 80 characters in length.                                                 |
| ODPS Endpoint                     | The endpoint of the MaxCompute project. This parameter is read only, which is automatically obtained from system configurations. |
| MaxCompute Project Name           | The name of the MaxCompute project.                                                                                              |
| AccessKey ID and AccessKey Secret | The logon credential, which consists of AccessKey ID and AccessKey Secret.                                                       |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

The connectivity test checks whether the project name and AccessKey information are correct.

Subsequent steps

For more information about how to configure the MaxCompute writer, see [Configure MaxCompute Reader](#) and [Configure the MaxCompute writer](#).

### 2.8.2.10 Add Memcached data sources

DataWorks supports writing data to Memcached (formerly known as OCS) data sources with the Memcached writer. You can configure data synchronization tasks for Memcached data sources in script mode.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select Memcache (OCS).

**5. Set the parameters for the Memcached data source.**

| Parameter             | Description                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description           | The description of the data source, which cannot exceed 80 characters in length.                                   |
| Data Source Type      | The type of the data source. Select Memcache.                                                                      |
| Proxy Host            | The memcached proxy.                                                                                               |
| Port                  | The memcached port, which defaults to 11211.                                                                       |
| Username and Password | The username and password used to connect to the data source.                                                      |

**6. Click Test Connectivity.****7. After the data source has passed the connectivity test, click Complete.**

The connectivity test checks whether the specified information is correct.

Subsequent steps

For more information about how to configure the Memcached writer, see [Configure the Memcache \(OCS\) writer](#).

### 2.8.2.11 Add MySQL data sources

DataWorks supports reading and writing data to MySQL data sources with the MySQL reader and writer.

**Note:**

Note the following information while adding MySQL data sources located in VPC networks.

- **User-created MySQL data sources**
  - You can configure data synchronization nodes that involves such data sources. However, connectivity testing is not supported. You can click **Complete** without testing the connectivity.
  - You must run such data synchronization tasks on custom resource groups. Ensure that each data source can connect to the corresponding resource group. For more information, see [Synchronize data when the source or destination is](#)

*deployed on a private network and Data integration when the networks of both data sources at the source and destination ends are disconnected.*

- **RDS for MySQL data sources**

**You do not need to specify the network type for the data source. Instead, DataWorks automatically identifies the network type.**

#### Procedure

1. **Log on to the DataWorks console as a workspace administrator.**
2. **In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.**
3. **In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.**
4. **In the dialog box, select MySQL.**
5. **Set the parameters for the MySQL data source.**

**MySQL data source types include ApsaraDB for RDS, Public IP Address Available, and Public IP Address Unavailable. You can select a data source type as needed.**

**The following table describes the parameters for a MySQL > ApsaraDB for RDS data source.**

| Parameter            | Description                                                                                                        |
|----------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Type     | The type of the data source. Select ApsaraDB for RDS.                                                              |
| Data Source Name     | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description          | The description of the data source, which cannot exceed 80 characters in length.                                   |
| RDS Instance ID      | You can view the instance ID in the RDS console.                                                                   |
| RDS Instance Account | You can view the account in the security settings of the RDS console.                                              |

| Parameter             | Description                                                   |
|-----------------------|---------------------------------------------------------------|
| Username and Password | The username and password used to connect to the data source. |

The following table describes the parameters for a MySQL > Public IP Address Available data source.

| Parameter             | Description                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Type      | The type of the data source. Select Public IP Address Available.                                                   |
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description           | The description of the data source, which cannot exceed 80 characters in length.                                   |
| JDBC URL              | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                |
| Username and Password | The username and password used to connect to the data source.                                                      |

The following table describes the parameters for a MySQL > Public IP Address Unavailable data source.

| Parameter             | Description                                                                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Type      | The type of the data source. Select Public IP Address Unavailable.                                                                                 |
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                                 |
| Description           | The description of the data source, which cannot exceed 80 characters.                                                                             |
| Resource Group        | The resource group on which data synchronization tasks that involve this data source are run. Each resource group consists of one or more servers. |
| JDBC URL              | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                                                |
| Username and Password | The username and password used to connect to the data source.                                                                                      |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the JDBC URL is correct and whether the username, and password are valid.
- Connectivity testing are not supported for data sources in VPC networks and data sources hosted on the premises without public IP addresses. You can click Complete without testing the connectivity.

Subsequent steps

For more information about how to configure the MySQL writer, see [Configure the MySQL reader](#) and [Configure MySQL Writer](#).

## 2.8.2.12 Add Oracle data sources

DataWorks supports reading and writing data to Oracle data sources with the Oracle reader and writer.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select Oracle.
5. Set the parameters for the Oracle data source.

Oracle data source types include Public IP Address Available and Public IP Address Unavailable. You can select a data source type as needed.

The following table describes the parameters for an Oracle > Public IP Address Available data source.

| Parameter        | Description                                                      |
|------------------|------------------------------------------------------------------|
| Data Source Type | The type of the data source. Select Public IP Address Available. |

| Parameter             | Description                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description           | The description of the data source, which cannot exceed 80 characters in length.                                   |
| JDBC URL              | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                |
| Username and Password | The username and password used to connect to the data source.                                                      |

The following table describes the parameters for an Oracle > Public IP Address Unavailable data source.

| Parameter             | Description                                                                                                                                                                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Type      | The type of the data source. Select Public IP Address Unavailable. You need to run data synchronization tasks that involve this type of data sources on custom resource groups. For more information, click <a href="#">here</a> in the wizard. |
| Data Source Name      | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                                                                                                                              |
| Description           | The description of the data source, which cannot exceed 80 characters.                                                                                                                                                                          |
| JDBC URL              | The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.                                                                                                                                                                             |
| Username and Password | The username and password used to connect to the data source.                                                                                                                                                                                   |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the JDBC URL is correct and whether the username, and password are valid.
- Connectivity testing are not supported for data sources in VPC networks and data sources hosted on the premises without public IP addresses. You can click Complete without testing the connectivity.

## Subsequent steps


For more information about how to configure the Oracle writer, see [Configure Oracle Reader](#) and [Configure Oracle Writer](#).

### 2.8.2.13 Add OSS data sources

Alibaba Cloud Object Storage Service (OSS) is a secure and reliable service that enables you to store large amounts of objects.

## Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select OSS.
5. Set the parameters for the OSS data source.

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Name | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                                                                                                                                                                                                                                                                                                                 |
| Description      | The description of the data source, which cannot exceed 80 characters in length.                                                                                                                                                                                                                                                                                                                                                   |
| Endpoint         | <p>The OSS endpoint, in the format of <code>http://oss.aliyuncs.com</code>. The OSS endpoint varies with the region.</p> <div> <b>Note:</b><br/>If you add the bucket name before the domain name (for example, <code>http://oss.aliyuncs.com</code>), the data source can pass the connectivity test but data synchronization will fail.</div> |



| Parameter                         | Description                                                                                                                                                                                                                                                                       |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bucket                            | The name of the OSS bucket. A bucket is a storage space that serves as a container for storing objects. You can create one or more buckets and add one or more objects to each bucket. DataWorks can only search for objects in the specified bucket during data synchronization. |
| AccessKey ID and AccessKey Secret | The AccessKey, which consists of AccessKey ID and AccessKey Secret. It serves as logon credentials.                                                                                                                                                                               |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

Note on connectivity testing

- Connectivity tests are conducted in the classic network to test whether the endpoint and AccessKey information are correct.
- Connectivity testing are not supported for data sources in VPC networks. You can click Complete without testing the connectivity.

Subsequent steps

For more information about how to configure the OSS writer, see [Configure the OSS reader](#) and [Configure the OSS writer](#).

## 2.8.2.14 Add a Table Store connection

Table Store is a NoSQL database service built on Apsara distributed operating system that allows you to store and access large amounts of structured data in real time.

Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move the pointer over the DataWorks icon in the upper-left corner, and then select Data Integration.
3. Choose Sync Resources > Connections in the left-side navigation pane, and then click Add Connection.
4. In the Add Connection dialog box that appears, select Table Store (OTS).

**5. Set the parameters for the Table Store connection.**

| Parameter                         | Description                                                                                                         |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Connection Name                   | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter. |
| Description                       | The description of the connection. The description can be up to 80 characters in length.                            |
| Endpoint                          | The endpoint of the Table Store service.                                                                            |
| Table Store Instance ID           | The ID of the Table Store instance.                                                                                 |
| AccessKey ID and AccessKey Secret | The logon credentials, which are the account name and the password.                                                 |

**6. Click Test Connection.****7. After the connectivity test is passed, click Complete.**

Note on connectivity testing

- On a classic network, the connectivity test checks whether the entered information is correct.
- Connectivity testing is not supported for data stores in VPCs. You can click Complete without testing the connectivity.

Subsequent steps

Now you have learned how to configure the Table Store connection. You can proceed with the next tutorial, such as configuring the Table Store reader. For more information, see [Configure the OTS reader](#).

### 2.8.2.15 Add a PostgreSQL connection

DataWorks provides the PostgreSQL reader and writer for you to read data from and write data to PostgreSQL data stores. You can configure data synchronization nodes for PostgreSQL data stores either by using the codeless UI or code editor.

**Note:**

Note the following information while connecting PostgreSQL data stores located in VPCs:

- **User-created PostgreSQL data stores**
  - You can configure data synchronization nodes that involve such data stores. However, connectivity testing is not supported. You can click **Complete** without testing the connectivity.
  - You must run such data synchronization nodes on custom resource groups. Ensure that each data store can connect to the corresponding resource group. For more information, see [Synchronize data when the source or destination is deployed on a private network](#) and [Data integration when the networks of both data sources at the source and destination ends are disconnected](#).
- **ApsaraDB for PostgreSQL data stores**

You do not need to specify the network type for the connection. Instead, DataWorks automatically identifies the network type.

#### Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move the pointer over the DataWorks icon in the upper-left corner, and then select **Data Integration**.
3. Choose **Sync Resources > Connections** in the left-side navigation pane, and then click **Add Connection**.
4. In the **Add Connection** dialog box that appears, select **PostgreSQL**.
5. Set the parameters for the PostgreSQL connection.

The PostgreSQL connection type can be set to **ApsaraDB for PostgreSQL**, **User-Created Data Store with Public IP Addresses**, or **User-Created Data Store without Public IP Addresses**.

The following table describes the parameters required if the PostgreSQL connection type is **ApsaraDB for PostgreSQL**.

| Parameter       | Description                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------|
| Connect To      | The connection type. In this example, the PostgreSQL connection type is <b>ApsaraDB for PostgreSQL</b> .            |
| Connection Name | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter. |

| Parameter               | Description                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------|
| Description             | The description of the connection. The description can be up to 80 characters in length.                     |
| RDS Instance ID         | The ID of the ApsaraDB for PostgreSQL instance . You can view the ID in the ApsaraDB for PostgreSQL console. |
| RDS Instance Account ID | The ID of the Alibaba Cloud account that has purchased the ApsaraDB for PostgreSQL instance.                 |
| Database Name           | The name of the database.                                                                                    |
| Username and Password   | The username and password used to connect to the database.                                                   |

The following table describes the parameters required if the PostgreSQL connection type is User-Created Data Store with Public IP Addresses.

| Parameter             | Description                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------|
| Connect To            | The connection type. In this example, the PostgreSQL connection type is User-Created Data Store with Public IP Addresses. |
| Connection Name       | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter.       |
| Description           | The description of the connection. The description can be up to 80 characters in length.                                  |
| JDBC URL              | The JDBC URL, in the format of jdbc:postgresql://ServerIP:Port/Database.                                                  |
| Username and Password | The username and password used to connect to the database.                                                                |

The following table describes the parameters required if the PostgreSQL connection type is User-Created Data Store without Public IP Addresses.

| Parameter  | Description                                                                                                                  |
|------------|------------------------------------------------------------------------------------------------------------------------------|
| Connect To | The connection type. In this example, the PostgreSQL connection type is User-Created Data Store without Public IP Addresses. |

| Parameter             | Description                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Name       | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter.                               |
| Description           | The description of the connection. The description can be up to 80 characters in length.                                                          |
| Resource Groups       | The resource group on which data synchronization nodes that involve this data store are run. Each resource group consists of one or more servers. |
| JDBC URL              | The JDBC URL, in the format of jdbc:postgresql://ServerIP:Port/Database.                                                                          |
| Username and Password | The username and password used to connect to the database.                                                                                        |

6. Click Test Connection.

7. After the connectivity test is passed, click Complete.

Note on connectivity testing

- On a classic network, the connectivity test checks whether the entered information is correct.
- In a VPC, if you use an instance to configure the connection, the connectivity test checks whether the entered information is correct.
- In a VPC, if you use an internal address as the JDBC URL to configure the connection, the connectivity test will fail.
- If you use a public address as the JDBC URL to configure the connection, the connectivity test checks whether the entered information is correct.

Subsequent steps

Now you have learned how to configure the PostgreSQL connection. You can proceed with the next tutorial, such as configuring the PostgreSQL writer. For more information, see [Configure PostgreSQL Reader](#) and [Configure the PostgreSQL writer](#).

## 2.8.2.16 Add Redis data sources

Redis is a document-based in-memory NoSQL database service for persistent storage. Based on its reliable master/slave hot backup mechanism and scalable cluster architecture, Redis can meet business needs that require high read/write performance and flexible capacity configuration. DataWorks supports reading

and writing data to Redis data sources with the Redis reader and writer. You can configure data synchronization tasks for Redis data sources in script mode.


#### Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
3. In the upper-right corner, click Add Data Source. A dialog box showing all supported data source types appears.
4. In the dialog box, select Redis.
5. Set the parameters for the Redis data source.

Data Source Type can be set to ApsaraDB or Public IP Address Available.

- **ApsaraDB:** These databases are located in the classic network. Databases that reside in the classic network within the same region can connect to each other, but those in different regions may not.
- **Public IP Address Available:** User-created databases hosted on the premises with public IP addresses. These databases are typically located in public networks, which may incur certain costs.

The following table describes parameters required if you select Redis > ApsaraDB.

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Type | <p>The data source type. In this example, ApsaraDB is selected.</p> <div> <b>Note:</b><br/>If you have not assigned the default role to Data Integration, log on to the RAM console with your Apsara Stack tenant account and perform authorization. Then, we recommend you refresh this configuration page.</div> |
| Data Source Name | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter.                                                                                                                                                                                                                                                                                    |

| Parameter         | Description                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------|
| Description       | The description of the data source, which cannot exceed 80 characters in length.                  |
| Redis Instance ID | The ID of the Redis instance. You can view this ID in the Redis console.                          |
| Redis Password    | The password used to connect to the Redis data source. Leave it blank if no password is required. |

The following table describes parameters required if you select Redis > Public IP Address Available.

| Parameter          | Description                                                                                                        |
|--------------------|--------------------------------------------------------------------------------------------------------------------|
| Data Source Type   | The data source type. In this example, Public IP Address Available is selected.                                    |
| Data Source Name   | The name of the data source, which can contain letters, numbers, and underscores (_). It must start with a letter. |
| Description        | The description of the data source, which cannot exceed 80 characters in length.                                   |
| Server Address     | The endpoint in the format of host:port.                                                                           |
| Add Server Address | Click the button to add an address.                                                                                |
| Redis Password     | The password used to connect to the Redis data source.                                                             |

6. Click Test Connectivity.

7. After the data source has passed the connectivity test, click Complete.

Subsequent steps

For more information about how to configure the Redis writer, see [Configure the Redis writer](#).

### 2.8.2.17 Add a MongoDB connection

MongoDB is a document-oriented database that is second only to Oracle and MySQL. DataWorks provides the MongoDB reader and writer for you to read data from and write data to MongoDB data stores. You can configure data synchronization nodes for MongoDB data stores in the code editor.



**Note:**

To add a MongoDB connection, you need to add the following IP addresses to the whitelist of the MongoDB data store in the corresponding console. You need to separate the IP addresses with a comma (,).

11.192.97.82, 11.192.98.76, 10.152.69.0/24, 10.153.136.0/24, 10.143.32.0/24, 120.27.160.26, 10.46.67.156, 120.27.160.81, 10.46.64.81, 121.43.110.160, 10.117.39.238, 121.43.112.137, 10.117.28.203, 118.178.84.74, 10.27.63.41, 118.178.56.228, 10.27.63.60, 118.178.59.233, 10.27.63.38, 118.178.142.154, 10.27.63.15, 100.64.0.0/8

#### Procedure

1. Log on to the DataWorks console as a workspace administrator.
2. Move the pointer over the DataWorks icon in the upper-left corner, and then select Data Integration.
3. Choose Sync Resources > Connections in the left-side navigation pane, and then click Add Connection.
4. In the Add Connection dialog box that appears, select MongoDB.




## 5. Set the parameters for the MongoDB connection.

The MongoDB connection type can be set to ApsaraDB for MongoDB or User-Created Data Store with Public IP Addresses.


- **ApsaraDB for MongoDB:** Classic networks are generally used. Classic networks in the same region can be connected, while classic networks in different regions cannot be connected.
- **User-Created Data Store with Public IP Addresses:** The public network is generally used, which may cost you certain fees.

The following table describes the parameters required if the MongoDB connection type is ApsaraDB for MongoDB.

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connect To      | <p>The connection type. In this example, the MongoDB connection type is ApsaraDB for MongoDB.</p> <div> <b>Note:</b><br/>If you have not assigned the default role to Data Integration, log on to the RAM console with your Apsara Stack tenant account and perform authorization. Then, refresh this configuration page.</div> |
| Connection Name | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter.                                                                                                                                                                                                                                                                                                |
| Description     | The description of the connection. The description can be up to 80 characters in length.                                                                                                                                                                                                                                                                                                                           |
| Region          | The region selected when the MongoDB instance is created.                                                                                                                                                                                                                                                                                                                                                          |
| Instance ID     | The ID of the MongoDB instance. You can view the MongoDB instance ID in the MongoDB console.                                                                                                                                                                                                                                                                                                                       |
| Database Name   | The name of the database you created in the MongoDB console. You can also specify the database username and password.                                                                                                                                                                                                                                                                                              |

| Parameter             | Description                                                |
|-----------------------|------------------------------------------------------------|
| Username and Password | The username and password used to connect to the database. |

The following table describes the parameters required if the MongoDB connection type is User-Created Data Store with Public IP Addresses.

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connect To            | The connection type. In this example, the MongoDB connection type is User-Created Data Store with Public IP Addresses.                                                                                                                                                                                                                                                                                                                                    |
| Connection Name       | The name of the connection. The name can contain letters, digits, and underscores (_). It must start with a letter.                                                                                                                                                                                                                                                                                                                                       |
| Description           | The description of the connection. The description can be up to 80 characters in length.                                                                                                                                                                                                                                                                                                                                                                  |
| Address               | The endpoint in the format of host:port.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Add Address           | <p>The extra endpoint in the format of host:port . To add an extra endpoint, click Add Address and specify the endpoint to add. To add more endpoints, repeat the preceding action.</p> <div> <b>Note:</b><br/>The endpoints that you add at a time must all be public endpoints or private endpoints. A mixture of public and private endpoints is not allowed.</div> |
| Database Name         | The name of the database.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Username and Password | The username and password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                |

6. Click Test Connection.

7. After the connectivity test is passed, click Complete.



**Note:**

- To connect to a MongoDB database located in a VPC, set Connect To to User-Created Data Store with Public IP Addresses.
- Currently, connectivity testing is not supported for data stores in VPCs.

## Subsequent steps

Now you have learned how to configure the MongoDB connection. You can proceed with the next tutorial, such as configuring the MongoDB writer. For more information, see [Configure MongoDB Reader](#) and [Configure MongoDB Writer](#).

## 2.8.3 Configure data synchronization tasks

### 2.8.3.1 Configure a data synchronization node by using the codeless UI

This topic describes how to configure a data synchronization node by using the codeless UI.

To configure a data synchronization node, follow these steps:

1. Add a connection.
2. Create a data synchronization node.
3. Select a source.
4. Select a destination.
5. Map the fields in the source and destination tables.
6. Configure the maximum transmission rate and dirty data check rules.
7. Schedule the data synchronization node.



#### Note:

The following sections describe the overall procedure. You can click the links in each step to refer to relevant documents and then return to the current page to proceed with subsequent steps.

## Add a connection

Data synchronization is supported between various homogenous and heterogeneous data stores. Before you configure a data synchronization node, add related connections in Data Integration. Added connections are listed as options when you configure data synchronization nodes. For more information about data store types supported by Data Integration, see [Supported data sources](#).

You can connect data stores of supported types to Data Integration. For more information about how to add a connection, see [Data sources](#).



#### Note:

- **Data Integration does not support connectivity testing for some data store types. For more information, see [Test data store connectivity](#).**
- **Some data stores are hosted on the premises, and they do not have public IP addresses or network connections cannot be directly established. Such data stores will fail the connectivity test. Data Integration allows you to add a custom resource group to resolve these issues. However, if you need to create data synchronization nodes for such data stores, you can only use the code editor . This is because you cannot obtain information such as table schema on the codeless UI if the network connection is unavailable.**

Create a data synchronization node



**Note:**

This section describes how to create a data synchronization node by using the codeless UI. Do not switch to the code editor.

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
3. In the Create Workflow dialog box, set Workflow Name and Description. Then, click Create.
4. Expand the created workflow, right-click Data Integration, choose Create Data Integration Node > Sync, and then set Node Name in the dialog box that appears.
5. Click Commit.

Select a source

After creating a data synchronization node, specify the source data store and table.



**Note:**

- For more information about how to specify the source data store, see [Configure Reader](#).
- Incremental data synchronization is required when configuring the source data store for some synchronization nodes. You can perform incremental data synchronization based on the business date specified in the parameter configuration of DataWorks.

Select a destination

After you have completed the source settings, you can configure the destination data store and table.



**Note:**

- For more information about how to specify the destination data store, see [Configure the writer](#).
- You can select the writing method for most nodes. For example, the writing method can be overwriting or appending. Supported writing methods vary according to the data store type.

Map the fields in the source and destination tables

After selecting the source and destination, you must specify the mapping between the fields in the source and destination tables. Options include Map Fields with the Same Name, Map Fields in the Same Line, Delete All Mappings, and Auto Layout.


| Configuration item            | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Map Fields with the Same Name | Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.                                                                                                                                                                                                                                                                                 |
| Map Fields in the Same Line   | Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.                                                                                                                                                                                                                                                                                                    |
| Delete All Mappings           | Click Delete All Mappings to remove mappings that have been established.                                                                                                                                                                                                                                                                                                                                                         |
| Auto Layout                   | The fields are automatically sorted based on specified rules.                                                                                                                                                                                                                                                                                                                                                                    |
| Change Fields                 | You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included , while other blank rows are ignored.                                                                                                                                                                                                                                                                |
| Add                           | <ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as \${bizdate}.</li><li>• You can enter functions supported by relational databases, such as now() and count(1).</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul> |

**Note:**

Ensure that the data type of a source field is the same as that of the mapped destination field or the data type conversion is feasible.

Configure the channel

When the preceding steps are complete, specify the efficiency of the corresponding data synchronization node.

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                        | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>                                                                                                                                |
| Concurrent Threads         | The maximum number of concurrent threads to read and write data to data storage within a single data synchronization node. You can configure the concurrency for a node on the codeless UI.                                                                                                                                                                                                       |
| Bandwidth Throttling       | Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.                                                                                                                  |
| Dirty Data Records Allowed | The maximum number of dirty data records allowed.                                                                                                                                                                                                                                                                                                                                                 |
| Resource Group             | The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance. |

Schedule the data synchronization node

This following section describes how to use scheduling parameters for data filtering.

On the data synchronization node editing page, click Properties on the right side.

You can declare the scheduling parameters by using `${variable name}`. After a variable is declared, enter the initial value of the variable. In this example, the initial value of the variable is identified by `$[]`. The content can be a time expression or a constant.

For example, if you enter `${today}` and `today=${[yyyymmdd]}` in the code, the time variable refers to the current date. For more information about how to perform addition and subtraction, see [Parameter configuration](#).

You can specify the configuration items of the node such as execution cycle, runtime, and dependencies on the page that appears. Data synchronization nodes have no ancestor nodes because their corresponding jobs are run before extract, transform, and load (ETL) jobs. We recommend that you specify the root node as their parent node.

Use custom scheduling parameters

DataWorks provides two default scheduling parameters. Declare them before use.

- **bizdate:** the date-based timestamp of data, which is the day before the node is executed.
- **cycetime:** the time when the node is executed, in the format of `yyymmddhhmiss`.
- DataWorks provides two default scheduling parameters: **bizdate** and **cycetime**.

After configuring the data synchronization node, save and commit the node.

### 2.8.3.2 Configure a data synchronization node by using the code editor

This topic describes how to configure a data synchronization node by using the code editor.

To configure a data synchronization node, follow these steps:

1. Add a connection.
2. Create a data synchronization node.
3. Apply a template.
4. Configure the reader.
5. Configure the writer.
6. Map the fields in the source and destination tables.
7. Configure the maximum transmission rate and dirty data check rules.
8. Schedule the data synchronization node.

**Note:**

The following sections describe the overall procedure. You can click the links in each step to refer to relevant documents and then return to the current page to proceed with subsequent steps.

### Add a connection

Data synchronization is supported between various homogenous and heterogeneous data stores. Before you configure a data synchronization node, add related connections in Data Integration. Added connections are listed as options when you configure data synchronization nodes. For more information about data store types supported by Data Integration, see [Supported data sources](#).

You can connect data stores of supported types to Data Integration. For more information about how to add a connection, see [Data sources](#).

**Note:**

- Data Integration does not support connectivity testing for some data store types. For more information, see [Test data store connectivity](#).
- Some data stores are hosted on the premises, and they do not have public IP addresses or network connections cannot be directly established. Such data stores will fail the connectivity test. Data Integration allows you to add a custom resource group to resolve these issues. However, if you need to create data synchronization nodes for such data stores, you can only use the code editor. This is because you cannot obtain information such as table schema on the codeless UI if the network connection is unavailable.

### Create a data synchronization node

**Note:**

This section describes how to create a data synchronization node in the code editor. Do not switch to the codeless UI.

1. Log on to the DataWorks console.
2. Open the DataStudio page, move the pointer over the Create icon, and then click Workflow.
3. In the Create Workflow dialog box, set Workflow Name and Description. Then, click Create.



4. Expand the created workflow, right-click Data Integration, choose Create Data Integration Node > Sync, and then set Node Name in the dialog box that appears.
5. Click Commit.

Apply a template

1. After creating a data synchronization node, click the Switch to Code Editor icon in the toolbar at the top.
2. Click OK in the Confirm dialog box to switch to the code editor.



**Note:**

The code editor supports more features than the codeless UI. For example, you can configure data synchronization nodes in the code editor even when the connectivity test fails.

3. Click the Apply Template icon in the toolbar at the top.
4. In the Apply Template dialog box that appears, set Source Connection Type, Connection, Target Connection Type, and Connection.
5. Click OK.

Configure the reader

After the template has been applied, the basic settings of the source are complete. You can modify the data store or table for the source.

```
{
 "type": "job",
 "version": "2.0",
 "steps": [
 // Do not modify the preceding lines. They indicate
 // the header code of the data synchronization node.
 {
 "stepType": "mysql",
 "parameter": {
 "datasource": "MySQL",
 "column": [
 "id",
 "value",
 "table"
],
 "socketTimeout": 3600000,
 "connection": [
 {
 "datasource": "MySQL",
 "table": [
 "`case`"
]
 }
],
 "where": "",
 "splitPk": "",
 "encoding": "UTF-8"
 }
 }
],
 "where": "",
 "splitPk": "",
 "encoding": "UTF-8"
},
```

```

 "name": "Reader",
 "category": "reader" // Indicates that these settings
are related to the reader.
 }, // The settings above are related to the reader.

```

The parameters are described as follows:

- **type:** the type of the synchronization node. You can only set it to job.
- **version:** the version of the synchronization node, which can be 1.0 or 2.0.



**Note:**

- For more information about how to configure the data store for the source in the code editor, see [Configure the reader](#).
- Incremental data synchronization is required when configuring the source data store for some synchronization nodes. You can perform incremental data synchronization based on the business date specified in the parameter configuration of DataWorks.

Configure the writer

After you have completed the source settings, you can configure the data store or table for the destination.

```

{
 "stepType": "odps",
 "parameter": {
 "partition": "",
 "truncate": true,
 "compress": false,
 "datasource": "odps_first",
 "column": [
 "x"
],
 "emptyAsNull": false,
 "table": ""
 },
 "name": "Writer",
 "category": "writer" // Indicates that these settings are
related to the writer.
}
], // The settings above are related to the writer.

```



**Note:**

- For more information about how to configure the data store for the destination in the code editor, see [Configure the writer](#).

- You can select the writing method for most nodes. For example, the writing method can be overwriting or appending. Supported writing methods vary according to the data store type.

Map the fields in the source and destination tables

The code editor only supports mapping of fields in the same row. Note that the data types of the fields must match.




**Note:**

Ensure that the data type of a source field is the same as that of the mapped destination field or the data type conversion is feasible.

Configure the channel

When the preceding steps are complete, specify the efficiency of the corresponding data synchronization node. The setting section describes the node efficiency, including the settings on the DMU number, thread concurrency, bandwidth throttling, dirty data policy, and resource group.

```
"setting": {
 "errorLimit": {
 "record": "1024" // The maximum number of dirty data
 records allowed.
 },
 "speed": {
 "throttle": false, // Indicates whether to enable
 bandwidth throttling.
 "concurrent": 1, // The maximum number of concurrent
 threads.
 "dmu": 1// The number of data migration units (DMUs).
 },
},
```

| Parameter          | Description                                                                                                                                                                                                                                                          |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div> |
| Concurrent Threads | The maximum number of concurrent threads to read and write data to data storage within a single data synchronization node. You can configure the concurrency for a node on the codeless UI.                                                                          |

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bandwidth Throttling       | Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.                                                                                                                  |
| Dirty Data Records Allowed | The maximum number of dirty data records allowed.                                                                                                                                                                                                                                                                                                                                                 |
| Resource Group             | The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance. |

Schedule the data synchronization node

This following section describes how to use scheduling parameters for data filtering.

On the data synchronization node editing page, click Properties on the right side.

You can specify the configuration items of the node such as execution cycle, runtime, and dependencies on the page that appears. Data synchronization nodes have no ancestor nodes because their corresponding jobs are run before extract, transform, and load (ETL) jobs. We recommend that you specify the root node as their parent node.

After configuring the data synchronization node, save and commit the node.

### 2.8.3.3 Configure the reader

#### 2.8.3.3.1 Configure the HBase reader

The HBase reader connects to a remote HBase data source through a Java client. Then, it uses the scan method to read data based on the specified rowkey range, converts the data into a dataset based on Data Integration data types, and sends the dataset to the writer.

## Features

- **HBase 0.94.x and 1.1.x versions are supported.**
  - **If you use HBase 0.94.x, set the plugin parameter to hbase094x.**

```
"reader": {
 "plugin": "hbase094x"
}
```

- **If you use HBase 1.1.x, set plugin parameter to hbase11x.**

```
"reader": {
 "plugin": "hbase11x"
}
```

- **Normal and multiVersionFixedColumn modes are supported.**
  - **Normal mode: The HBase reader reads only the latest version of data from an HBase table and converts it into a wide table (that is, binary table). For example:**

```
hbase(main):017:0> scan 'users'
ROW COLUMN+CELL
lisi column=address:city,
timestamp=1457101972764, value=beijing
lisi column=address:contry,
timestamp=1457102773908, value=china
lisi column=address:province,
timestamp=1457101972736, value=beijing
lisi column=info:age, timestamp=
1457101972548, value=27
lisi column=info:birthday,
timestamp=1457101972604, value=1987-06-17
lisi column=info:company, timestamp=1457101972653, value=baidu
xiaoming column=address:city, timestamp=1457082196082, value=
hangzhou
xiaoming column=address:contry, timestamp=1457082195729, value=
china
xiaoming column=address:province, timestamp=1457082195773, value=
zhejiang
xiaoming column=info:age, timestamp=1457082218735, value=29
xiaoming column=info:birthday, timestamp=1457082186830, value=1987
-06-17
xiaoming column=info:company, timestamp=1457082189826, value=
alibaba
2 row(s) in 0.0580 seconds }
```

**The HBase reader converts the data read from HBase to the following table.**

| rowKey | addres:<br>city | address:<br>contry | address:<br>province | info:<br>age | info:<br>birthday | info:<br>company |
|--------|-----------------|--------------------|----------------------|--------------|-------------------|------------------|
| lisi   | beijing         | china              | beijing              | 27           | 1987-06-17        | baidu            |

| rowKey   | address:<br>city | address:<br>contry | address:<br>province | info:<br>age | info:<br>birthday | info:<br>company |
|----------|------------------|--------------------|----------------------|--------------|-------------------|------------------|
| xiaoming | hangzhou         | china              | zhejiang             | 29           | 1987-06-17        | alibaba          |

- **multiVersionFixedColumn mode:** The HBase reader reads data from an HBase table and converts it into a narrow table. Each data record consists of the four columns: rowKey, family:qualifier, timestamp, and value. You need to specify the columns from which the HBase reader reads data, and the HBase reader converts each version of a table cell into a data record.

```
hbase(main):018:0> scan 'users',{VERSIONS=>5}
ROW COLUMN+CELL
lisi column=address:city,
timestamp=1457101972764, value=beijing
lisi column=address:contry,
timestamp=1457102773908, value=china
lisi column=address:province,
timestamp=1457101972736, value=beijing
lisi column=info:age, timestamp=
1457101972548, value=27
lisi column=info:birthday,
timestamp=1457101972604, value=1987-06-17
lisi column=info:company,
timestamp=1457101972653, value=baaidu
xiaoming column=address:city,
timestamp=1457082196082, value=hangzhou
xiaoming column=address:contry,
timestamp=1457082195729, value=china
xiaoming column=address:province,
timestamp=1457082195773, value=zhejiang
xiaoming column=info:age, timestamp=
1457082218735, value=29
xiaoming column=info:age, timestamp=
1457082178630, value=24
xiaoming column=info:birthday,
timestamp=1457082186830, value=1987-06-17
xiaoming column=info:company,
timestamp=1457082189826, value=alibaba
2 row(s) in 0.0260 seconds }
```

The HBase reader converts the data read from HBase to the following table.


| rowKey | column:qualifier | timestamp     | value      |
|--------|------------------|---------------|------------|
| lisi   | address:city     | 1457101972764 | beijing    |
| lisi   | address:contry   | 1457102773908 | china      |
| lisi   | address:province | 1457101972736 | beijing    |
| lisi   | info:age         | 1457101972548 | 27         |
| lisi   | info:birthday    | 1457101972604 | 1987-06-17 |
| lisi   | info:company     | 1457101972653 | beijing    |

| rowKey   | column:qualifier | timestamp     | value      |
|----------|------------------|---------------|------------|
| xiaoming | address:city     | 1457082196082 | hangzhou   |
| xiaoming | address:contry   | 1457082195729 | china      |
| xiaoming | address:province | 1457082195773 | zhejiang   |
| xiaoming | info:age         | 1457082218735 | 29         |
| xiaoming | info:age         | 1457082178630 | 24         |
| xiaoming | info:birthday    | 1457082186830 | 1987-06-17 |
| xiaoming | info:company     | 1457082189826 | alibaba    |

The following table lists data types supported by the HBase reader.

| Data Integration data type | HBase data type         |
|----------------------------|-------------------------|
| Long                       | INT, SHORT, and LONG    |
| Double                     | FLOAT and DOUBLE        |
| String                     | STRING and BINARYSTRING |
| Date                       | DATE                    |
| Boolean                    | BOOLEAN                 |

#### Parameters

| Parameter    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Require | Default value |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------------|
| haveKerberos | <p>Indicates whether Kerberos authentication is required. It is required if you specify haveKerberos as True.</p> <div> <b>Note:</b><ul style="list-style-type: none"><li>If this value is true, the following five Kerberos-related parameters must be specified: kerberosKeytabFilePath, kerberosPrincipal, hbaseMasterKerberosPrincipal, hbaseRegionserverKerberosPrincipal, and hbaseRpcProtection.</li><li>If the value is false, Kerberos authentication is not required and you do not need to specify these parameters.</li></ul></div> | No      | false         |

| Parameter          | Description                                                                                                                                                                                                                                                                                                   | Required | Default value |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>hbaseConfig</b> | The properties of the HBase cluster, in JSON format. This parameter must contain the <b>hbase.zookeeper.quorum</b> configuration option, which indicates the ZooKeeper ensemble servers. It can also contain optional configuration options such as those related to the cache and batch for scan operations. | Yes      | None          |
| <b>mode</b>        | The mode in which data is read from the HBase data source. Valid values: <b>normal</b> and <b>multiVersionFixedColumn</b> .                                                                                                                                                                                   | Yes      | None          |
| <b>table</b>       | The name of the HBase table from which data is read. The name is case sensitive.                                                                                                                                                                                                                              | Yes      | None          |
| <b>encoding</b>    | The encoding, to which a string is converted using <b>byte[]</b> . Currently, UTF-8 and GBK are supported.                                                                                                                                                                                                    | No       | utf-8         |



| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Required | Default value |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>The HBase columns from which data is read.</p> <ul style="list-style-type: none"><li>• <b>In normal mode:</b></li></ul> <p>Specify the name parameter if you need to specify a column in the HBase table. The format must be columnFamily:columnName except for the rowkey. The type parameter indicates the source data type and the format parameter indicates the date format. Specify the value parameter if you need a constant column. The HBase reader automatically creates a constant column instead of reading it from the HBase table. An example is provided as follows:</p> <pre>"column":<br/>[<br/>  {<br/>    "Name": "rowkey",<br/>    "type": "string"<br/>  },<br/>  {<br/>    "value": "test",<br/>    "type": "string"<br/>  }<br/>]</pre> <p>For each column, you need to specify the type parameter and either the name or value parameter.</p> <ul style="list-style-type: none"><li>• <b>In multiVersionFixedColumn mode:</b></li></ul> <p>Specify the name parameter if you need to specify a column in the HBase table. The format must be columnFamily:columnName except for the rowkey. The type parameter indicates the source data type and the format parameter indicates the date format. You cannot create constant columns in multiVersionFixedColumn mode. An example is provided as follows:</p> <pre>"column":<br/>[<br/>  {<br/>    "name": "rowkey",<br/>    "type": "string"<br/>  },<br/>  {<br/>    "value": "test",<br/>    "type": "string"<br/>  }<br/>]</pre> | Yes      | None          |

| Parameter     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Required | Default value |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| range         | <p>The rowkey range that the HBase reader reads.</p> <ul style="list-style-type: none"> <li>• <b>startRowkey</b>: The start rowkey.</li> <li>• <b>endRowkey</b>: The end rowkey.</li> <li>• <b>isBinaryRowkey</b>: The operation called by <code>byte[]</code> to convert the specified start and end rowkeys. Default value: false. If the value is true, <code>Bytes.toBytesBinary(rowkey)</code> is called. If the value is false, <code>Bytes.toBytes(rowkey)</code> is called. An example is provided as follows:</li> </ul> <pre>"range": {   "startRowkey": "aaa",   "endRowkey": "ccc",   "isBinaryRowkey": false }</pre> | No       | None          |
| scanCacheSize | The number of rows read by an HBase client with each RPC connection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | No       | 256           |
| scanBatchSize | The number of columns read by an HBase client with each RPC connection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | No       | 100           |

Configure the HBase reader in wizard mode

**Currently, wizard mode is not supported for the HBase reader.**

Configure the HBase reader in script mode

**In the following script, a task is configured to read data from an HBase data source in normal mode.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "hbase", // The reader type.
 "parameter": {
 "mode": "normal", // The mode in which data is read
 from the HBase data source. Valid values: normal and multiVersionFixedColumn.
 "scanCacheSize": 256, // The number of rows read by an
 HBase client with each RPC connection.
 "scanBatchSize": 100, // The number of columns read
 by an HBase client with each RPC connection.
 "hbaseVersion": "9.4x/11x", // The HBase version.
 "column": [// The columns.
 {
 "name": "rowkey", // The column name.
 "type": "string" // The data type.
 }
]
 }
 }
]
}
```

```

 },
 {
 "name": "columnFamilyName1: columnname1 ",
 "type": "string"
 },
 {
 "name": "columnFamilyName2:columnName2",
 "format": "yyyy-MM-dd",
 "type": "date"
 },
 {
 "name": "columnFamilyName3:columnName3",
 "type": "long"
 }
],
"range": { // The rowkey range that the HBase reader
reads.
 "endRowkey": "", // The end rowkey.
 "isBinaryRowkey": true, // The operation called by
byte[] to convert the specified start and end rowkeys. Default value:
false.
 "startRowkey": "" // The start rowkey.
},
"maxVersion": "", // The number of versions read by the
HBase reader when multiple versions are available.
"encoding": "UTF-8", // The encoding.
"table": "", // The table name.
"hbaseConfig": { // The properties of the HBase cluster
, in JSON format.
 "hbase.zookeeper.quorum": "hostname",
 "hbase.rootdir": "hdfs://ip:port/database",
 "hbase.cluster.distributed": "true"
}
},
"name": "Reader",
"category": "reader"
},
{ // The following template is used to configure the reader.
For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
}
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",

```

```
 "to": "Writer"
 }
]
 }
}
```

### 2.8.3.3.2 Configure the HDFS reader

The HDFS reader enables reading data from a remote Hadoop Distributed File System (HDFS). Specifically, the HDFS reader connects to an HDFS, reads data from the HDFS, converts the data to a format that is readable by the Data Integration service, and sends the converted data to the writer.

**Example:**

Text is the default file format of Hive tables. With this file format, tables are stored as texts in the HDFS and are not compressed. To read Hive tables in the text format, the HDFS reader works like the OSS reader. The Optimized Row Columnar (ORC) format provides an efficient method to store Hive data. The HDFS reader utilizes the OrcSerde class provided by Hive to read and parse ORC file data.



#### Note:

An admin user account and read/write permissions for files are required for data synchronization.

Related commands are described as follows:

- Create an admin user account and the root directory. Add the admin user to a Hadoop user group and the supergroup.

```
useradd -m -G supergroup -g hadoop -p admin admin
```

- `-G supergroup`: specifies the supergroup to which the user belongs.
- `-g hadoop`: specifies the user group to which the user belongs.
- `-p admin admin`: sets a password for the admin user account.
- View the list of files in the directory.

```
hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
```

The Hadoop command format is `hadoop fs -command`. Replace command with a specific command.

- **Copy the part-00000 file to the local file system.**

```
hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
```

- **Edit the file.**

```
vim part-00000
```

- **Log out of the current user account.**

```
exit
```

- **Connect to each host on the host list and create an admin user account on each connected host.**

```
pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -g hadoop -p admin admin
```

- `pssh -h /home/hadoop/slave4pssh`: **connects to each host on the host list.**
- `useradd -m -G supergroup -g hadoop -p admin admin`: **creates an admin account.**

## Features

**Currently, the HDFS reader supports the following features:**

- **File formats:** text, ORC, RC, Sequence, CSV, and Parquet. What is stored in each file must be a two-dimensional table.
- **Constant columns, and column pruning.** It can read various types of data, all stored as strings.
- **Recursive reading, and regular expressions with asterisks (\*) and question marks (?).**
- **ORC file compression options:** SNAPPY and ZLIB.
- **Sequence file compression option:** LZO.
- **Concurrent reading from multiple files.**
- **CSV file compression options:** GZIP, BZ2, ZIP, LZO, LZO\_DEFLATE, and SNAPPY.
- **Currently, the HDFS reader support Hive 1.1.1 and Hadoop 2.7.1 (compatible with Apache JDK 1.6). The HDFS reader can properly work with Hadoop 2.5.0, Hadoop 2.6.0, and Hive 1.2.0 during testing.**



**Note:**

Currently, the HDFS reader cannot use multiple threads to read data from a single file.

## Data types

### RC files

Metadata in RC files are stored in databases managed by Hive, and in different formats depending on the data type. However, the HDFS reader cannot query metadata from such databases. If you need to synchronize an RC file, you need to manually specify the data type for each column. If the data type is BIGINT, DOUBLE, or FLOAT, specify the data type as Bigint/Double/Float. If the data type is VARCHAR or CHAR, specify the data type as String.

RC file data types are automatically converted into the data types supported by Data Integration.

| Data Integration data type | HDFS data type                     |
|----------------------------|------------------------------------|
| Integer                    | TINYINT, SMALLINT, INT, and BIGINT |
| Floating point             | FLOAT, DOUBLE, and DECIMAL         |
| String                     | STRING, CHAR, and VARCHAR          |
| Date and time              | DATE and TIMESTAMP                 |
| Boolean                    | BOOLEAN                            |
| Binary                     | BINARY                             |

### Parquet files

RC file data types are automatically converted into the data types supported by Data Integration.

| Data Integration data type | HDFS data type          |
|----------------------------|-------------------------|
| Integer                    | INT32, INT64, and INT96 |
| Floating point             | FLOAT and DOUBLE        |
| String                     | FIXED_LEN_BYTE_ARRAY    |
| Date and time              | DATE and TIMESTAMP      |
| Boolean                    | BOOLEAN                 |
| Binary                     | BINARY                  |

### Text, ORC, and sequence files

Metadata in text and ORC files are stored in databases, such as MySQL databases, managed by Hive. However, the HDFS reader cannot query metadata from such databases. If you need data type conversion during synchronization, you need to manually specify the data types.

Text, ORC, and sequence file data types are automatically converted into the data types supported by Data Integration.

| Data Integration data type | HDFS data type                                               |
|----------------------------|--------------------------------------------------------------|
| Integer                    | TINYINT, SMALLINT, INT, and BIGINT                           |
| Floating point             | FLOAT and DOUBLE                                             |
| String                     | STRING, CHAR, VARCHAR, STRUCT, MAP, ARRAY, UNION, and BINARY |
| Date and time              | DATE and TIMESTAMP                                           |
| Boolean                    | BOOLEAN                                                      |

The data types are described as follows:



- **LONG:** integer strings in HDFS files, such as 123456789.
- **DOUBLE:** double value strings in HDFS files, such as 3.1415.
- **BOOLEAN:** boolean strings in the HDFS files, such as true and false (case-insensitive).
- **DATE:** date and time strings in HDFS files, such as 2014-12-31 00:00:00.




**Note:**

The **TIMESTAMP** data type of Hive is accurate to nanoseconds. If you convert **TIMESTAMP**-type Hive data (such as 2015-08-21 22:40:47.397898389) in text and ORC files into the date type in Data Integration, then the data will be accurate to seconds. If you need nanosecond-scale accuracy, convert **TIMESTAMP**-type data into the string type in Data Integration.


## Parameters


| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Require | Default value   |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-----------------|
| path      | <p>The path of the file to be read. If you need the HDFS to read multiple files, use a regular expression such as <code>/hadoop/data_201704*</code>.</p> <ul style="list-style-type: none"><li>• If you specify a single HDFS file, the HDFS reader uses only one thread for data reading.</li><li>• If you specify multiple HDFS files, the HDFS reader uses multiple threads. The number of threads is limited by the transmission rate, in Mbit/s.</li></ul> <div> <b>Note:</b><br/>The actual number of threads is determined by the smaller of the number of HDFS files to be read and the specified transmission rate.</div> <ul style="list-style-type: none"><li>• When the parameter value includes a wildcard, the HDFS reader attempts to read all files that match the regular expression. If the path is ended with a slash (/), the HDFS reader reads all files in the specified directory. For example, if you specify the path as <code>/bazhen/</code>, the HDFS reader reads all files in the <code>bazhen</code> directory. Currently, the HDFS reader only supports asterisks (*) and question marks (?) as file name wildcards. The syntax is similar to that of file name wildcards used in the Linux command line.</li></ul> <div> <b>Note:</b><ul style="list-style-type: none"><li>• Data Integration considers all the files in a data synchronization job as a single table. Ensure that all the files in each data synchronization job can adapt to the same schema and grant Data Integration the permission to read all these files.</li><li>• During Hive table creation, you can specify partitions. For example, if you specify <code>partition(day="20150820",hour="09")</code>, a directory named <code>/20150820</code> and a subdirectory <code>/09</code> are created in the corresponding table directory of the HDFS.</li></ul></div> | Yes     | None            |
| 794       | Therefore, if you need the HDFS reader to read the data of a partition, specify the file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |         | Issue: 20200116 |



| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Required | Default value |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| fileType        | <p>The file format. Valid values: text, orc, rc, seq, csv, and parquet. The HDFS reader automatically recognizes the file format and uses corresponding read policies. Before synchronizing data, the HDFS reader checks whether all the source files match the specified type. If any source file does not match the type, the task fails.</p> <p>The valid values of the fileType parameter is described as follows:</p> <ul style="list-style-type: none"> <li>• text: the text file format.</li> <li>• orc: the ORC file format.</li> <li>• rc: the RC file format.</li> <li>• seq: the sequence file format.</li> <li>• csv: the common HDFS file format, that is, two-dimensional logical table.</li> <li>• parquet: the common parquet file format.</li> </ul> <div>  <b>Note:</b> <p>Since text and ORC files are different in the file format, the HDFS reader parses files in the two format in different ways. After being converted from a collection data type into the string type of Data Integration, the data in a file with the text file format can be different from that in the same file with the ORC file format. Collection data types include Map, Array, Struct, and Union. The following example uses the conversion from the Map type to the string type as an example:</p> <ul style="list-style-type: none"> <li>• The HDFS reader converts Map-type ORC file data into a string: {job=80, team=60, person=70}.</li> <li>• The HDFS reader converts Map-type text file data into a string: job:80, team:60, person:70.</li> </ul> </div> | Yes      | None          |
| Issue: 20200116 | <p>The conversion results show that the data remains unchanged but the formats differ slightly. Therefore, if the data to be synchronized match</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |          | 795           |

| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Required | Default value |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column         | <p>The columns to be read. <b>type</b>: the source data type. <b>index</b>: the ID of the column in the source table, starting from 0. <b>value</b>: the column value if the column is a constant column. To convert all data into the string type, specify this parameter as "column": ["*"].</p> <p>You also can specify the column parameter as follows:</p> <pre>{   "type": "long",   "index": 0 // The first INT-type column of the source file. }, {   "type": "string",   "value": "alibaba" // The value of the current column is a constant "alibaba". }</pre> | Yes      | None          |
| fieldDelimiter | <p>The column delimiter. You need to specify the column delimiter for text files, and the default delimiter is a comma (.). You do not need to specify the column delimiter for ORC files, and the default delimiter is \u0001.</p> <ul style="list-style-type: none"><li>• If you need each row to be converted into a column in the destination table, use a string that does not exist in every row, such as \u0001.</li><li>• Do not use \n as the delimiter.</li></ul>                                                                                              | No       | ,             |
| encoding       | The encoding of the file to be read.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | No       | utf-8         |
| nullFormat     | <p>The string that represents null. Since no standard strings can represent null in text files, Data Integration provides the nullFormat parameter to define which string represents a null pointer.</p> <p>For example, if you specify nullFormat:"null", then Data Integration considers "null" as a null pointer.</p>                                                                                                                                                                                                                                                 | No       | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Required | Default value |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| compress  | <p>The compression option. Available compression options for CSV files are gzip, bz2, zip, lzo, lzo_deflate, hadoop-snappy, and framing-snappy.</p> <div> <b>Note:</b><ul style="list-style-type: none"><li>• Do not mix up lzo and lzo_deflate.</li><li>• Since Snappy does not have a uniform stream format, Data Integration currently only supports the most popular two compression formats: hadoop-snappy (Snappy stream format in Hadoop) and framing-snappy (Snappy stream format recommended by Google).</li><li>• rc indicates the RC file format.</li><li>• This parameter is not required for ORC files.</li></ul></div> | No       | None          |

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Required | Default value |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| parquetSchema   | <p>The schema of the source file. This parameter is only required and takes effect if the fileType parameter is set to parquet. Format:</p> <pre>message messageType {   required, dataType, columnName;   ..... ; }</pre> <p>The configurations are described as follows:</p> <ul style="list-style-type: none"> <li>• <b>messageTypeName</b>: the name of the MessageType object.</li> <li>• <b>required</b>: indicates whether the field is required or optional. The value optional is recommended.</li> <li>• <b>dataType</b>: the data type of the field. Valid values : Boolean, Int32, Int64, Int96, Float, Double, Binary (select Binary if the data type is STRING ), and fixed_len_byte_array.</li> </ul> <p> <b>Note:</b><br/>Note that each line, including the last one, must end with a semicolon (;).</p> <p><b>Example:</b></p> <pre>message m {   optional int64 id;   optional int64 date_id;   optional binary datetimestring;   optional int32 dspId;   optional int32 advertiserId;   optional int32 status;   optional int64 bidding_req_num;   optional int64 imp;   optional int64 click_num; }</pre> | No       | None          |
| csvReaderConfig | <p>The configurations for reading CSV files. The parameter value must match the Map type. A specific CSV reader is used to read data from CSV files, which supports many configurations.</p> <p>The following example provides common configurations:</p> <pre>csvReaderConfig   "safetySwitch": false.</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | No       | None          |

Configure the HDFS reader in script mode

**In the following script, a task is configured to read data from an HDFS.**

```
{
 "type": "job",
 "version": "2.0",
 "steps": [
 {
 "stepType": "hdfs", // The reader type.
 "parameter": {
 "path": "", // The file path to read.
 "datasource": "", // The data source.
 "column": [
 {
 "index": 0, // The ID of the corresponding
source table column.
 "type": "string" // The data type.
 },
 {
 "index": 1,
 "type": "long"
 },
 {
 "index": 2,
 "type": "double"
 },
 {
 "index": 3,
 "type": "boolean"
 },
 {
 "format": "yyyy-MM-dd HH:mm:ss", // The time
format.
 "index": 4,
 "type": "date"
 }
],
 "fieldDelimiter": ",", // The column delimiter.
 "encoding": "UTF-8", // The encoding.
 "fileType": "" // The file format.
 },
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "concurrent": 3, // The maximum number of concurrent
threads.
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
```

```
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}
```

### 2.8.3.3.3 Configure MaxCompute Reader

This topic describes the data types and parameters supported by MaxCompute Reader and how to configure it by using the codeless UI and code editor.

MaxCompute Reader can read data from a MaxCompute data store by using the MaxCompute Tunnel service based on the source project, table, partition, and table fields you have configured.

MaxCompute Reader cannot read views. It can only read partitioned tables and non-partitioned tables. When enabling MaxCompute Reader to read partitioned tables, you need to specify the partitioning information. For example, set `pt` to 1 and `ds` to `hangzhou` for table `t0`. The partitioning information is not required for non-partitioned tables. Additionally, you can select some or all of the table fields, change the order in which the fields are arranged, and add constant fields and partition columns. Note that partition columns are not table fields.



#### Data types

The following table lists the data types supported by MaxCompute Reader.

| Category      | Data Integration data type | MaxCompute data type               |
|---------------|----------------------------|------------------------------------|
| Integer       | Long                       | Bigint, Int, Tinyint, and Smallint |
| Boolean       | Boolean                    | Boolean                            |
| Date and time | Date                       | Datetime and Timestamp             |
| Float         | Double                     | Float, Double, and Decimal         |
| Binary        | Bytes                      | Binary                             |
| Complex       | String                     | Array, Map, and Struct             |

## Parameters

| Parameter  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Required                             | Default value |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|---------------|
| datasource | The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Yes                                  | None          |
| table      | The name of the source table. The name is case insensitive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Yes                                  | None          |
| partition  | <p>The partition that MaxCompute Reader reads. Linux shell wildcards are supported. An asterisk (*) represents any number of characters (in other words, zero or more characters) and a question mark (?) represents that the previous character can be either included or not. For example, a partition table, named test, has four partitions: pt=1 and ds=hangzhou, pt=1 and ds=shanghai, pt=2 and ds=hangzhou, and pt=2 and ds=beijing.</p> <ul style="list-style-type: none"><li>• If you need to read data from the partition with pt=1 and ds=shanghai, set "partition": "pt=1/ds=shanghai".</li><li>• If you need to read data from all the partitions with pt=1, set "partition": "pt=1/ds=*".</li><li>• If you need to read data from all the partitions in the table test, set "partition": "pt=*/ds=*".</li></ul> | Required only for partitioned tables | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Required | Default value |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>The columns in the source table that MaxCompute Reader reads. Assume that the fields of a table named test are id, name, and age.</p> <ul style="list-style-type: none"><li>To read the fields in turn, set <code>"column": ["id", "name", "age"]</code> or <code>"column": ["*"]</code>.</li></ul> <div> <b>Note:</b><br/>We do not recommend that you set <code>"column": ["*"]</code>. This is because data synchronization may fail if the source table changes in the column order, data type, or column number.</div> <ul style="list-style-type: none"><li>To read the name and id fields in turn, set <code>"column": ["name", "id"]</code>.</li><li>You can add a constant field to extracted data for the purpose of proper mapping between source table columns and destination table columns. Each constant must be enclosed in single quotation marks ('). For example, if you set <code>"column": ["age", "name", "'1988-08-08 08:08:08'", "id"]</code>, the data extracted includes an age column, a name column, a constant "1988-08-08 08:08:08", and an id column in turn.</li></ul> <p>The single quotation marks (') are used to identify constant columns, and the constant column values exclude the single quotation marks (').</p> <div> <b>Note:</b><ul style="list-style-type: none"><li>MaxCompute Reader does not use SELECT statements to read data. Therefore, you cannot specify function fields.</li><li>The column parameter must explicitly specify a set of columns to be synchronized. It cannot be left empty.</li></ul></div> | Yes      | None          |

Configure MaxCompute Reader by using the codeless UI

**Create a data synchronization node, and configure the node.**



## 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.

| Parameter                     | Description                                                                                                                                                          |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection                    | The datasource parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks. |
| Table                         | The table parameter in the preceding parameter description.                                                                                                          |
| Partition Key Column          | The column for partition.                                                                                                                                            |
| Compression                   | Specifies whether to enable compression. Valid values: Enable and Disable.                                                                                           |
| Convert Empty Strings to Null | Specifies whether to convert empty strings to null.                                                                                                                  |



### Note:

To synchronize all columns in the table, set "column": [""]. The partition parameter supports wildcards and includes one or more partitions.

- "partition": "pt=20140501/ds=\*" indicates that all partitions in the ds partition need to be synchronized.
- "partition": "pt=top?" indicates that the partitions with pt=top and pt=to need to be synchronized.


You can specify the partition columns to be synchronized, such as a partition column named pt. Assume that the partition column of a MaxCompute source table is pt=\${bdp.system.bizdate}. You can configure the pt column to be synchronized. Ignore it if the column is marked as unidentified. To synchronize all partitions, set pt=\${\*}. To synchronize some of the partitions, specify the corresponding dates.

## 2. Configure field mapping (the column parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

| Configuration item            | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Map Fields with the Same Name | Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.                                                                                                                                                                                                                                                                                 |
| Map Fields in the Same Line   | Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.                                                                                                                                                                                                                                                                                                    |
| Delete All Mappings           | Click Delete All Mappings to remove mappings that have been established.                                                                                                                                                                                                                                                                                                                                                         |
| Auto Layout                   | The fields are automatically sorted based on specified rules .                                                                                                                                                                                                                                                                                                                                                                   |
| Change Fields                 | You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, while other blank rows are ignored.                                                                                                                                                                                                                                                                 |
| Add                           | <ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as \${bizdate}.</li><li>• You can enter functions supported by relational databases, such as now() and count(1).</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul> |

## 3. Configure the channel.

| Parameter | Description                                                                                                                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU       | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div> |

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Concurrent Threads         | The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.                                                                                                                     |
| Bandwidth Throttling       | Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.                                                                                                                  |
| Dirty Data Records Allowed | The maximum number of dirty data records allowed.                                                                                                                                                                                                                                                                                                                                                 |
| Resource Group             | The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance. |

Configure MaxCompute Reader by using the code editor

**In the following code, a node is configured to read data from a MaxCompute data store.**

```
{
 "type": "job",
 "version": "2.0",
 "steps": [
 {
 "stepType": "odps", // The reader type.
 "parameter": {
 "partition": [], // The partition that MaxCompute Reader
reads.
 "isCompress": false, // Indicates whether to enable
compression.
 "datasource": "", // The connection name.
 "column": [// The columns to be synchronized.
 "id"
],
 "emptyAsNull": true,
 "table": "" // The table name.
 },
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
 "stepType": "stream",
 "name": "Writer",
 "category": "writer"
 }
]
}
```

```

 "parameter":{
 },
 "name":"Writer",
 "category":"writer"
 }
],
"setting":{
 "errorLimit":{
 "record":"0">// The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "concurrent":1,// The maximum number of concurrent threads
.
 "dmu":1// The DMU value.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}

```

#### 2.8.3.3.4 Configure MongoDB Reader

This topic describes the data types and parameters supported by MongoDB Reader and how to configure it by using the codeless UI and code editor.

MongoDB Reader connects to a remote MongoDB data store by using a Java client named MongoClient and reads data from the data store. The latest version of MongoDB has improved the locking feature from database locks to document locks . With the powerful functionalities of indexes in MongoDB, MongoDB Reader can successfully read data from MongoDB data stores.



#### Note:

- If you use ApsaraDB for MongoDB, the MongoDB database has a root account by default. For security concerns, Data Integration only supports access to a MongoDB database by using a MongoDB database account. When adding a MongoDB connection, do not use the root account for access.
- JavaScript syntax is not supported for queries.

MongoDB Reader shards data in the MongoDB database according to specified rules, reads data from the database with multiple threads, and converts the data into a format readable by Data Integration.

#### Data types

MongoDB Reader supports most MongoDB data types. Ensure that your data types are supported.

The following table lists the data types supported by MongoDB Reader.

| Category | MongoDB data type                                           |
|----------|-------------------------------------------------------------|
| Long     | Int, Long, Document.Int, and Document.Long                  |
| Double   | Double and Document.Double                                  |
| String   | String, Array, Document.String, Document.Array, and Combine |
| Date     | Date and Document.Date                                      |
| Boolean  | Bool and Document.Bool                                      |
| Bytes    | Bytes and Document.Bytes                                    |



#### Note:

- The Document data type is also called the Object data type, which stores embedded documents.
- The following content describes how to use the Combine data type:

When MongoDB Reader reads data from a MongoDB data store, it combines and converts multiple fields in MongoDB documents to a JavaScript Object Notation (JSON) string.

For example, doc1, doc2, and doc3 are three MongoDB documents with different fields, which are represented by keys instead of key-value pairs. The keys a

and **b** represent common fields in all the three documents, while the key **x\_n** represents an unfixed field.

```
doc1: a b x_1 x_2
```

```
doc2: a b x_2 x_3 x_4
```

```
doc3: a b x_5
```

To import the preceding three MongoDB documents to MaxCompute, you need to specify the fields to retain, set a name for each combined string, and set the data type of each combined string to **Combine** in the configuration file. Ensure that the name of each combined string is unique from that of any existing field in the documents.

```
"column": [
 {
 "name": "a",
 "type": "string",
 },
 {
 "name": "b",
 "type": "string",
 },
 {
 "name": "doc",
 "type": "combine",
 }
]
```

The output in MaxCompute is as follows:

| odps_column1 | odps_column2 |
|--------------|--------------|
| a            | b            |
| a            | b            |
| a            | b            |

#### Parameters

| Parameter       | Description                                                                                                                | Required | Default value |
|-----------------|----------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| datasource      | The connection name. It must be identical to the name of the added connection. You can add connections in the code editor. | Yes      | None          |
| collection Name | The name of the MongoDB collection.                                                                                        | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Required | Default value |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <b>The columns in MongoDB.</b> <ul style="list-style-type: none"><li>• <b>name:</b> the column name.</li><li>• <b>type:</b> the data type of the column.</li><li>• <b>splitter:</b> the delimiter. Specify this parameter only when you need to convert the string to an array. MongoDB supports arrays, but Data Integration does not. The array elements read by MongoDB are joined into a string by using this delimiter.</li></ul>                                                                                                                                                                                                                                                                                                              | Yes      | None          |
| query     | <b>The filter condition for obtaining data from MongoDB. Only the time type is supported. For example, you can use the statement <code>"query": "{ 'operationTime': { '\$gte': ISODate('{\$last_day}T00:00:00.424+0800') } }"</code> to obtain data where the time specified by operationTime is not earlier than 00:00 on the day specified by <code>last_day</code>. In the preceding JSON string, <code>last_day</code> is a scheduling parameter of DataWorks. The format is <code>yyyy-mm-dd</code>. You can use conditional operators (<code>\$gt</code>, <code>\$lt</code>, <code>\$gte</code>, and <code>\$lte</code>), logical operators (AND and OR), and functions (max, min, sum, avg, and ISODate) supported by MongoDB as needed.</b> | No       | None          |

Configure MongoDB Reader by using the codeless UI

**Currently, the codeless UI is not supported for MongoDB Reader.**

Configure MongoDB Reader by using the code editor

**In the following code, a node is configured to read data from a MongoDB data store.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "reader": {
 "plugin": "mongodb", // The name of the plug-in.
 "parameter": {
 "datasource": "datasourceName", // The name of the
data store.
 "collectionName": "tag_data", // The name of the
MongoDB collection.
 "query": "",
 "column": [
```

```
{
 "name": "unique_id", // The field name.
 "type": "string" // The data type.
},
{
 "name": "sid",
 "type": "string"
},
{
 "name": "user_id",
 "type": "string"
},
{
 "name": "auction_id",
 "type": "string"
},
{
 "name": "content_type",
 "type": "string"
},
{
 "name": "pool_type",
 "type": "string"
},
{
 "name": "frontcat_id",
 "type": "array",
 "splitter": ""
},
{
 "name": "categoryid",
 "type": "array",
 "splitter": ""
},
{
 "name": "gmt_create",
 "type": "string"
},
{
 "name": "taglist",
 "type": "array",
 "splitter": " "
},
{
 "name": "property",
 "type": "string"
},
{
 "name": "scorea",
 "type": "int"
},
{
 "name": "scoreb",
 "type": "int"
},
{
 "name": "scorec",
 "type": "int"
},
{
 "name": "a.b",
 "type": "document.int"
},
{
```



```

 "name": "a.b.c",
 "type": "document.array",
 "splitter": " "
 }
]
 },
 { // The following template is used to configure the writer.
 For more information, see the document of the corresponding writer.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "throttle": false, // A value of false indicates that
 the bandwidth is not throttled. A value of true indicates that the
 bandwidth is throttled. The maximum transmission rate takes effect
 only if you set this parameter to true.
 "concurrent": 1, // The maximum number of concurrent threads
 .
 "dmu": 1 // The DMU value.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

**Note:**

**Currently, you cannot retrieve data elements from arrays.**

### 2.8.3.3.5 Configure the DB2 reader

The DB2 reader connects to a remote Db2 database and runs SELECT statements to select and read data from the database.

Specifically, the DB2 reader connects to a remote Db2 database over JDBC. Then, it generates SELECT statements based on your configurations and sends the statements to the database. After the database successfully runs the statements, the DB2 reader retrieves the results, formats the results based on the data types defined in the corresponding data integration task, and sends the formatted results to the writer.

- The DB2 reader generates SQL statements based on the table, column, and where parameters, and sends the generated SQL statements to the Db2 database.
- The DB2 reader directly sends the querySql parameter setting to the Db2 database.

The DB2 reader supports most Db2 data types. Since still some of the Db2 data types are not supported, verify that your data types are supported.


The following table lists data types supported by the DB2 reader.

| Data Integration data type | Db2 data type                                                                                  |
|----------------------------|------------------------------------------------------------------------------------------------|
| Integer                    | SMALLINT                                                                                       |
| Floating point             | DECIMAL, REAL, and DOUBLE                                                                      |
| String                     | CHAR, CHARACTER, VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARCHAR, CLOB, LONG VARGRAPHIC, and DBCLOB |
| Date and time              | DATE, TIME, and TIMESTAMP                                                                      |
| Boolean                    | N/A                                                                                            |
| Binary                     | BLOB                                                                                           |

#### Parameters

| Parameter         | Description                                                                                                                                                                                                                          | Required | Default value |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>datasource</b> | The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.                                                                  | Yes      | None          |
| <b>jdbcUrl</b>    | The JDBC connectivity URL, used to connect to the Db2 database. In accordance with Db2 official specifications, the URL format must be jdbc:db2://ip:port/database. You can also specify the information of the attachment facility. | Yes      | None          |
| <b>username</b>   | The username used to connect to the data source.                                                                                                                                                                                     | Yes      | None          |
| <b>password</b>   | The password used to connect to the data source.                                                                                                                                                                                     | Yes      | None          |
| <b>table</b>      | The name of the source table. You can select only one source table for each task.                                                                                                                                                    | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Required | Default value |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>The source table columns to be synchronized. Arrange the column names in a JSON array. The default value is [ * ], which indicates all columns in the source table.</p> <ul style="list-style-type: none"><li>• You can also select some of the columns to synchronize.</li><li>• You can enter the column names in an order that is different from that specified by the schema of the source table.</li><li>• Constants are supported. The column names must be arranged in compliance with SQL syntax supported by MySQL. For example, ["id", "1", "'const name'", "null", "upper('abc_lower')", "2.3", "true"].</li></ul> <ul style="list-style-type: none"><li>- id: the name of a regular column.</li><li>- 1: an integer constant.</li><li>- 'const name': a string constant, which is enclosed in a pair of single quotation marks (').</li><li>- null: a null pointer.</li><li>- upper('abc_lower'): a function expression.</li><li>- 2.3: a floating-point constant.</li><li>- true: a Boolean constant.</li></ul> | Yes      | None          |
| splitPk   | <p>If you specify the splitPk parameter, the table is sharded based on the shard key indicated by this parameter. The DB2 reader then initiates concurrent data synchronization threads, which improves efficiency.</p> <ul style="list-style-type: none"><li>• We recommend that you set the splitPk parameter to the primary key. The table is sharded most evenly if it is sharded based on the primary key.</li><li>• Currently, you can only specify the splitPk parameter to an integer-type column. If you specify this parameter to a column of another type, the DB2 reader returns an error.</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                | No       | Null          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Required | Default value |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| where     | The WHERE clause. The DB2 reader generates SQL statements based on the table and column information and the WHERE clause you have configured, and uses the generated SQL statements for data filtering and reading. For example, for daily incremental synchronization, set this parameter as follows: <code>gmt_create&gt;\$bizdate</code> . The WHERE clause can be used for incremental synchronization. If you leave the WHERE clause unspecified, all data is synchronized. | No       | None          |
| querySql  | The SQL statement used for refined data filtering. If you specify this parameter, the DB2 reader ignores the table, column, and where parameters and uses this parameter for data filtering.<br><br>For example, if you need to join multiple tables for data synchronization, set this parameter to <code>select a ,b from table_a join table_b on table_a.id = table_b.id</code> .                                                                                             | No       | None          |
| fetchSize | The number of data records read per batch. This parameter determines the number of interactions between the reader and the database and affects reading efficiency.<br><br> <b>Note:</b><br>A value larger than 2048 can lead to OOM during the data synchronization process.                                                                                                                 | No       | 1024          |

Configure the DB2 reader in wizard mode

**Currently, wizard mode is not supported for the DB2 reader.**

Configure the DB2 reader in script mode

**In the following script, a task is configured to read data from a Db2 database.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "db2", // The reader type.
 "parameter": {
 "password": "", // The password.

```

```

 "jdbcUrl": "", // The JDBC connectivity URL, used to
connect to the Db2 database.
 "column": [
 "id"
],
 "where": "", // The WHERE clause.
 "splitPk": "", // If you specify the splitPk parameter
, the table is sharded based on the shard key indicated by this
parameter.
 "table": "", // The table name.
 "username": "" // The username.
 },
 "name": "Reader",
 "category": "reader"
},
{ // The following template is used to configure the writer.
For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
}
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

**Note**

- **Data synchronization between primary and secondary databases**

A secondary Db2 database is deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binlogs. Especially when network conditions are unfavorable, data latency between the primary and secondary databases is unavoidable, which can lead to data inconsistency.

- **Concurrency control**

DB2 is a relational database management system (RDBMS), which supports strong consistency for data queries. For example, if a writer is writing data to a Db2 database at the same time as the DB2 reader is reading from it, the DB2 reader cannot read the updated data. This is because of the snapshot isolation feature which enables the transaction of the DB2 reader only to observe a state of the data as when the transaction started.

Data consistency cannot be ensured when you enable the DB2 reader to run concurrent threads in a single data synchronization task. If you specify the `splitPk` parameter, the DB2 reader shards the table based on the `splitPk` parameter and runs multiple concurrent threads for data synchronization. These concurrent threads belong to different transactions, and they read data at different times. This means that the concurrent threads observe different snapshots.

Theoretically, the data inconsistency issue is unavoidable if a single data synchronization task includes multiple threads. However, two workarounds are available.

- Do not enable concurrent threads in a single data synchronization task. Essentially, do not specify the `splitPk` parameter. In this way, data consistency is ensured while data is synchronized at a low efficiency.
- Disable writers to ensure that the data is unchanged during the data synchronization task. For example, lock the table and disable data synchronization between primary and secondary databases. In this way, data is synchronized efficiently but your ongoing services can be interrupted.

- **Character encoding**

The DB2 reader uses JDBC which can automatically convert encoding of characters. Therefore, DB2 Therefore, you do not need to specify the encoding.

- **Incremental synchronization**

The DB2 reader supports incremental synchronization based on `SELECT` statements with `WHERE` clauses.

- For data in batches, incremental add, update, and delete operations (including logically delete operations) are distinguished by timestamps. In this case,

specify the WHERE clause based on the timestamp. The timestamp must be later than the latest timestamp involved in the last synchronization.

- For streaming data, specify the WHERE clause based on the data record ID. The data record ID must be larger than the maximum ID involved in the last synchronization.

If incremental data cannot be distinguished, the DB2 reader cannot perform incremental synchronization but can perform full synchronization only.

- Syntax validation

The DB2 reader allows you to specify custom SELECT statements by using the `querySql` parameter. However, the DB2 reader does not verify the syntax and security of the SELECT statements in the `querySql` parameter.

### 2.8.3.3.6 Configure the MySQL reader

The MySQL reader connects to a remote MySQL database over JDBC. Then, it generates SQL statements based on your configurations and sends the statements to the database. After the database successfully runs the statements, the MySQL reader retrieves the results, formats the results based on the data types defined in the corresponding data synchronization task, and sends the formatted results to the writer.

In short, the MySQL reader connects to a remote MySQL database over JDBC and runs SQL statements to select and read data from the database.

The MySQL reader supports reading data from both tables and views. It can read some or all of the columns in a table or view. You can reorder the columns, create constant columns, and define columns by using MySQL function expressions such as `now()`.

#### Data Types

The following table lists data types supported by the MySQL reader.

| Data Integration data type | MySQL data type                                         |
|----------------------------|---------------------------------------------------------|
| Integer                    | INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT           |
| Floating point             | FLOAT, DOUBLE, and DECIMAL                              |
| String                     | VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT |

| Data Integration data type | MySQL data type                                     |
|----------------------------|-----------------------------------------------------|
| Date and time              | DATE, DATETIME, TIMESTAMP, TIME, and YEAR           |
| Boolean                    | BIT and BOOL                                        |
| Binary                     | TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY |

- Data types that are not listed in the table are not supported.
- The MySQL reader considers tinyint(1) as the integer type.

#### Parameters

| Parameter         | Description                                                                                                                                                         | Required | Default value |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>datasource</b> | The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name. | Yes      | None          |
| <b>table</b>      | The name of the source table. You can select only one source table for each task.                                                                                   | Yes      | None          |



| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Required | Default value |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>The source table columns to be synchronized. Arrange the column names in a JSON array. The default value is [ * ], which indicates all columns in the source table.</p> <ul style="list-style-type: none"><li>• You can also select some of the columns to synchronize.</li><li>• You can enter the column names in an order that is different from that specified by the schema of the source table.</li><li>• Constants are supported. The column names must be arranged in compliance with SQL syntax supported by MySQL. For example, ["id", "table", "1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3", "true"].</li></ul> <ul style="list-style-type: none"><li>- <b>id</b>: the name of a regular column.</li><li>- <b>table</b>: the name of a column that contains reserved keywords.</li><li>- <b>1</b>: an integer constant.</li><li>- <b>'mingya.wmy'</b>: a string constant, which is enclosed in a pair of single quotation marks (').</li><li>- <b>null</b>: a null pointer.</li><li>- <b>CHAR_LENGTH(s)</b>: a function used to calculate the string length.</li><li>- <b>2.3</b>: a floating-point constant.</li><li>- <b>true</b>: a Boolean constant.</li></ul> | Yes      | None          |


| Parameter                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Required | Default value |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| splitPk                                     | <p>If you specify the splitPk parameter, the table is sharded based on the shard key indicated by this parameter. The MySQL reader then initiates concurrent data synchronization threads, which improves efficiency.</p> <ul style="list-style-type: none"><li>• We recommend that you set the splitPk parameter to the primary key. The table is sharded most evenly if it is sharded based on the primary key.</li><li>• Currently, you can only specify the splitPk parameter to an integer-type column. If you specify this parameter to a column of another type, the MySQL reader ignores the splitPk parameter and initiates only one data synchronization thread.</li><li>• If you leave the splitPk parameter unspecified, the MySQL reader initiates only one data synchronization thread for this task.</li></ul> | No       | None          |
| where                                       | <p>The WHERE clause. If you need to synchronize data generated on the current day, set this parameter to <code>gmt_create&gt;\$bizdate</code>.</p> <ul style="list-style-type: none"><li>• The WHERE clause can be used for incremental synchronization. If you leave the WHERE clause unspecified, all data is synchronized.</li><li>• Do not set this parameter to limit 10, which violates the rules of MySQL WHERE clause syntax.</li></ul>                                                                                                                                                                                                                                                                                                                                                                               | No       | None          |
| querySql<br>(only available in script mode) | <p>The SQL statement used for refined data filtering. If you specify this parameter, the MySQL Reader ignores the table, column, where, and splitPk parameters and uses this parameter for data filtering. For example, if you need to join multiple tables for data synchronization, set this parameter to <code>select a,b from table_a join table_b on table_a.id = table_b.id</code>. The data source parses the username and password from this parameter.</p>                                                                                                                                                                                                                                                                                                                                                           | No       | None          |

| Parameter                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Required | Default value |
|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>singleOrMulti</b> (applies only to sharded tables) | Indicates whether the table is sharded. After you switch from wizard mode to script mode, the following configuration is automatically generated: <code>"singleOrMulti": "multi"</code> . However, if you use script mode since the beginning, the configuration is not automatically generated and you need to manually specify this parameter. If you leave this parameter unspecified, the MySQL reader can only read data from the first shard. The <code>singleOrMulti</code> parameter is only used by the frontend, but not by the backend. | Yes      | <b>multi</b>  |

Configure the MySQL reader in wizard mode

### 1. Select data sources.

Configure the source and destination for the data synchronization task.

| Parameter          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Data Source</b> | It refers to the <code>datasource</code> parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.                                                                                                                                                                                                                                                                                                                                                    |
| <b>Table</b>       | It refers to the <code>table</code> parameter provided in the preceding table.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Filter</b>      | The filter that filters out data that does not need to be synchronized. Currently, LIMIT clauses are not supported. The SQL syntax is determined by the selected data source.                                                                                                                                                                                                                                                                                                                                                                |
| <b>Shard Key</b>   | <p>You can specify a column in the source table as the shard key. We recommend that you use the primary key or an indexed column as the shard key.</p> <p>The MySQL reader shards the table based on the shard key and initiates concurrent threads for data synchronization, which improves reading efficiency.</p> <div> <b>Note:</b><br/>Whether the <code>Shard Key</code> parameter is available varies according to the data source selected.</div> |

## 2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

- After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.
- After you click Auto Layout, fields are automatically sorted based on specific rules.
- Change Fields: You can manually edit fields in the source table. Each line indicates a field name. Empty lines are ignored.

The rules for adding fields are described as follows:

- You can enter constants. Each constant must be enclosed in apostrophes ('). For example, 'abc' and '123'.
- You can use relative time parameters, such as \${bizdate}.
- The fields can be functions supported by relational databases, such as now() and count(1).
- Fields that cannot be parsed are indicated by Unidentified.

## 3. Configure the channel.

| Parameter            | Description                                                                                                                                                                                                                                                                 |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                  | The data processing capabilities. A DMU represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.                                                                                   |
| Concurrent Jobs      | The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.                                                                                                                                                       |
| Bandwidth Throttling | Indicates whether to enable bandwidth throttling. You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value. |

| Parameter                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Dirty Data Records Allowed</b> | <b>The maximum number of dirty data records allowed.</b>                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Task Resource Group</b>        | <b>The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers. The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.</b> |

Configure the MySQL reader in script mode

**In the following script, a task is configured to read data from a table that is not sharded.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "mysql", // The reader type.
 "parameter": {
 "column": [// The columns to be synchronized.
 "id"
],
 "connection": [
 {
 "datasource": "", // The data source.
 "table": [// The table name.
 "xxx"
]
 }
],
 "where": "", // The WHERE clause.
 "splitPk": "", // The shard key.
 "encoding": "UTF-8" // The encoding.
 },
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 }
 }
}
```

allowed.

```

 },
 "speed":{
 "throttle":false, // The value false means that the
 bandwidth is not throttled. The value true means that the bandwidth
 is throttled. The maximum transmission rate takes effect only if you
 specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
 threads.
 "dmu":1 // The number of DMUs.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}

```

### 2.8.3.3.7 Configure Oracle Reader

This topic describes the data types and parameters supported by Oracle Reader and how to configure it by using the codeless UI and code editor.

You can use Oracle Reader to read data from Oracle. Oracle Reader connects to a remote Oracle database and runs a SELECT statement to select and read data from the database.

Specifically, Oracle Reader connects to a remote Oracle database through Java Database Connectivity (JDBC), generates a SELECT statement based on your configurations, and sends the statement to the database. The Oracle database runs the statement and returns the result. Then, Oracle Reader assembles the returned data to abstract datasets in custom data types supported by Data Integration, and passes the datasets to a writer.

- Oracle Reader generates the SELECT statement based on the table, column, and where parameters that you have configured, and sends the generated the SELECT statement to the Oracle database.
- If you specify the `querySql` parameter, Oracle Reader directly sends the value of this parameter to the Oracle database.

#### Data types

Oracle Reader supports most Oracle data types. Ensure that your data types are supported.

The following table lists the data types supported by Oracle Reader.

| Category      | Oracle data type                                                                                                                                                                                                  |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integer       | NUMBER, ROWID, INTEGER, INT, and SMALLINT                                                                                                                                                                         |
| Float         | NUMERIC, DECIMAL, FLOAT, DOUBLE PRECISION, and REAL                                                                                                                                                               |
| String        | LONG, CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, CLOB, NCLOB, CHARACTER, CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, and NCHAR VARYING |
| Date and time | TIMESTAMP and DATE                                                                                                                                                                                                |
| Boolean       | BIT and BOOLEAN                                                                                                                                                                                                   |
| Binary        | BLOB, BFILE, RAW, and LONG RAW                                                                                                                                                                                    |


#### Parameters

| Parameter  | Description                                                                                                                | Require | Default value |
|------------|----------------------------------------------------------------------------------------------------------------------------|---------|---------------|
| datasource | The connection name. It must be identical to the name of the added connection. You can add connections in the code editor. | Yes     | None          |
| table      | The name of the table to be synchronized.                                                                                  | Yes     | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Required | Default value |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>An array of columns to be synchronized from the configured table, in JSON format. The default value is [ * ], which indicates all columns.</p> <ul style="list-style-type: none"><li>• Column pruning is supported, which means that you can select and export specific columns.</li><li>• Change of the column order is supported, which means that you can export the columns in an order different from that specified in the schema of the table.</li><li>• Constants are supported. The column names must be arranged in JSON format.</li></ul> <pre>["id", "1", "'mingya.wmy'", "null", "to_char(a + 1)", "2.3", "true"]</pre> <ul style="list-style-type: none"><li>- id: a column name.</li><li>- 1: an integer constant.</li><li>- 'mingya.wmy': a string constant, which is enclosed in a pair of single quotation marks (').</li><li>- null: a null pointer.</li><li>- to_char(a + 1): a function expression.</li><li>- 2.3: a float value.</li><li>- true: a Boolean value.</li></ul> <ul style="list-style-type: none"><li>• The column parameter must be specified.</li></ul> | Yes      | None          |



| Parameter                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Required | Default value |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| splitPk                                         | <p>The field used for data sharding when Oracle Reader extracts data. If you specify the <code>splitPk</code> parameter, the table is sharded based on the shard key specified by this parameter. Data Integration then initiates concurrent data synchronization threads, which improves efficiency.</p> <ul style="list-style-type: none"> <li>• We recommend that you set the <code>splitPk</code> parameter to the primary key of the table. Based on the primary key, data can be well distributed to different shards, but not intensively distributed to certain shards.</li> <li>• The data types supported by <code>splitPk</code> include integer, string, float, and date.</li> <li>• If you do not specify the <code>splitPk</code> parameter or leave it empty, Oracle Reader synchronizes data through a single thread.</li> </ul> | No       | None          |
| where                                           | <p>The WHERE clause. Oracle Reader generates a SELECT statement based on the table, column, and where parameters that you have configured, and uses the generated SELECT statement to select and read data. For example, set this parameter to <code>row_number()</code> during a test. For example, set this parameter to <code>id&gt;2 and sex=1</code>.</p> <ul style="list-style-type: none"> <li>• The WHERE clause can be used for synchronizing incremental data.</li> <li>• If you do not specify the where parameter or leave it empty, all data is synchronized.</li> </ul>                                                                                                                                                                                                                                                            | No       | None          |
| querySql<br>(only available in the code editor) | <p>The SELECT statement used for refined data filtering. If you specify this parameter, Data Integration directly filters data based on this parameter. For example, if you need to join multiple tables for data synchronization, set this parameter to <code>select a,b from table_a join table_b on table_a.id = table_b.id</code>. If you specify the <code>querySql</code> parameter, Oracle Reader ignores the table, column, and where parameters that you have configured.</p>                                                                                                                                                                                                                                                                                                                                                           | No       | None          |


| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                   | Required | Default value |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| fetchSize | <p>The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects reading efficiency.</p> <div> <b>Note:</b><br/>A value larger than 2048 may lead to the out of memory (OOM) error during the data synchronization process.</div> | No       | 1024          |

Configure Oracle Reader by using the codeless UI

### 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.

| Parameter  | Description                                                                                                                                                                       |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection | The <code>datasource</code> parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks. |
| Table      | The <code>table</code> parameter in the preceding parameter description.                                                                                                          |
| Filter     | The filter condition for the data to be synchronized. Currently, filtering based on the limit keyword is not supported. The SQL syntax is determined by the selected data store.  |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shard Key | <p>The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key or an indexed column. Only integer fields are supported.</p> <p>If data sharding is performed based on the configured shard key, data can be read concurrently to improve data synchronization efficiency.</p> <div> <b>Note:</b><br/>The Shard Key parameter is displayed only when you configure the source connection for a data synchronization node.</div> |


2. Configure field mapping (the `column` parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

| Configuration item            | Description                                                                                                                                                      |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Map Fields with the Same Name | Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.                 |
| Map Fields in the Same Line   | Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.                                    |
| Delete All Mappings           | Click Delete All Mappings to remove mappings that have been established.                                                                                         |
| Auto Layout                   | The fields are automatically sorted based on specified rules .                                                                                                   |
| Change Fields                 | You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, while other blank rows are ignored. |

| Configuration item | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add                | <ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as \${bizdate}.</li><li>• You can enter functions supported by relational databases, such as now() and count(1).</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul> |

### 3. Configure the channel.

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                        | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>                                                                                                                                       |
| Concurrent Threads         | <p>The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.</p>                                                                                                                     |
| Bandwidth Throttling       | <p>Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.</p>                                                                                                                  |
| Dirty Data Records Allowed | <p>The maximum number of dirty data records allowed.</p>                                                                                                                                                                                                                                                                                                                                                 |
| Resource Group             | <p>The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance.</p> |

Configure Oracle Reader by using the code editor

In the following code, a node is configured to read data from an Oracle database:

```
{
 "type": "job",
```

```

"version":"2.0",// The version number.
"steps":[
 {
 "stepType":"oracle",
 "parameter":{
 "fetchSize":1024,// The number of data records to read
at a time.
 "datasource":"","// The connection name.
 "column":[// The columns to be synchronized.
 "id",
 "name"
],
 "where":"","// The WHERE clause.
 "splitPk":"","// The shard key.
 "table":"","// The name of the table to be synchronized.
 },
 "name":"Reader",
 "category":"reader"
 },
 {// The following template is used to configure the writer.
For more information, see the document of the corresponding writer.
 "stepType":"stream",
 "parameter":{},
 "name":"Writer",
 "category":"writer"
 }
],
"setting":{
 "errorLimit":{
 "record":"0"// The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "concurrent":1,// The maximum number of concurrent threads
.
 "dmu":1// The DMU value.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
} "to":"Writer"
}
}

```

---

}

#### Additional instructions

- **Data synchronization between primary and secondary databases**

A secondary Oracle database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binlogs. Especially when network conditions are unfavorable, data latency between the primary and secondary databases is unavoidable, which can lead to data inconsistency.

- **Concurrency control**

Oracle is a relational database management system (RDBMS), which supports strong consistency for data queries. A database snapshot is created before a data synchronization node starts. Oracle Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, the reader cannot obtain the new data.

Data consistency cannot be ensured when you enable Oracle Reader to run concurrent threads on a single data synchronization node.

Oracle Reader shards the table based on the `splitPk` parameter and runs multiple concurrent threads to synchronize data. These concurrent threads belong to different transactions, and they read data at different times. This means that the concurrent threads observe different snapshots.

Theoretically, the data inconsistency issue is unavoidable if a single data synchronization node includes multiple threads. However, two workarounds are available:

- **Do not enable concurrent threads on a single data synchronization node.**  
Essentially, do not specify the `splitPk` parameter. In this way, data consistency is ensured while data is synchronized at a low efficiency.
- **Disable writers to ensure that the data is unchanged during data synchronization.** For example, lock the table and disable data synchronization between primary and secondary databases. In this way, data is synchronized efficiently but your ongoing services can be interrupted.

- **Character encoding**

Oracle Reader uses JDBC, which can automatically convert encoding of characters. Therefore, you do not need to specify the encoding.

- **Incremental data synchronization**

Oracle Reader connects to a database through JDBC and uses a `SELECT` statement with a `WHERE` clause to read incremental data in either of the following ways:

- For batch data, incremental add, update, and delete operations (including logical delete operations) are distinguished by timestamps. Specify the `WHERE` clause based on the timestamp. The timestamp must be later than the latest timestamp in the last synchronization.
- For streaming data, specify the `WHERE` clause based on the data record ID. The data record ID must be larger than the maximum ID involved in the last synchronization.

If incremental data cannot be distinguished, Oracle Reader cannot perform incremental synchronization but can perform full synchronization only.

- **Syntax validation**

Oracle Reader allows you to specify custom `SELECT` statements by using the `querySql` parameter but does not verify the syntax of the custom `SELECT` statements.

### 2.8.3.3.8 Configure the OSS reader

The OTS reader connects to the remote Object Storage Service (OSS) server with the official Java SDK. Then, it reads data from the server, converts the data to a format that is readable by the Data Integration service, and sends the formatted data to the writer.

Since OSS servers store unstructured data only, the OSS reader currently supports the following features:

- **TXT files that store logical two-dimensional tables.**
- **Data format similar to CSV with custom delimiters.**
- **Constant columns, and column pruning. It can read various types of data, all stored as strings.**
- **Recursive reading, and file name-based filtering.**

- **File compression options: GZIP, BZIP2, and ZIP.**



**Note:**

**Each compressed package can contain only one file.**

- **Concurrent multi-object reading.**

**The following two features are not supported:**


- **Using multiple concurrent threads to read an uncompressed object.**
- **Using multiple concurrent threads to read a compressed object.**



**The OSS reader supports the following OSS data types: BIGINT, DOUBLE, STRING, DATETIME, and BOOLEAN.**

Parameters

| Parameter         | Description                                                                                                                                                                | Required   | Default value |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------------|
| <b>datasource</b> | <b>The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.</b> | <b>Yes</b> | <b>None</b>   |



| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Required | Default value |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| Object    | <p>The name of the object to be synchronized. You can specify multiple objects. For example, if a bucket has a folder named yunshi and this folder contains a file named ll.txt, then you can specify the Object as yunshi/ll.txt.</p> <ul style="list-style-type: none"><li>• If you specify a single OSS object, the OSS reader uses only one thread for data reading . Concurrent multi-thread reading of a single uncompressed object is coming soon.</li><li>• If you specify multiple OSS objects, the OSS reader uses multiple threads. The actual number of threads is determined by the number of channels.</li><li>• When a name includes a wildcard, the OSS reader attempts to read all objects that match the name.</li></ul> <div> <b>Note:</b><br/>Data Integration considers all the objects in a data synchronization job as a single table. Ensure that all the objects in each data synchronization job can adapt to the same schema, and grant Data Integration the permission to read all these objects.</div> | Yes      | None          |


| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Required | Default value                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------|
| column         | <p>The columns to be read. <b>type</b>: the source data type. <b>index</b>: the ID of the column in the source table, starting from 0. <b>value</b>: the column value if the column is a constant column.</p> <p>To convert all data into the string type, specify this parameter as follows:</p> <pre>json "column": ["*"]</pre> <p>You also can specify the column parameter as follows:</p> <pre>json "column": {   "type": "long",   "index": 0 // The first INT-type column of the source file. }, {   "type": "string",   "value": "alibaba" // The value of the current column is a constant "alibaba". }</pre> <p> <b>Note:</b><br/>If you specify a column, you must specify the type and one of the index and value.</p> | Yes      | The type configuration item defaults to string. |
| fieldDelimiter | <p>The column delimiter.</p> <p> <b>Note:</b><br/>You need to specify the column delimiter for the OSS reader.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Yes      | ,                                               |
| compress       | The compression option. By default, compression is disabled. Available compression options are gzip, bzip2, and zip.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | No       | Compression disabled                            |
| encoding       | The encoding of the file to be read.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | No       | utf-8                                           |

| Parameter              | Description                                                                                                                                                                                                                                                                                                                     | Required | Default value |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>nullFormat</b>      | The string that represents null. Since no standard strings can represent null in text files, Data Integration provides the <b>nullFormat</b> parameter to define which string represents a null pointer. For example, if you specify <code>nullFormat="null"</code> , then Data Integration considers "null" as a null pointer. | No       | None          |
| <b>skipHeader</b>      | Indicates whether to skip the header (if exists) of a CSV-like file. The <b>skipHeader</b> parameter is not supported for compressed files.                                                                                                                                                                                     | No       | false         |
| <b>csvReaderConfig</b> | The configurations for reading CSV-like files. The parameter value must match the Map type. A specific CSV reader is used to read data from CSV-like files, which supports many configurations.                                                                                                                                 | No       | None          |

Configure the OSS reader in wizard mode

### 1. Select data sources.

Configure the source and destination for the data synchronization task.

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Data Source</b>        | It refers to the <b>datasource</b> parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.                                                                                                                                                                                                                   |
| <b>Object Name Prefix</b> | <p>It refers to the <b>Object</b> parameter provided in the preceding table.</p> <div>  <b>Note:</b><br/>           If the OSS files are named based on the date (for example, <code>aaa/20171024abc.txt</code>), you can specify this parameter as <code>aaa/\${bdp.system.bizdate}abc.txt</code>.         </div> |
| <b>Field Delimiter</b>    | It refers to the <b>fieldDelimiter</b> parameter provided in the preceding table, and defaults to a comma (,).                                                                                                                                                                                                                                                                                        |
| <b>Encoding</b>           | It refers to the <b>encoding</b> parameter provided in the preceding table, and defaults to UTF-8.                                                                                                                                                                                                                                                                                                    |
| <b>Null String</b>        | It refers to the <b>nullFormat</b> parameter provided in the preceding table, and defines a string that represents null.                                                                                                                                                                                                                                                                              |

| Parameter          | Description                                                                                |
|--------------------|--------------------------------------------------------------------------------------------|
| Compression Format | It refers to the nullFormat parameter provided in the preceding table, and defaults to No. |
| Include Header     | It refers to the No parameter provided in the preceding table, and defaults to No.         |

2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

- After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.
- After you click Auto Layout, fields are automatically sorted based on specific rules.
- Change Fields: You can manually edit fields in the source table. Each line indicates a field name. Empty lines are ignored.

The rules for adding fields are described as follows:

- You can enter constants. Each constant must be enclosed in apostrophes ('). For example, 'abc' and '123'.
- You can use relative time parameters, such as \${bizdate}.
- The fields can be functions supported by relational databases, such as now() and count(1).
- Fields that cannot be parsed are indicated by Unidentified.

3. Configure the channel.

| Parameter       | Description                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU             | The data processing capabilities. A DMU represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources. |
| Concurrent Jobs | The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.                                                                     |

| Parameter                         | Description                                                                                                                                                                                                                                                                        |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Bandwidth Throttling</b>       | <b>Indicates whether to enable bandwidth throttling. You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.</b> |
| <b>Dirty Data Records Allowed</b> | <b>The maximum number of dirty data records allowed.</b>                                                                                                                                                                                                                           |
| <b>Task Resource Group</b>        | <b>The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.</b>                                             |

Configure the OSS reader in script mode

**In the following script, a task is configured to read data from an OSS data source.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "oss", // The reader type.
 "parameter": {
 "nullFormat": "", // The string that represents null.
 "compress": "", // The compression option.
 "datasource": "", // The data source.
 "column": [// The columns to be synchronized.
 {
 "index": 0, // The ID of the corresponding
source table column.
 "type": "string" // The data type.
 },
 {
 "index": 1,
 "type": "long"
 },
 {
 "index": 2,
 "type": "double"
 },
 {
 "index": 3,
 "type": "boolean"
 },
 {
 "format": "yyyy-MM-dd HH:mm:ss", // The time
format.
 "index": 4,
 "type": "date"
 }
],
 "skipHeader": "", // Indicates whether to skip the
file header. The first row in a CSV-like file can be the header, and
the header need to be skipped.
 }
]
 }
}
```

```

 "encoding": "", // The encoding.
 "fieldDelimiter": ",", // The column delimiter.
 "fileFormat": "", // The format of the file saved by
the OSS reader.
 "object": [] // The object name prefix.
 },
 {
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
"setting": {
 "errorLimit": {
 "record": "" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

### 2.8.3.3.9 Configure FTP Reader

This topic describes the data types and parameters supported by FTP Reader and how to configure it by using the codeless UI and code editor.

FTP Reader allows you to read data from a remote FTP server. Specifically, FTP Reader connects to an FTP server, reads data from the server, converts the data to a format that is readable by Data Integration, and sends the converted data to a writer.

FTP Reader can read only FTP files that store logical two-dimensional tables, for example, text information in the CSV format.

FTP servers store unstructured data only. FTP Reader currently supports the following features:

- Reads TXT files that store logical two-dimensional tables. FTP Reader can read only TXT files.
- Reads data stored in formats similar to CSV with custom delimiters.
- Reads data of various types as strings, and supports constants and column pruning.
- Supports recursive reading and file name-based filtering.
- Supports the following file compression options: GZIP, BZIP2, ZIP, LZO, and LZO\_DEFLATE.
- Reads multiple files concurrently.

The following two features are not supported:


- Uses concurrent threads to read an uncompressed file.
- Uses concurrent threads to read a compressed file.

A remote FTP file does not distinguish between data types. The data types are defined by FTP Reader.


| Data Integration data type | FTP file data type |
|----------------------------|--------------------|
| Long                       | Long               |
| Double                     | Double             |
| String                     | String             |
| Boolean                    | Boolean            |
| Date                       | Date               |

#### Parameters

| Parameter  | Description                                                                                                                | Required | Default value |
|------------|----------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| datasource | The connection name. It must be identical to the name of the added connection. You can add connections in the code editor. | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Required | Default value |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| path      | <p>The path of the FTP file to read. You can specify multiple FTP file paths.</p> <ul style="list-style-type: none"><li>• If you specify a single FTP file, FTP Reader uses only one thread to read the file. Concurrent multi-thread reading of a single uncompressed file is coming soon.</li><li>• If you specify multiple FTP files, FTP Reader uses multiple threads to read these files. The actual number of threads is determined by the number of channels.</li><li>• When a path includes a wildcard, FTP Reader attempts to read all files that match the path. If the path is ended with a slash (/), FTP Reader reads all files in the specified directory. For example, if you specify the path as /bazhen/, FTP Reader reads all files in the bazhen directory. Currently, FTP Reader only supports asterisks (*) as file name wildcards.</li></ul> <div> <b>Note:</b><ul style="list-style-type: none"><li>• We do not recommend that you use asterisks (*) because this may cause Java virtual machine (JVM) memory overflow.</li><li>• Data Integration considers all the files on a data synchronization node as a single table. Ensure that all the files on each data synchronization node can adapt to the same schema, and grant Data Integration the permission to read all these files.</li><li>• Ensure that the data format is similar to CSV.</li><li>• An error occurs if no readable files exist in the specified path.</li></ul></div> | Yes      | None          |



| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Required | Default value                                     |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------------------------------------------|
| column         | <p>The columns to read. The type field specifies the source data type. The index field specifies the ID of the column in the source table, starting from 0. The value field specifies the column value if the column is a constant column.</p> <p>By default, FTP Reader reads all data as strings. Specify this parameter as <code>"column": ["*"]</code>. You can also specify the column parameter as follows:</p> <pre>{   "type": "long",   "index": 0    // The first int-type                column of the source file. }, {   "type": "string",   "value": "alibaba" // The value of                     the current column is a constant "alibaba". }</pre> <p>For the column parameter, you must specify the type field and specify one of the index and value fields.</p> | Yes      | By default, FTP Reader reads all data as strings. |
| fieldDelimiter | <p>The column delimiter.</p> <div>  <b>Note:</b><br/>           You need to specify the column delimiter for FTP Reader. The default delimiter is comma (.). The default setting for the column delimiter on the codeless UI is comma (,), too.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                    | Yes      | ,                                                 |
| skipHeader     | Specifies whether to skip the header (if exists) of a CSV-like file. The skipHeader parameter is not supported for compressed files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | No       | false                                             |
| encoding       | The encoding format of the file to read.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | No       | UTF-8                                             |

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Required | Default value |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| nullFormat      | <p>The string that represents null. No standard strings can represent null in text files. Therefore, Data Integration provides the nullFormat parameter to define which string represents a null pointer.</p> <p>For example, if you specify nullFormat:"null", then Data Integration considers "null" as a null pointer.</p>                                                                                                                                                         | No       | None          |
| markDoneFile    | <p>The name of the file the existence of which indicates that the data synchronization node can start. Data Integration checks whether the file exists before data synchronization. If the file does not exist, Data Integration checks again later. Data Integration starts the data synchronization node only after the file is detected.</p>                                                                                                                                       | No       | None          |
| maxRetryTime    | <p>The maximum number of checks for the file the existence of which indicates that the data synchronization node can start. By default, 60 checks are allowed. Data Integration checks for the file every 1 minute, and the whole process lasts at most 60 minutes.</p>                                                                                                                                                                                                               | No       | 60            |
| csvReaderConfig | <p>The configurations for reading CSV-like files. The parameter value must match the Map type. A specific CSV reader is used to read data from CSV-like files, which supports many configurations.</p>                                                                                                                                                                                                                                                                                | No       | None          |
| fileFormat      | <p>The format of the file saved by FTP Reader. By default, FTP Reader converts the data to a two-dimensional table and stores the table in a CSV file. If you specify this parameter to binary, Data Integration converts data to the binary format for replication and transmission.</p> <p>Generally, you need to specify this parameter only when you want to replicate the complete directory structure between storage systems such as FTP and Object Storage Service (OSS).</p> | No       | None          |

Configure FTP Reader by using the codeless UI

### 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.


| Parameter          | Description                                                                                                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection         | The <code>datasource</code> parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks. |
| File Path          | The <code>path</code> parameter in the preceding parameter description.                                                                                                           |
| File Type          | The format of the file saved by FTP Reader. The default format is CSV.                                                                                                            |
| Field Delimiter    | The <code>fieldDelimiter</code> parameter in the preceding parameter description. The default delimiter is comma (,).                                                             |
| Encoding           | The <code>encoding</code> parameter in the preceding parameter description. The default encoding format is UTF-8.                                                                 |
| Null String        | The <code>nullFormat</code> in the preceding parameter description, which defines a string that represents the null value.                                                        |
| Compression Format | The <code>compress</code> parameter in the preceding parameter description. Files are not compressed by default.                                                                  |
| Include Header     | The <code>skipHeader</code> parameter in the preceding parameter description. The default value is No.                                                                            |

## 2. Configure field mapping (the column parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

| Configuration item            | Description                                                                                                                                      |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Map Fields with the Same Name | Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match. |
| Map Fields in the Same Line   | Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.                    |
| Delete All Mappings           | Click Delete All Mappings to remove mappings that have been established.                                                                         |

## 3. Configure the channel.

| Parameter                  | Description                                                                                                                                                                                                                                                                      |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                        | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>             |
| Concurrent Threads         | The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.    |
| Bandwidth Throttling       | Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value. |
| Dirty Data Records Allowed | The maximum number of dirty data records allowed.                                                                                                                                                                                                                                |

| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource Group | The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance. |

Configure FTP Reader by using the code editor

**In the following code, a node is configured to read data from an FTP server.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "ftp", // The reader type.
 "parameter": {
 "path": [], // The file path.
 "nullFormat": "", // The string that represents null.
 "compress": "", // The compression option.
 "datasource": "", // The connection name.
 "column": [// The columns to be synchronized.
 {
 "index": 0, // The first int-type column of the
 source file.
 "type": "" // The data type.
 }
],
 "skipHeader": "", // Specifies whether to skip the file
 header.
 "fieldDelimiter": ",", // The column delimiter.
 "encoding": "UTF-8", // The encoding.
 "fileFormat": "csv" // The format of the file saved by
 FTP Reader.
 },
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
 For more information, see the document of the corresponding writer.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "throttle": false, // A value of false indicates that
 the bandwidth is not throttled. A value of true indicates that the
```

```
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "concurrent":1,// The maximum number of concurrent threads
.
 "dmu":1// The DMU value.
}
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}
```

### 2.8.3.3.10 Configure the OTS reader

The OTS reader enables reading incremental data from a specific range of Table Store (OTS). The specific range can be:

- An entire table
- Specified ranges
- Specified partitions

Table Store is a NoSQL database service built on the Alibaba Cloud distributed operating system named Apsara. It supports storage and real-time access for large volumes of structured data. Table Store stores data into tables of instances that can seamlessly scale by taking advantage of data partitioning and load balancing.

The OTS reader connects to the remote Table Store server with the official Java SDK . Then, it reads data from the server, converts the data to a format that is readable by the Data Integration service, and sends the formatted data to the writer.

The OTS reader creates tasks as many as the specified number of concurrent threads. Each thread is responsible for running a task.

The OTS reader supports all Table Store data types.

| Data Integration data type | Table Store data type |
|----------------------------|-----------------------|
| Integer                    | INTEGER               |
| Float                      | DOUBLE                |
| String                     | STRING                |
| Boolean                    | BOOLEAN               |
| Binary                     | BINARY                |

**Note:**

Table Store does not support the date type. Therefore, the OTS reader uses long-type Unix timestamps to record the time when errors occur.

## Parameters

| Parameter    | Description                                                                                                                                                                                                                                                                                                                                                      | Required | Default value |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| endpoint     | The endpoint of the Table Store server.                                                                                                                                                                                                                                                                                                                          | Yes      | None          |
| accessId     | The AccessKey ID used to access Table Store.                                                                                                                                                                                                                                                                                                                     | Yes      | None          |
| accessKey    | The AccessKey Secret used to access Table Store.                                                                                                                                                                                                                                                                                                                 | Yes      | None          |
| instanceName | <p>The name of the Table Store instance name.</p> <p>After you enable the Table Store service, you need to create an instance in the console before creating and managing tables.</p> <p>Instances are the basic unit for Table Store resource management. All access control and resource measurement for applications are completed at the instance level.</p> | Yes      | None          |
| table        | The name of the source table. You can set the value to only one table name.                                                                                                                                                                                                                                                                                      | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Required | Default value |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>The source table columns to be synchronized. Arrange the column names in a JSON array. Since Table Store is a NoSQL database service, you must specify the column names.</p> <ul style="list-style-type: none"><li>• Regular columns are supported. For example, {"name":"col1"}.</li><li>• You can also select some of the columns to synchronize. The OTS reader only reads the specified columns.</li><li>• Constant columns are supported. For example , {"type":"STRING", "value":"DataX"}. The type configuration item indicates the constant type . Valid values: STRING, INT, DOUBLE, BOOL, BINARY, INF_MIN, and INF_MAX. If you set the type to Binary, specify the value by using the Base64 encoding. The INF_MIN value indicates the minimum value allowed by Table Store, and the INF_MAX value indicates the maximum value allowed. If you set the type to INF_MIN or INF_MAX, do not specify the value. Otherwise, an error occurs.</li><li>• The OTS reader does not support functions or custom expressions as columns because Table Store does not support such columns.</li></ul> | Yes      | None          |



| Parameter            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Required | Default value |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| <b>begin and end</b> | <p>The table range from which data to be read. You can specify both or neither of these two parameters. The begin and end parameters defines the range of primary keys that correspond to the data in need of synchronization. If you do not need to limit the range, specify the parameters as {"type":"INF_MIN"} and {"type":"INF_MAX"}. For example, if you need to synchronize the data with a primary key of [DeviceID, SellerID], specify the begin and end parameters as follows:</p> <pre> "range": {   "begin": [     {"type":"INF_MIN"}, // Specify the     minimum value of DeviceID.     {"type":"INT", "value":"0"} //     Specify the minimum value of SellerID.   ],   "end": [     {"type":"INF_MAX"}, // Specify the     maximum value of DeviceID.     {"type":"INT", "value":"9999"} //     Specify the maximum value of SellerID.   ] } </pre> <p>Specify the begin and end parameters as follows if you need to synchronize an entire table:</p> <pre> "range": {   "begin": [     {"type":"INF_MIN"}, // Specify the     minimum value of DeviceID.     {"type":"INF_MIN"} // Specify the     minimum value of SellerID.   ],   "end": [     {"type":"INF_MAX"}, // Specify     the maximum value of DeviceID.     {"type":"INF_MAX"} // Specify     the maximum value of SellerID.   ] } </pre> | Yes      | Null          |
| <b>split</b>         | <p>The rules for sharding data. We do not recommend that you specify this parameter.</p> <p>If the Table Store server suffers from a heavy load but the OTS reader cannot automatically shards data, this parameter takes effect.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | No       | None          |
| Issue: 20200116      | The values in the split parameter must fall in the range indicated by the begin and end parameters, and must set to the values in the partition key.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |          | 851           |

Configure the OTS reader in script mode

**In the following script, a task is configured to read data from a Table Store table.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "ots", // The reader type.
 "parameter": {
 "datasource": "", // The data source.
 "column": [// The columns to be synchronized.
 {
 "name": "column1" // The column name.
 },
 {
 "name": "column2"
 },
 {
 "name": "column3"
 },
 {
 "name": "column4"
 },
 {
 "name": "column5"
 }
],
 "range": {
 "split": [
 {
 "type": "INF_MIN"
 },
 {
 "type": "STRING",
 "value": "splitPoint1"
 },
 {
 "type": "STRING",
 "value": "splitPoint2"
 },
 {
 "type": "STRING",
 "value": "splitPoint3"
 },
 {
 "type": "INF_MAX"
 }
],
 "end": [
 {
 "type": "INF_MAX"
 },
 {
 "type": "INF_MAX"
 },
 {
 "type": "STRING",
 "value": "end1"
 },
 {
 "type": "INT",
```

```

 "value":"100"
 }
],
 "begin":[
 {
 "type":"INF_MIN"
 },
 {
 "type":"INF_MIN"
 },
 {
 "type":"STRING",
 "value":"begin1"
 },
 {
 "type":"INT",
 "value":"0"
 }
]
 },
 "table":"" // The table name.
},
"name":"Reader",
"category":"reader"
},
{ // The following template is used to configure the writer.
For more information, see the corresponding section.
 "stepType":"stream",
 "parameter":{},
 "name":"Writer",
 "category":"writer"
}
],
"setting":{
 "errorLimit":{
 "record":"0" // The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
threads.
 "dmu":1 // The number of DMUs.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}

```

```
}
```

### 2.8.3.3.11 Configure PostgreSQL Reader

This topic describes the data types and parameters supported by PostgreSQL Reader and how to configure it by using the codeless UI and code editor.

PostgreSQL Reader connects to a remote PostgreSQL database and runs a **SELECT** statement to select and read data from the database. ApsaraDB for Relational Database Service (RDS) provides the PostgreSQL storage engine.

Specifically, PostgreSQL Reader connects to a remote PostgreSQL database through Java Database Connectivity (JDBC), generates a **SELECT** statement based on your configurations, and sends the statement to the database. The PostgreSQL database runs the statement and returns the result. Then, PostgreSQL Reader assembles the returned data to abstract datasets in custom data types supported by Data Integration, and passes the datasets to a writer.

- PostgreSQL Reader generates the **SELECT** statement based on the `table`, `column`, and `where` parameters that you have configured, and sends the generated SQL statement to the PostgreSQL database.
- If you specify the `querySql` parameter, PostgreSQL Reader directly sends the value of this parameter to the PostgreSQL database.

#### Data types

PostgreSQL Reader supports most PostgreSQL data types. Ensure that your data types are supported.

The following table lists the data types supported by PostgreSQL Reader.

| Category      | PostgreSQL data type                             |
|---------------|--------------------------------------------------|
| Integer       | bigint, bigserial, integer, smallint, and serial |
| Float         | double, precision, money, numeric, and real      |
| String        | varchar, char, text, bit, and inet               |
| Date and time | date, time, and timestamp                        |
| Boolean       | boolean                                          |
| Binary        | bytea                                            |

**Note:**


- Except for the preceding data types, other types are not supported.
- You need to convert the money, inet, and bit types by using syntax such as `a_inet::varchar`.

## Parameters

| Parameter  | Description                                                                                                                | Required | Default value |
|------------|----------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| datasource | The connection name. It must be identical to the name of the added connection. You can add connections in the code editor. | Yes      | None          |
| table      | The name of the table to be synchronized.                                                                                  | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Required | Default value |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| column    | <p>An array of columns to be synchronized from the configured table, in JSON format. The default value is [ * ], which indicates all columns.</p> <ul style="list-style-type: none"><li>• Column pruning is supported, which means that you can select and export specific columns.</li><li>• Change of the column order is supported, which means that you can export the columns in an order different from that specified in the schema of the table.</li><li>• Constants are supported. The column names must be arranged in compliance with SQL syntax supported by MySQL. For example, ["id", "table", "1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3", "true"].<ul style="list-style-type: none"><li>- id: a column name.</li><li>- table: the name of a column that contains reserved keywords.</li><li>- 1: an integer constant.</li><li>- 'mingya.wmy': a string constant, which is enclosed in a pair of single quotation marks (').</li><li>- 'null': a string.</li><li>- to_char(a + 1): a function expression.</li><li>- 2.3: a float value.</li><li>- true: a Boolean value.</li></ul></li><li>• The column parameter must explicitly specify a set of columns to be synchronized. It cannot be left empty.</li></ul> | Yes      | None          |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Required | Default value |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| splitPk   | <p>The field used for data sharding when PostgreSQL Reader extracts data. If you specify the <code>splitPk</code> parameter, the table is sharded based on the shard key specified by this parameter. Data Integration then initiates concurrent data synchronization threads, which improves efficiency.</p> <ul style="list-style-type: none"><li>• We recommend that you set the <code>splitPk</code> parameter to the primary key of the table. Based on the primary key, data can be well distributed to different shards, but not intensively distributed to certain shards.</li><li>• Currently, the <code>splitPk</code> parameter supports data sharding only for integers but not for other data types such as string, float, and date. If you specify this parameter to a column of an unsupported type, PostgreSQL Reader ignores the <code>splitPk</code> parameter and synchronizes data through a single thread.</li><li>• If you do not specify the <code>splitPk</code> parameter or leave it empty, Data Integration synchronizes data through a single thread.</li></ul> | No       | None          |
| where     | <p>The WHERE clause. PostgreSQL Reader generates a SELECT statement based on the <code>table</code>, <code>column</code>, and <code>where</code> parameters that you have configured, and uses the generated SELECT statement to select and read data. For example, set this parameter to <code>id&gt;2 and sex=1</code>.</p> <ul style="list-style-type: none"><li>• The WHERE clause can be used for synchronizing incremental data.</li><li>• If you do not specify the <code>where</code> parameter or leave it empty, all data is synchronized.</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | No       | None          |

| Parameter                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Required | Default value |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| queryString<br>(only available in the code editor) | The SELECT statement used for refined data filtering. If you specify this parameter, Data Integration directly filters data based on this parameter. For example, if you need to join multiple tables for data synchronization, set this parameter to <code>select a,b from table_a join table_b on table_a.id = table_b.id</code> . If you specify the queryString parameter, PostgreSQL Reader ignores the table, column, and where parameters that you have configured. | No       | None          |
| fetchSize                                          | <p>The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects reading efficiency.</p> <div> <b>Note:</b><br/>A value larger than 2048 may lead to the out of memory (OOM) error during the data synchronization process.</div>                                                             | No       | 512           |


Configure PostgreSQL Reader by using the codeless UI

### 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.

| Parameter  | Description                                                                                                                                                                      |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection | The datasource parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks.             |
| Table      | The table parameter in the preceding parameter description.                                                                                                                      |
| Filter     | The filter condition for the data to be synchronized. Currently, filtering based on the limit keyword is not supported. The SQL syntax is determined by the selected data store. |



| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shard Key | <p>The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key or an indexed column. Only integer fields are supported.</p> <p>If data sharding is performed based on the configured shard key, data can be read concurrently to improve data synchronization efficiency.</p> <div> <b>Note:</b><br/>The Shard Key parameter is displayed only when you configure the source connection for a data synchronization node.</div> |


2. Configure field mapping (the column parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

| Configuration item            | Description                                                                                                                                                      |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Map Fields with the Same Name | Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.                 |
| Map Fields in the Same Line   | Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.                                    |
| Delete All Mappings           | Click Delete All Mappings to remove mappings that have been established.                                                                                         |
| Auto Layout                   | The fields are automatically sorted based on specified rules .                                                                                                   |
| Change Fields                 | You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, while other blank rows are ignored. |

| Configuration item | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add                | <ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as \${bizdate}.</li><li>• You can enter functions supported by relational databases, such as now() and count(1).</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul> |

### 3. Configure the channel.

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DMU                        | <p>The billing unit of Data Integration.</p> <div> <b>Note:</b><br/>Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>                                                                                                                                       |
| Concurrent Threads         | <p>The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.</p>                                                                                                                     |
| Bandwidth Throttling       | <p>Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.</p>                                                                                                                  |
| Dirty Data Records Allowed | <p>The maximum number of dirty data records allowed.</p>                                                                                                                                                                                                                                                                                                                                                 |
| Resource Group             | <p>The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance.</p> |

Configure PostgreSQL Reader by using the code editor

**In the following code, a node is configured to read data from a PostgreSQL database.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "postgresql", // The reader type.
 "parameter": {
 "datasource": "", // The connection name.
 "column": [// The columns to be synchronized.
 "col1",
 "col2"
],
 "where": "", // The WHERE clause.
 "splitPk": "", // The shard key based on which the
 table is sharded. Data Integration initiates concurrent threads to
 synchronize data.
 "table": "" // The name of the table to be synchronized.
 },
 "name": "Reader",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
 For more information, see the document of the corresponding writer.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "throttle": false, // A value of false indicates that
 the bandwidth is not throttled. A value of true indicates that the
 bandwidth is throttled. The maximum transmission rate takes effect
 only if you set this parameter to true.
 "concurrent": 1, // The maximum number of concurrent threads
 .
 "dmu": 1 // The DMU value.
 }
 },
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
}
```

---

}

#### Additional instructions

- **Data synchronization between primary and secondary databases**

A secondary PostgreSQL database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binlogs. Especially when network conditions are unfavorable, data latency between the primary and secondary databases is unavoidable, which can lead to data inconsistency.

- **Concurrency control**

PostgreSQL is a relational database management system (RDBMS), which supports strong consistency for data queries. A database snapshot is created before a data synchronization node starts. PostgreSQL Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, the reader cannot obtain the new data.

Data consistency cannot be ensured when you enable PostgreSQL Reader to run concurrent threads on a single data synchronization node.

PostgreSQL Reader shards the table based on the `splitPk` parameter and runs multiple concurrent threads to synchronize data. These concurrent threads belong to different transactions, and they read data at different times. This means that the concurrent threads observe different snapshots.

Theoretically, the data inconsistency issue is unavoidable if a single data synchronization node includes multiple threads. However, two workarounds are available:

- **Do not enable concurrent threads on a single data synchronization node.**  
Essentially, do not specify the `splitPk` parameter. In this way, data consistency is ensured while data is synchronized at a low efficiency.
- **Disable writers to ensure that the data is unchanged during data synchronization.** For example, lock the table and disable data synchronization between primary and secondary databases. In this way, data is synchronized efficiently but your ongoing services can be interrupted.

- **Character encoding**

A PostgreSQL database supports only EUC\_CN and UTF-8 encoding formats for simplified Chinese characters. PostgreSQL Reader uses JDBC, which can automatically convert encoding of characters. Therefore, you do not need to specify the encoding.

If data is written to the PostgreSQL database in an encoding format different from that specified by the PostgreSQL database, PostgreSQL Reader cannot recognize this inconsistency and may export garbled characters.

- **Incremental data synchronization**

PostgreSQL Reader connects to a database through JDBC and uses a `SELECT` statement with a `WHERE` clause to read incremental data in either of the following ways:

- For batch data, incremental add, update, and delete operations (including logical delete operations) are distinguished by timestamps. Specify the `WHERE` clause based on the timestamp. The timestamp must be later than the latest timestamp in the last synchronization.
- For streaming data, specify the `WHERE` clause based on the data record ID. The data record ID must be larger than the maximum ID involved in the last synchronization.

If incremental data cannot be distinguished, PostgreSQL Reader cannot perform incremental synchronization but can perform full synchronization only.

- **Syntax validation**

PostgreSQL Reader allows you to specify custom `SELECT` statements by using the `querySql` parameter but does not verify the syntax of the custom `SELECT` statements.

### 2.8.3.3.12 Configure the LogHub reader

Log Service is an all-in-one real-time data logging service that has been developed by Alibaba Group and tested in many big data scenarios. Log Service supports collection, reading and writing, shipping, search, and analysis of logs, and improves the capacity of processing and analyzing large amounts of logs. The LogHub reader consumes real-time log data in LogHub by using the Java SDK for

**Log Service, converts the data to a format that is readable by the Data Integration service, and sends the converted data to the writer.**

#### Implementation

**The LogHub reader consumes real-time log data in LogHub by using the following version of Java SDK for Log Service:**

```
<dependency>
 <groupId>com.aliyun.openservices</groupId>
 <artifactId>aliyun-log</artifactId>
 <version>0.6.7</version>
</dependency>
```

**In Log Service, a Logstore is a basic unit for collecting, storing, and querying log data. Logstore read and write logs are stored on a shard. Each Logstore consists of several partitions, each of which is defined by a left-closed and right-open interval of MD5. There is no overlapping between intervals. The range of all intervals covers all the allowed MD5 values. Each partition can independently provide some services.**

- **Write: 5 MB/s, 2000 times/s.**
- **Read: 10 MB/s, 100 times/s.**


**The LogHub reader consumes log data in shards by following this process (GetCursor and BatchGetLog APIs):**



- **Obtains a cursor based on the time range.**
- **Reads logs based on the cursor and step parameters and returns the next cursor.**
- **Moves the cursor continuously to consume logs.**
- **Splits and performs concurrent tasks based on shards.**

**The following table lists data types supported by the LogHub reader.**



DataX data type	LogHub data type
String	String

## Parameters

Parameter	Description	Required	Default value
endpoint	The Log Service endpoint, which is a URL for accessing a project and log data. It varies depending on the Alibaba Cloud region where the project resides and the project name.	Yes	None
accessId	The AccessKey ID for accessing Log Service.	Yes	None
accessKey	The AccessKey Secret for accessing Log Service.	Yes	None
project	The name of the project. A project is the basic unit for managing resources in Log Service. You can exercise access control at the project level, and isolate resources among different projects.	Yes	None
logstore	The name of the Logstore. A Logstore is the basic unit for collecting, storing, and querying log data in Log Service.	Yes	None
batchSize	The number of log entries queried from Log Service at a time.	No	128
column	<p>The column name in each log entry. You can configure a column that stores metadata in a source table of LogHub in such a way that the metadata in this column is inserted into the destination table. Supported columns include "C_Topic", "C_MachineUUID", "C_HostName", "C_Path", and "C_LogTime", which indicate the log topic, unique identifier of the machine, host name, path, and log time, respectively.</p> <div> <b>Note:</b> The column name is case insensitive.</div>	Yes	None

Parameter	Description	Require	Default value
<b>BegindateTime</b>	<p>The start time of data consumption in the format of yyyyMMddHHmmss, such as 20180111013000. This parameter defines the left boundary of the left-closed and right-open interval, and can work with the scheduling time parameter in DataWorks.</p> <div> <b>Note:</b> Specify the beginDateTime and endDateTime parameters to determine the time range for consuming data.</div>	You must specify either this parameter or the endTimestampMillis parameter.	Empty string
<b>Enddatetime</b>	<p>The end time of data consumption in the format of yyyyMMddHHmmss, such as 20180111013010. This parameter defines the right boundary of the left-closed and right-open interval, and can work with the scheduling time parameter in DataWorks.</p> <div> <b>Note:</b> Specify the beginDateTime and endDateTime parameters to determine the time range for consuming data.</div>	No	None



Parameter	Description	Require	Default value
<b>beginTimestampMillis</b>	<p>The start time of data consumption, measured in milliseconds. This parameter defines the left boundary of the left-closed and right-open interval.</p> <div>  <b>Note:</b> <ul style="list-style-type: none"> <li>Specify the <b>beginDateTime</b> and <b>endDateTime</b> parameters to determine the time range for consuming data.</li> <li>The value -1 indicates the position where the cursor starts in Log Service (<b>CursorMode.BEGIN</b>).</li> </ul> </div>	You must specify either this parameter or the <b>beginDateTime</b> parameter. We recommend that you specify the <b>beginDateTime</b> parameter.	None
<b>endTimestampMillis</b>	<p>The end time of data consumption, measured in milliseconds. This parameter defines the right boundary of the left-closed and right-open interval.</p> <div>  <b>Note:</b> <ul style="list-style-type: none"> <li>Specify the <b>beginTimestampMillis</b> and <b>endTimestampMillis</b> parameters to determine the time range.</li> <li>The value -1 indicates position where the cursor ends in Log Service (<b>CursorMode.END</b>). We recommend that you specify the <b>beginDateTime</b> parameter.</li> </ul> </div>	You must specify either this parameter or the <b>beginDateTime</b> parameter.	None

Configure the LogHub reader in wizard mode

### 1. Select data sources.

Configure the source and destination for the data synchronization task.

Parameter	Description
Data Source	The datasource parameter provided in the preceding table . Select a data source type, and enter the name of a data source that has been configured in DataWorks.
Logstore	The Logstore name.
Start Time	The start time of data consumption in the format of yyyyMMddHHmmss, such as 20180111013000. This parameter defines the left boundary of the left-closed and right-open interval, and can work with the scheduling time parameter in DataWorks.
End Time	The end time of data consumption in the format of yyyyMMddHHmmss, such as 20180111013010. This parameter defines the right boundary of the left-closed and right-open interval, and can work with the scheduling time parameter in DataWorks.
Number of Entries Read Per Batch	The number of entries queried from the Log Service at a time.

## 2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

- After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.
- After you click Auto Layout, fields are automatically sorted based on specific rules.
- Change Fields: You can manually edit fields in the source table. Each line indicates a field name. Empty lines are ignored.

The rules for adding fields are described as follows:

- You can enter constants. Each constant must be enclosed in apostrophes (''). For example, 'abc' and '123'.
- You can use relative time parameters, such as \${bizdate}.
- The fields can be functions supported by relational databases, such as now() and count(1).
- Fields that cannot be parsed are indicated by Unidentified.

## 3. Control the tunnel.

Parameter	Description
DMU	The data processing capabilities. A data migration unit (DMU) represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.
Concurrent Jobs	The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.
Transmission Rate	You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

Parameter	Description
<b>Task Resource Group</b>	<b>The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.</b>

Configure the LogHub reader in script mode

**The following is a script configuration sample. For more information, see the section "Parameters".**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "loghub", // The reader type.
 "parameter": {
 "datasource": "", // The data source.
 "column": [// The columns to be synchronized.
 "col0",
 "col1",
 "col2",
 "col3",
 "col4",
 "C_topic", // The log topic.
 "C_hostname", // The host name.
 "C_path", // The path.
 "C_logtime" // The log time.
],
 "beginDateTime": "", // The start time of data consumption
 "batchSize": "", // The number of entries that are
 queried from Log Service at a time.
 "endDateTime": "", // The end time of data consumption.
 "fieldDelimiter": ",", // The column delimiter.
 "encoding": "UTF-8", // The encoding.
 "logstore": "///: The name of the target Logstore.
 },
 "name": "Reader ",
 "category": "reader"
 },
 { // The following template is used to configure the writer. For
 more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer ",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the bandwidth
 is not throttled. The value true means that the bandwidth is throttled
 }
 }
}
```

```
. The maximum transmission rate takes effect only if you specify this
parameter as true.
 "concurrent":"1",// The maximum number of concurrent threads.
 "dmu":1// The number of DMUs.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}
```

### 2.8.3.3.13 Configure the OTSReader-Internal reader

**Table Store** (previously known as OTS) is a NoSQL database service built on the **Apsara** distributed system, enabling you to store and access large amounts of structured data in real time. Table Store organizes data into instances and tables that can seamlessly scale by using data partitioning and load balancing.

**OTSReader-Internal** is used to export data for the OTS Internal model, and the OTS reader is used to export data for the OTS Public model.

The OTS Internal model supports multi-version columns, so OTSReader-Internal also provides two modes of exporting data:

- **Multi-version mode:** Table Store supports the storage of multiple versions of columns, and this mode allows you to export data of multiple versions.

This reader converts a cell into a 4-tuple of a one-dimensional table: **PrimaryKey** (columns 1 to 4), **ColumnName**, **Timestamp**, and **Value**. This process is similar to that for the multi-version mode of the HBase reader. Each {primary key, column name, timestamp, value} tuple is sent to the writer as four columns in DataX records.

- **Normal mode:** This mode allows you to export the latest version of each column in each row, which is the same as the normal mode of the HBase reader.

The OTS reader connects to Table Store server and reads data by using the official Java SDK. The OTS reader optimizes the read process by providing features such as performing retry attempts when a timeout or exceptional occurs.

Currently, the OTSReader-Internal reader supports all data types of Table Store.

Data Integration data type	Table Store data type
Long	Integer
Double	Double
String	String
Boolean	Boolean
Bytes	Binary

## Parameters



Parameter	Description	Required	Default value
mode	The mode in which data is read. Valid values: <b>normal</b>   <b>multiVersion</b> .	Yes	None
endpoint	The endpoint of the Table Store server.	Yes	None
accessId	The AccessKey ID for accessing Table Store.	Yes	None
accessKey	The AccessKey Secret for accessing Table Store.	Yes	None
instanceName	The name of the Table Store instance. The Table Store service is managed based on instances.  To create and manage tables after you enable the Table Store service, you must create instances on its console. An instance is also the basic unit for managing Table Store resources. Table Store exercises access control and measures resources at the instance level.	Yes	None
table	The name of the table to be extracted. You can enter only one table name. In Table Store, you do not need to synchronize data among tables.	Yes	None

Parameter	Description	Required	Default value
range	<p>The range of the exported data: [begin,end).</p> <ul style="list-style-type: none"><li>• If the value of the begin parameter is smaller than that for the end parameter, data is read in a positive sequence.</li><li>• If the value of the begin parameter is smaller than that for the end parameter, data is read in an inverted sequence.</li><li>• The value of the begin parameter cannot be equal to that for the end parameter.</li><li>• The following value types are supported: string, integer, and binary. Binary data is passed in as Base64 strings in binary format. INF_MIN represents an infinitely small value and INF_MAX represents an infinitely large value.</li></ul>	No	By default, data is read from the beginning of the table to the end of the table.

Parameter	Description	Required	Default value
range: {"begin": "}"	<p>The starting range of the exported data. Enter an empty array, PK prefix, or complete primary key. When data is read in a positive sequence, the default PK suffix is INF_MIN. When data is read in an inverted sequence, default PK suffix is INF_MAX. An example is described as follows:</p> <p>If your table has two primary keys with the types of string and integer, specify this parameter in any of the following three formats:</p> <ul style="list-style-type: none"> <li>• []: indicates that data is read from the beginning of the table.</li> <li>• [{"type": "string", "value": "a"}]: indicates that data is read from [{"type": "string", "value": "a"}, {"type": "INF_MIN"}].</li> <li>• [{"type": "string", "value": "a"}, {"type": "INF_MIN"}]</li> </ul> <p>The JSON format does not support binary data. For the primary key column with the type of binary, to pass in binary data, you must use the Java Base64.encodeBase64String method to convert binary data into a string, and then enter the string for the value parameter. An example is described as follows (Java):</p> <ul style="list-style-type: none"> <li>• <code>byte[] bytes = "hello".getBytes();</code> Create binary data. The byte value of the hello string is used.</li> <li>• <code>String inputValue = Base64.encodeBase64String(bytes);</code> Use Base64 encoding schemes to convert binary data into a string.</li> </ul> <p>Run the preceding code, and then the string "aGVsbG8=" is returned for the inputValue parameter.</p>	No	Data is read from the beginning of the table.
874	Finally, enter the string for the value parameter: {"type": "binary", "value": "aGVsbG8="}.	Issue: 20200116	



Parameter	Description	Required	Default value
range: {"end"}	<p>The end range of the exported data. Enter an empty array, PK prefix, or complete primary key. When data is read in a positive sequence, the default PK suffix is INF_MIN. When data is read in an inverted sequence, default PK suffix is INF_MAX. An example is described as follows:</p> <p>If your table has two primary keys with the types of string and integer, specify this parameter in any of the following three formats:</p> <ul style="list-style-type: none"> <li>• []: indicates that data is read from the beginning of the table.</li> <li>• [{"type": "string", "value": "a"}]: indicates from [{"type": "string", "value": "a"} to {"type": "INF_MIN"}].</li> <li>• [{"type": "string", "value": "a"}, {"type": "INF_MIN"}].</li> </ul> <p>The JSON format does not support binary data. For the primary key column with the type of binary, to pass in binary data, you must use the Java Base64.encodeBase64String method to convert binary data into a string, and then enter the string for the value parameter. An example is described as follows (Java):</p> <ul style="list-style-type: none"> <li>• <code>byte[] bytes = "hello".getBytes();</code> Create binary data. The byte value of the hello string is used.</li> <li>• <code>String inputValue = Base64.encodeBase64String(bytes);</code> Use Base64 encoding schemes to convert binary data into a string.</li> </ul> <p>Run the preceding code, and then the string "aGVsbG8=" is returned for the inputValue parameter.</p> <p>Finally, enter the string for the value parameter: {"type": "binary", "value": "aGVsbG8="}.</p>	No	Data is read until the end of the table .

Parameter	Description	Require	Default value
range: {"split"}	<p>If an excessively large number of data needs to be exported, you can perform concurrent export tasks. This parameter allows you to split the data in the current range and perform concurrent tasks based on the specified split points.</p> <div>  <b>Note:</b> <ul style="list-style-type: none"> <li>• The value for the split parameter must be the shard key (the first column of PrimaryKey), and the value type must be the same as that of the partition key.</li> <li>• The specified value must fall within the value range of the begin and end parameters.</li> <li>• The values for the split parameter must be sorted in the descending or ascending order based on the data reading sequence that is determined by values of begin and end parameters.</li> </ul> </div>	No	No split point is specified by default.
column	<p>The columns to be exported. Both regular and constant columns can be exported.</p> <p><b>Mode:</b> You can export multiple versions of columns.</p> <p><b>Regular column format:</b> {"name": "{your column name}"}</p>		
timeRange (only applicable to the multi-version mode)	<p>The time range of the requested data: [begin,end).</p> <div>  <b>Note:</b> <p>The value for the begin parameter must be smaller than that for the end parameter.</p> </div>	No	The data of all versions is read by default.

Parameter	Description	Required	Default value
<b>timeRange</b> :{"begin "} (only applicable to the multi-version mode)	The start time of the time range for reading data. Value range: 0 to LONG_MAX.	No	0
<b>timeRange</b> :{"end "} (only applicable to the multi-version mode)	The end time of the time range for reading data. Value range: 0 to LONG_MAX.	No	Long Max(9223372036854775806L)
<b>maxVersion</b> (only applicable to the multi-version mode)	The specified version. Value range: 1 to INT32_MAX.	No	The data of all versions is read by default.

Configure the OTSReader-Internal reader in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the OTSReader-Internal reader in script mode

### Multi-version mode

```
{
 "type": "job",
 "version": "1.0",
 "configuration": {
 "reader": {
 "plugin": "otsreader-internalreader ",
 "parameter": {
 "mode": "multiversion ",
 "endpoint": "",
 "accessId": "",
 "accessKey": "",
 "instanceName": "",
 "table": "<table>",
 "range": {
```

```

 "begin": [
 {
 "type": "string",
 "value": "a"
 },
 {
 "type": "INF_MIN"
 }
],
 "end": [
 {
 "type": "string",
 "value": "g"
 },
 {
 "type": "INF_MAX"
 }
],
 "split": [
 {
 "type": "string",
 "value": "b"
 },
 {
 "type": "string",
 "value": "c"
 }
]
 },
 "column": [
 {
 "name": "attr1"
 }
],
 "timeRange": {
 "begin": 14000000000,
 "end": 16000000000
 },
 "maxVersion": 10
}
},
"writer": {}
}

```

### Normal mode

```

{
 "type": "job",
 "version": "1.0",
 "configuration": {
 "reader": {
 "plugin": "otsreader-internalreader ",
 "parameter": {
 "mode": "normal",
 "endpoint": "",
 "accessId": "",
 "accessKey": "",
 "instanceName": "",
 "table": "<table>",
 "range": {
 "begin": [

```

```
 "type": "string",
 "value": "a"
 },
 {
 "type": "INF_MIN"
 }
],
 "end": [
 {
 "type": "string",
 "value": "g"
 },
 {
 "type": "INF_MAX"
 }
],
 "split": [
 {
 "type": "string",
 "value": "b"
 },
 {
 "type": "string",
 "value": "c"
 }
]
 },
 "column": [
 {
 "name": "pk1"
 },
 {
 "name": "pk2"
 },
 {
 "name": "attr1"
 },
 {
 "type": "string",
 "value": ""
 },
 {
 "type": "int",
 "value": ""
 },
 {
 "type": "double",
 "value": ""
 },
 {
 "type": "binary",
 "value": "aGVsbG8="
 }
]
},
"writer": {}
```

```
}
```

### 2.8.3.3.14 Configure the OTSStream reader

The OTSStream reader is mainly used to exporting the incremental data of Table Store. Incremental data can be considered as operation logs that include data and operation information.

Unlike plug-ins for exporting full data, the OTSStream reader for exporting incremental data only supports the multi-version mode. You cannot export the data of specified columns when using the OTSStream reader for exporting incremental data. The reason is related to the implementation of exporting incremental data. The following section describes the implementation process.

Before using the OTSStream reader, ensure that the Stream feature is enabled. You can enable this feature when creating the table or using the update table API in the SDK.

The method for enabling this feature is described as follows:

```
Syncclient client = new syncclient ("","","","");
Enable this feature when creating the table.
CreateTableRequest createTableRequest = new CreateTableRequest(
 tableMeta);
createTableRequest.setStreamSpecification(new StreamSpecification(
 true, 24)); // The value 24 indicates that the incremental data is
 retained for 24 hours.
client.createTable(createTableRequest);
If this feature is not enabled when the table is created, enable it
with the update table API.
UpdateTableRequest updateTableRequest = new UpdateTableRequest("
 tableName");
updateTableRequest.setStreamSpecification(new StreamSpecification(true
 , 24));
client.updateTable(updateTableRequest);
```

#### Implementation

You can enable the Stream feature and set the expiration time by using the update table API in the SDK. After the Stream feature is enabled, the Table Store server saves your operation logs additionally. Each partition has a sequential operation log queue. Each operation log is removed by garbage collection after the specified expiration time.

The Table Store SDK provides several Stream APIs for reading these operation logs. The OTSStream reader gets incremental data with these APIs, transforms incremental data into multiple 6-tuples (pk, colName, version, colValue, opType, sequenceInfo), and imports them into MaxCompute.

## Exported data format

**In the multi-version mode of Table Store, table data is organized in a three-level architecture: row, column, and version. One row can have multiple columns. The column name is not fixed, and each column can have multiple versions. Each version has a specific timestamp (the version number).**

**You can perform read/write operations by using Table Store APIs. Table Store stores the incremental data by storing the records of recent write and modify operations on table data. Incremental data can be considered as a set of operation records.**

**Table Store supports the following three types of modify operations:**

- **PutRow:** Writes a row. If the row already exists, it is overwritten.
- **UpdateRow:** Updates a row without changing other data of the original row. You can add column values, overwrite column values if the corresponding version of the column already exists, delete all the versions of a column, or delete a version of a column.
- **DeleteRow:** Deletes a row.

**Table Store generates incremental data records based each type of operation. The reader reads these records and exports the data in the format of DataX.**

**Table Store supports the feature of dynamic columns and multi-version mode. Therefore, a row exported by the reader corresponds to a version of a column rather than a row in Table Store. A row in Table Store may correspond to multiple exported rows. Each exported row includes the primary key value, column name, timestamp of the version for the column (version number), value of the version, and operation type. If the isExportSequenceInfo parameter is set to true, time series information is also included.**

**When the data is transformed into the DataX format, four types of operations are defined:**

- **U (UPDATE):** Writes a version of a column.
- **DO (DELETE\_ONE\_VERSION):** Deletes a version of a column.
- **DA (DELETE\_ALL\_VERSION):** Deletes all the versions of a column. Delete all the versions of the corresponding column according to primary key and column name.

- **DR (DELETE\_ROW):** Deletes a row. Delete all the data of the row according to primary key.

In the following example, the table has two primary key columns: pkName1 and pkName2.

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V1	pk2_V1	col_a	1441803688001	col_val1	U
pk1_V1	pk2_V1	col_a	1441803688002	col_val2	U
pk1_V1	pk2_V1	col_b	1441803688003	col_val3	U
pk1_V2	pk2_V2	col_a	1441803688000	N/A	DO
pk1_V2	pk2_V2	col_b	N/A	N/A	DA
pk1_V3	pk2_V3	N/A	N/A	N/A	DR
pk1_V3	pk2_V3	col_a	1441803688005	col_val1	U

In this example, seven rows are exported, corresponding to three rows in the Table Store table. The primary keys for the three rows are (pk1\_V1, pk2\_V1), (pk1\_V2, pk2\_V2), and (pk1\_V3, pk2\_V3).

- For the row whose primary key is (pk1\_V1, pk2\_V1), three operations are included: writing two versions of column col\_a and one version of column col\_b.
- For the row whose primary key is (pk1\_V2, pk2\_V2), two operations are included: deleting one version of column col\_a and deleting all versions of column col\_b.
- For the row whose primary key is (pk1\_V3, pk2\_V3), two operations are included: deleting the row and writing one version of column col\_a.

Currently, the OTSStream reader supports all OTS types. The following table lists data types supported by the OTSStream reader.

Data Integration data type	OTSStream data type
Integer	Integer
Floating point	Double



Data Integration data type	OTSSStream data type
String	String
Boolean	Boolean
Binary	Binary

## Parameters

Parameter	Description	Required	Default value
dataSource	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
dataTable	The name of the table from which the incremental data is exported. You must enable the Stream feature for this table. You can enable the feature during table creation or by using the update table API.	Yes	None

Parameter	Description	Required	Default value
statusTable	<p>The name of the table used by the reader to store status records. These records help to filter out the data that are not covered by the target range and improve export efficiency. A statusTable is the table to store status records. If no such table exists, the reader automatically creates one. When a task of exporting batch data is completed, you do not need to delete the table. The status records in the table can be used for the next export task.</p> <ul style="list-style-type: none"><li>• You do not need to manually create a statusTable. All you need to provide is a table name. The reader attempts to create a statusTable under your instance. If no such table exist, the reader automatically creates one. If such a table already exists, the reader determines whether the Meta of the table is consistent with expectations. If not, an exception is thrown.</li><li>• When an export task is completed, you do not need to delete the table. The statuses of the table can be used for the next export task.</li><li>• The table enables TTL and data expires automatically. Therefore, the data volume is small.</li><li>• You can use a statusTable to store status records of multiple dataTables that are managed by the same instance. The status records are independent of each other.</li></ul> <p>In conclusion, you must configure a name such as TableStoreStreamReaderStatusTable. Note that the name must not be the same as that for any business-related table.</p>	Yes	None

Parameter	Description	Required	Default value
startTimestampMillis	<p>The left border of the time range (left-closed and right-open) of incremental data, measured in milliseconds.</p> <ul style="list-style-type: none"><li>• The reader finds a point corresponding to startTimestampMillis from the statusTable, and starts to read and export data from this point.</li><li>• If the reader cannot find the corresponding point, it starts to read incremental data retained by the system from the first entry, and skip the data which is written later than startTimestampMillis.</li></ul>	No	None
endTimestampMillis	<p>The right boundary of the time range of the incremental data (left-closed and right-open), measured in milliseconds.</p> <ul style="list-style-type: none"><li>• The reader exports data from the time specified by the startTimestampMillis parameter and ends at the data entry with a timestamp that is later than or equal to the endTimestampMillis.</li><li>• If the reader has read all the incremental data, it stops reading data even before the time specified by the endTimestampMillis parameter.</li></ul>	No	None
date	<p>The date when data is exported. The format is yyyyMMdd, for example, 20151111. You must specify this parameter or the startTimestampMillis and endTimestampMillis parameters. For example, Alibaba Cloud Data Process Center performs scheduling only at the day level. Therefore, the date parameter is provided.</p>	No	None
isExportSequenceInfo	<p>Specifies whether to export time-series information. Time-series information includes the time when data is written. The default value is false, indicating that time series information is not exported.</p>	No	None

Parameter	Description	Required	Default value
maxRetries	The maximum number of retries for each request of reading incremental data from Table Store. The default value is 30. Retries are performed at certain intervals. The total time of 30 retries is approximately 5 minutes. Typically, you can keep the default settings.	No	None
startTimeString	The left boundary of the time range (left-closed and right-open) of incremental data, measured in milliseconds in the format of yyyyymmddhh24miss.	No	None
endTimeString	The right boundary of the time range (left-closed and right-open) of incremental data, measured in milliseconds in the format of yyyyymmddhh24miss.	No	None

Configure the OTSStream reader in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the OTSStream reader in script mode

**An example is described as follows. For more information about parameters, see the corresponding section.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "otdsstream", // The plug-in name.
 "parameter": {
 "statusTable": "TableStoreStreamReaderStatusTable", //
The name of the table that stores status records.
 "maxRetries": 30, // The maximum number of retries per
request of reading incremental data from Table Store, which defaults
to 30.
 "isExportSequenceInfo": false, // Specifies whether to
export the time series information.
 "datasource": "${srcdatasource}", // The data source.
 "startTimeString": "${starttime}", // The left
boundary of the time range (left-closed and right-open) for the
incremental data.
 "table": "", // The table name.
 "endTimeString": "${endtime}" // The right boundary
of the time range (left-closed and right-open) for the incremental
data.
 },
 "name": "Reader ",
 "category": "reader"
 },
 { // The following template is used to configure the writer.
For more information, see the corresponding section.
```

```

 "stepType": "stream",
 "parameter": {},
 "name": "Writer ",
 "category": "writer"
 }
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

### 2.8.3.3.15 Configure the RDBMS reader

The relational database management system (RDBMS) reader connects to a remote RDBMS data source and runs SELECT statements to select and read data from the data source. The RDBMS reader is a common reader that enables reading data from DM, DB2, PPAS, or Sybase. If you need the RDBMS reader to read data from a data source, register the driver for the corresponding data source type.

Specifically, the RDBMS reader connects to a remote RDBMS data source over JDBC . Then, it generates SELECT statements based on your configurations and sends the statements to the data source. After the data source successfully runs the statements, the RDBMS reader retrieves the results, formats the results based on the data types defined in the corresponding data integration task, and sends the formatted results to the writer.

The RDBMS reader generates SQL statements based on the table, column, and where parameters, and sends the generated SQL statements to the RDBMS data source. The RDBMS reader directly sends the querySql parameter setting to the RDBMS data source.

**The RDBMS reader supports most common data types such as integer, floating point, and string. Since still some data types are not supported, verify that your data types are supported.**


## Parameters

Parameter	Description	Required	Default value
jdbcUrl	<p>The JDBC connectivity URL, used to connect to the data source. The format must be in accordance with official specifications. You can also specify the information of the attachment facility. The format varies with the data source type. The RDBMS selects an appropriate driver for data reading based on the format.</p> <ul style="list-style-type: none"> <li>• Format for DM data sources: jdbc:dm://ip:port/database</li> <li>• Format for DB2 data sources: jdbc:db2://ip:port/database</li> <li>• Format for PPAS data sources: jdbc:edb://ip:port/database</li> </ul> <p>You can enable the RDBMS reader to support a new data source type by using the following methods:</p> <ul style="list-style-type: none"> <li>• Switch to the directory of the RDBMS reader, \${DATAX_HOME}/plugin/reader/rdbmsreader. \${DATAX_HOME} indicates the main directory of DataX.</li> <li>• Add the driver of your database to the drivers array in the plugin.json file of the plugin.json directory. The RDBMS reader automatically selects an appropriate driver for connecting to a database.</li> </ul> <pre>{   "name": "rdbmsreader",   "class": "com.alibaba.datax.plugin.reader.rdbmsreader.RdbmsReader",   "description": "useScene: prod. mechanism : Jdbc connection using the database, execute select sql, retrieve data from the ResultSet . warn: The more you know about the database , the less problems you encounter.",   "developer": "alibaba",   "drivers": [     "dm.jdbc.driver.DmDriver",     "com.ibm.db2.jcc.DB2Driver",     "com.sybase.jdbc3.jdbc.SybDriver",     "com.edb.Driver"   ] },..</pre> <p>- Add the package of the driver to the libs subdirectory of the rdbmsreader directory.</p> <pre>tree ├── libs │   └── Dm7JdbcDriver16.jar</pre>	Yes	None

Parameter	Description	Required	Default value
password	The password used to connect to the data source.	Yes	None
table	The name of the source table.	Yes	None
column	<p>The source table columns to be synchronized. Arrange the column names in a JSON array. The default value is [ * ], which indicates all columns in the source table.</p> <ul style="list-style-type: none"> <li>You can also select some of the columns to synchronize.</li> <li>You can enter the column names in an order that is different from that specified by the schema of the source table.</li> <li>Constants are supported. Use the JSON format. For example, ["id", "1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3", "true"]. <ul style="list-style-type: none"> <li>id: The name of a regular column.</li> <li>1: An integer constant.</li> <li>'bazhen.csy': A string constant.</li> <li>null: A null pointer.</li> <li>to_char(a + 1): A function expression.</li> <li>2.3: A floating-point constant.</li> <li>true: A Boolean constant.</li> </ul> </li> </ul>	Yes	None



Parameter	Description	Required	Default value
<b>splitPk</b>	<p>If you specify the <b>splitPk</b> parameter, the table is sharded based on the shard key indicated by this parameter. The RDBMS reader then initiates concurrent data synchronization threads, which improves efficiency.</p> <ul style="list-style-type: none"><li>• We recommend that you set the <b>splitPk</b> parameter to the primary key. The table is sharded most evenly if it is sharded based on the primary key.</li><li>• Currently, you can only specify the <b>splitPk</b> parameter to an integer-type column. If you specify this parameter to a column of another type, the RDBMS reader returns an error.</li><li>• If you do not specify this parameter, the table is not sharded and the RDBMS reader synchronizes all data with only one thread.</li></ul>	No	An empty string
<b>where</b>	<p>The WHERE clause. The RDBMS reader generates SQL statements based on the table and column information and WHERE clauses you have configured, and uses the generated SQL statements for data filtering and reading. For example, set this parameter to limit 10 during a test. If you need to synchronize data generated on the current day, set this parameter to <code>gmt_create&gt;\$bizdate</code>.</p> <ul style="list-style-type: none"><li>• The WHERE clause can be used to incremental synchronization.</li><li>• If you leave the WHERE clause unspecified, all data is synchronized.</li></ul>	No	None

Parameter	Description	Required	Default value
querySql	<p>The SQL statement used for refined data filtering. If you specify this parameter, the RDBMS reader ignores the table, column, and where parameters and uses this parameter for data filtering.</p> <p>For example, if you need to join multiple tables for data synchronization, set this parameter to <code>select a ,b from table_a join table_b on table_a.id = table_b.id.</code></p>	No	None
fetchSize	<p>The number of data records read per batch. This parameter determines the number of interactions between the reader and the database and affects reading efficiency.</p> <div>  <b>Note:</b>  A value larger than 2048 can lead to OOM during the data synchronization process. </div>	No	1024

Configure the RDBMS reader in wizard mode

**Currently, wizard mode is not supported for the RDBMS reader.**

Configure the RDBMS reader in script mode

**In the following script, a task is configured to write data to an RDBMS data source.**

```
{
 "job": {
 "setting": {
 "speed": {
 "byte": 1048576
 },
 "errorLimit": {
 "record": 0,
 "percentage": 0.02
 }
 },
 "content": [
 {
 "reader": {
 "name": "rdbmsreader",
 "parameter": {
 "username": "xxx",
 "password": "xxx",
 "column": [
 "id",
 "name"
]
 }
 }
]
]
 }
}
```

```
 "splitPk": "pk",
 "connection": [
 {
 "table": [
 "table"
],
 "jdbcUrl": [
 "jdbc:dm://ip:port/database"
]
 }
],
 "fetchSize": 1024,
 "where": "1 = 1"
 },
 "writer": {
 "name": "streamwriter",
 "parameter": {
 "print": true
 }
 }
}
]
```

In the following script, a task is configured to synchronize data from a custom SQL data source to a MaxCompute data source.

```
{
 "job": {
 "setting": {
 "speed": {
 "byte": 1048576
 },
 "errorLimit": {
 "record": 0,
 "percentage": 0.02
 }
 },
 "content": [
 {
 "reader": {
 "name": "rdbmsreader",
 "parameter": {
 "username": "xxx",
 "password": "xxx",
 "column": [
 "id",
 "name"
],
 "splitPk": "pk",
 "connection": [
 {
 "querySql": [
 "SELECT * from dual"
],
 "jdbcUrl": [
 "jdbc:dm://ip:port/database"
]
 }
]
 }
 }
]
]
 }
}
```

```
 "fetchSize": 1024,
 "where": "1 = 1"
 },
 },
 "writer": {
 "name": "streamwriter",
 "parameter": {
 "print": true
 }
 }
}
]
}
}
```

### 2.8.3.3.16 Configuring the StreamCompute reader

The StreamCompute reader automatically generates data from the memory. It is mainly used for performance testing for data synchronization and basic functional testing.

The following table lists data types supported by the StreamCompute reader.

Data type	Description
STRING	A sequence of characters.
LONG	A 64-bit two's complement integer.
DATE	A value that represents dates.
BOOL	A Boolean data type that has one of two possible values.
BYTES	An 8-bit signed two's complement integer.

## Parameters

Parameter	Description	Required	Default value
column	<p>The column data and type of the source data. Multiple columns can be configured. You can set to generate random strings and specify the range. The example is as follows:</p> <pre>"column" : [   {     "random": "8,15"   },   {     "random": "10,10"   } ]</pre> <p>The parameters in the example are described as follows:</p> <ul style="list-style-type: none"><li>• "random": "8,15": generates a random string that is 8 to 15 bytes in length.</li><li>• "random": "10,10": generates a 10-byte random string.</li></ul>	Yes	None
sliceRecordCount	The number of columns generated repeatedly.	Yes	None

Configure the StreamCompute reader in wizard mode

**The StreamCompute reader cannot be configured in wizard mode.**

Configure the StreamCompute reader in script mode

**In the following script, a task is configured to read data from a StreamCompute data source.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "stream", // The reader type.
 "parameter": {
 "column": [// The fields.
 {
 "type": "string", // The data type.
 "value": "field" // The value.
 },
 {

```

```

 "type": "long",
 "value": 100
 },
 {
 "dateFormat": "yyyy-MM-dd HH:mm:ss", // The
format of the time.
 "type": "date",
 "value": "2014-12-12 12:12:12"
 },
 {
 "type": "bool",
 "value": true
 },
 {
 "type": "bytes",
 "value": "byte string"
 }
],
 "sliceRecordCount": "100000" // The number of columns
repeatedly generated.
},
 "name": "Reader",
 "category": "reader"
},
 { // The following template is used to configure the writer.
For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // false: The bandwidth is not throttled
. true: The bandwidth is throttled. The maximum transmission rate
takes effect only if you specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
 },
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
}

```

}

### 2.8.3.3.17 Configure Elasticsearch Reader

This topic describes the working principles, features, and parameters of Elasticsearch Reader.

#### Working principles

- **Elasticsearch Reader reads data from Elasticsearch by slicing scroll queries. The slices are processed by multiple threads of a data synchronization node.**
- **Data types are converted based on the mapping configuration of Elasticsearch.**

#### Basic settings

```
{
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 },
 "setting":{
 "errorLimit":{
 "record":"0" // The maximum number of dirty data records
allowed.
 },
 "jvmOption":"",
 "speed":{
 "concurrent":3,
 "throttle":false
 }
 },
 "steps":[
 {
 "category":"reader",
 "name":"Reader",
 "parameter":{
 "column":[// The fields to read.
 "id",
 "name"
],
 "endpoint":"", // The endpoint.
 "index":"", // The index name.
 "password":"", // The password.
 "scroll":"", // The scroll ID.
 "search":"", // The search criteria. The value is the
same as the Elasticsearch query that uses the _search API.
 "type":"default",
 "username":"" // The username.
 },
 "stepType":"elasticsearch"
 },
 {
 "category":"writer",
 "name":"Writer",
 "parameter":{ },

```

```
 "stepType":"stream"
 }
],
 "type":"job",
 "version":"2.0" // The version number.
 }
```

#### Advanced features

- **Supports storing all data of an Elasticsearch document in one column.**

**You can create a column to store all data of an Elasticsearch document.**

- **Supports converting semi-structured data to structured data.**

Item	Description
Background	Data in Elasticsearch is deeply nested. Elasticsearch may contain fields of various types and lengths and may use Chinese names. To facilitate data computing and storage in downstream businesses, Elasticsearch Reader supports converting semi-structured data to structured data.
Principle	Elasticsearch Reader flattens nested JSON data obtained from Elasticsearch to single-dimensional data based on the paths of properties in the JSON data. Then, Elasticsearch Reader maps the single-dimensional data to structured tables. In this way, Elasticsearch data in a complex structure is converted to multiple structured tables.



Item	Description
<b>Solution</b>	<ul style="list-style-type: none"> <li>- Elasticsearch Reader converts nested JSON data to single-dimensional data by using the following path formats: <ul style="list-style-type: none"> <li>■ Property</li> <li>■ Property.Child property</li> <li>■ Property[0].Child property</li> </ul> </li> <li>- If a property has multiple child properties, Elasticsearch Reader traverses all data of the property and splits the data to multiple tables or multiple rows in the following format: <p>Property[*].Child property</p> </li> <li>- Elasticsearch Reader merges data in a string array to one property in the following format and removes duplicates: <p>Property[] where duplicates are removed</p> </li> <li>- Elasticsearch Reader merges multiple properties to one property in the following format: <p>Property 1,Property 2</p> </li> <li>- Elasticsearch Reader presents optional properties in the following format: <p>Property 1 Property 2</p> </li> </ul>

#### Parameters

Parameter	Description	Required	Default value
endpoint	The endpoint of Elasticsearch.	Yes	None
username	The username for HTTP authentication.	No	Empty string
password	The password for HTTP authentication.	No	Empty string
index	The index name in Elasticsearch.	Yes	None

Parameter	Description	Required	Default value
type	The type name in the index of Elasticsearch.	No	Index name
pageSize	The number of data records to read at a time.	No	100
search	The query parameter of Elasticsearch.	Yes	None
scroll	The scroll parameter of Elasticsearch, which sets the timestamp of the snapshot taken for a scroll.	Yes	None
sort	The field based on which the returned results are sorted.	No	None
retryCount	The number of retries after a failure.	No	300
connTimeOut	The connection timeout of the client.	No	600,000
readTimeOut	The data reading timeout of the client.	No	600,000
multiThread	Specifies whether to use multiple threads for an HTTP request.	No	true
column	The fields to read.	Yes	None
full	Specifies whether to create a column to record all data of an Elasticsearch document.	No	false

Parameter	Description	Required	Default value
multi	<b>Specifies whether to split an array to multiple rows . If you enable this feature, you need to specify additional settings.</b>	No	false

**Additional settings:**

```
"full":false,
 "multi": {
 "multi": true,
 "key":"crn_list[*]"
 }
```

## 2.8.3.4 Configure the writer

### 2.8.3.4.1 Configure the DataHub writer

DataHub is a real-time data distribution platform designed to process streaming data. You can publish and subscribe applications to streaming data in DataHub and distribute the data to other platforms. This allows you to easily analyze streaming data and build applications based on the streaming data.

Based on the Apsara platform of Alibaba Cloud, DataHub features high availability, low latency, high scalability, and high throughput. Seamlessly integrated with StreamCompute, the stream computing engine of Alibaba Cloud, DataHub allows you to easily use SQL to analyze streaming data. DataHub also supports synchronizing streaming data to various Alibaba Cloud services such as MaxCompute and OSS.

**Note:**

Strings can only be UTF-8 encoded. The size of each string must not exceed 1 MB.


**Channel type**

The source is connected to the sink through a single channel, and then to the destination. Therefore, the channel type configured for the writer must be the same as that configured for the reader. Currently, two channel types are available:

memory channel and file channel. The following example shows how to set the channel type to file.

```
"agent.sinks.dataXSinkWrapper.channel": "file"
```

#### Parameters

Parameter	Description	Required	Default value
<b>accessId</b>	The AccessKey ID for accessing DataHub.	Yes	None
<b>accessKey</b>	The AccessKey Secret for accessing DataHub.	Yes	None
<b>endpoint</b>	The endpoint of DataHub.	Yes	None
<b>maxRetryCount</b>	The maximum number of retries if a task fails.	No	None
<b>mode</b>	The mode for writing strings.	Yes	None
<b>parseContent</b>	The data that has been parsed.	Yes	None
<b>project</b>	<p>A project is an organizational unit in DataHub, which contains one or more topics.</p> <div>  <b>Note:</b>            DataHub projects are independent from MaxCompute projects. Projects that you created in MaxCompute cannot be used in DataHub.         </div>	Yes	None
<b>topic</b>	Topic is the smallest unit for data subscription and publication. You can use topics to distinguish different types of streaming data.	Yes	None
<b>maxCommitSize</b>	The amount of data, in MB, the DataHub writer buffers before sending it to the destination. This mechanism aims to improve writing efficiency.	No	1 MB
<b>batchSize</b>	The number of data records that the DataHub writer buffers before sending them to the destination. This mechanism aims to improve writing efficiency.	No	1024
<b>maxCommitInterval</b>	The maximum interval at which the DataHub writer sends data to the destination. When an interval ends, the DataHub writer sends buffered data even if the data amount does not reach the above two thresholds.	No	30000

Parameter	Description	Required	Default value
parseMode	The method of parsing log entries. The value default indicates that no log parsing is required. The value csv indicates that a delimiter is inserted between fields for each log entry.	No	default

Configure the DataHub writer in wizard mode

**Currently, wizard mode is not supported for the DataHub writer.**

Configure the DataHub writer in script mode

**In the following script, a task is configured to write data to a DataHub data source.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "datahub", // The writer type.
 "parameter": {
 "datasource": "", // The data source.
 "topic": "", // Topic is the smallest unit for data
 subscription and publication. You can use topics to distinguish
 different types of streaming data.
 "maxRetryCount": 500, // The maximum number of retries
 if a task fails.
 "maxCommitSize": 1048576 // The amount of data, in MB
 , that the DataHub writer buffers before sending it to the destination
 .
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "concurrent": 20, // The maximum number of concurrent
 threads.
 "throttle": false, // The value false means that the
 bandwidth is not throttled. The value true means that the bandwidth
 is throttled. The maximum transmission rate takes effect only if you
 specify this parameter as true.
 "dmu": 20 // The number of DMUs.
 }
 }
}
```

```
 },
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
 }
}
```

#### 2.8.3.4.2 Configure the DB2 writer

The DB2 writer enables writing data to tables stored on Db2 databases. To write data into a Db2 table, the DB2 writer connects to the remote Db2 database through JDBC, and runs `INSERT INTO` statements. Data is written into the Db2 table in batches.

The DB2 writer is designed for ETL developers to import data from data warehouses to Db2 databases. It also serves as a data migration tool for database administrators and other users.

The DB2 writer reads data from the channel, connects to a remote Db2 database through JDBC, and then runs `INSERT INTO` statements. The rows that violate the unique index constraint or primary key constraint cannot be written into the Db2 database. To improve performance, the DB2 writer makes batch updates with the `PreparedStatement` method and sets `rewriteBatchedStatements=true`. In this way, the DB2 writer buffers data, and submits a write request when the amount of data in the buffer reaches a specific threshold.



##### Note:

The `INSERT INTO` privilege is required for data synchronization tasks with the DB2 writer. Whether other permissions are required depends on the SQL statements specified in the `preSql` and `postSql` parameters.

The DB2 writer supports most Db2 data types. Since still some of the Db2 data types are not supported, verify that your data types are supported.

The following table lists data types supported by the DB2 writer.

Data Integration data type	Db2 data type
Integer	SMALLINT

Data Integration data type	Db2 data type
Floating point	DECIMAL, REAL, and DOUBLE
String	CHAR, CHARACTER, VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARCHAR, CLOB, LONG VARGRAPHIC, and DBCLOB
Date and time	DATE, TIME, and TIMESTAMP
Boolean	N/A
Binary	BLOB

## Parameters

Parameter	Description	Require	Default value
<b>jdbcUrl</b>	The JDBC connectivity URL, used to connect to the Db2 database. In accordance with Db2 official specifications, the URL format must be <b>jdbc:db2://ip:port/database</b> . You can also specify the information of the attachment facility.	Yes	None
<b>username</b>	The username used to connect to the data source.	Yes	None
<b>password</b>	The password used to connect to the data source.	Yes	None
<b>table</b>	The name of the destination table.	Yes	None
<b>column</b>	The columns in the destination table to which data is written. Separate the columns with a comma (.). Example: "column": ["id", "name", "age"]. Set the value to an asterisk (*) if data is written to all the columns in the destination table. Example: "column": ["*"].	Yes	None
<b>preSql</b>	The SQL statement runs before the data synchronization task starts. Currently, you can run only one SQL statement. For example, you can run a statement to clear outdated data.	No	None
<b>postSql</b>	The SQL statement runs after the data synchronization task ends. Currently, you can run only one SQL statement in wizard mode but multiple SQL statements in script mode. For example, you can run a statement to add a timestamp.	No	None

Parameter	Description	Required	Default value
batchSize	The number of data records to write per batch. Setting this parameter can greatly reduce the interactions between Data Integration and the Db2 database over the network, and increase the throughput. However, an excessively large value may cause the running Data Integration process to become out of memory (OOM).	No	1024

Configure the DB2 writer in wizard mode

**Currently, wizard mode is not supported for the DB2 writer.**

Configure the DB2 writer in script mode

**In the following script, a task is configured to write data to a Db2 database.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "db2", // The writer type.
 "parameter": {
 "postSql": [], // The SQL statement runs after the data
 synchronization task ends.
 "password": "", // The password.
 "jdbcUrl": "jdbc:db2://ip:port/database", //The JDBC
 connectivity URL, used to connect to the Db2 database.
 "column": [
 "id"
],
 "batchSize": 1024, // The number of data records to
 write per batch.
 "table": "", // The table name.
 "username": "", // The username.
 "preSql": [] // The SQL statement runs before the data
 synchronization task starts.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 }
 },
}
```



```

 "speed":{
 "throttle":false, // The value false means that the
 bandwidth is not throttled. The value true means that the bandwidth
 is throttled. The maximum transmission rate takes effect only if you
 specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
 threads.
 "dmu":1 // The number of DMUs.
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
 }
}

```

### 2.8.3.4.3 Configure the FTP writer

The FTP writer allows you to write one or more files in CSV format into a remote FTP file. At the underlying level, this writer converts the data that is readable by the Data Integration service to CSV files, and writes these files into the remote FTP server using FTP network protocols. You must configure the data source before configuring the FTP writer.



#### Note:

For more information, see [Add FTP data sources](#).

The FTP writer can only write data into FTP files that store logical two-dimensional tables, for example, text information in the CSV format.

This writer enables you to convert data that is readable by the Data Integration service to FTP files. FTP files stores non-structured data. The advantages and disadvantages of the FTP writer are described as follows:

- Only supports text files and the schema in the text file must be a two-dimensional table. It does not support the blob type, such as video data.
- Supports CSV and text files with custom delimiters.
- Does not support text compression when data is written to the destination table.
- Supports multi-thread writing, with each thread performing write operations on a subfile.

Currently, the FTP writer does not support the following two features:

- Concurrent writing for a single file.

- Providing varying data types. The FTP does not provide data types, and the FTP writer writes data of the string type into FTP files.

#### Parameters

Parameter	Description	Required.	Default value
<b>datasource</b>	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
<b>timeout</b>	The timeout period for the connection to the FTP server, measured in milliseconds.	No	60000 (1 minute)
<b>path</b>	The path of the FTP file system. The write can write data into multiple files in the path.	Yes	None
<b>FileName</b>	The name of the file into which data is written. A random suffix is added to the file name to form the actual name of the file into which the data is written on each thread.	Yes	None
<b>writeMode</b>	<p>The mode in which the FTP writer clears existing data before writing data. Valid values:</p> <ul style="list-style-type: none"> <li>· <b>truncate</b>: The writer clears all the files prefixed by fileName in the path before writing data.</li> <li>· <b>append</b>: No processing is performed on the file before the FTP writer imports data into this file. In the Data Integration service, the FTP writer uses the original file name in the data source. No duplicate file names are allowed.</li> <li>· <b>nonConflict</b>: An error is reported if a file prefixed by fileName exists in the path.</li> </ul>	Yes	None
<b>fieldDelimiter</b>	The column delimiter of the file to be written.	Yes. A single character is used.	None
<b>compress</b>	The compress option. The gzip and bzip2 compression options are supported.	No	No

Parameter	Description	Required.	Default value
encoding	The encoding of the file to be read.	No	UTF-8
nullFormat	The string that represents null. Since no standard strings can represent null in text files, Data Integration provides the nullFormat parameter to define which string represents a null pointer.  For example, if you specify nullFormat: "null", Data Integration considers "null" as a null pointer.	No	None
dateFormat	The date format, for example, "dateFormat": "yyyy-MM-dd".	No	None
fileFormat	The file format, including CSV and text. For the CSV format, if you want to write the data that includes column delimiters, the delimiters are escaped with quotation marks. For text format, the data to be written is separated by column delimiters without being escaped.	No	text
header	The header used when a txt file is written, for example, ['id', 'name', 'age'].	No	None
Markdonefilename	The name of the file marked as "done". After a synchronization task is completed, a MarkDoneFile is generated, based on which you can determine whether the task is executed successfully.	No	None

Configure the FTP writer in wizard mode

### 1. Select data sources.

Configure the source and destination for the data synchronization task.

Parameter	Description
Data Source	The datasource parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.
File Path	The path parameter provided in the preceding table.

Parameter	Description
Column Delimiter	The fieldDelimiter parameter provided in the preceding table. Default value: a comma (,)
Encoding	The encoding parameter provided in the preceding table. Default value: UTF-8.
Null String	The nullFormat parameter provided in the preceding table , which defines a string that represents null.
Compression Format	The nullFormat parameter provided in the preceding table . Default value: No.
Include Header	The skipHeader parameter in the preceding table. Default value: No.
Prefix Conflict	The writeMode parameter provided in the preceding table, which defines a string that represents null.

2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.

3. Configure the channel.

Parameter	Description
DMU	The data processing capabilities. A data migration unit ( DMU) represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.
Concurrent Jobs	The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.
Transmission Rate	You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.

Parameter	Description
<b>Dirty Data Records Allowed</b>	<b>The maximum number of dirty data records allowed.</b>
<b>Task Resource Group</b>	<b>The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.</b>

Configure the FTP writer in script mode

**In the following script, a task is configured to write data to an FTP database.**

```
{
 "type":"job",
 "version":"2.0",// The version number.
 "steps":[
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType":"stream",
 "Parameter ":{},
 "name":"Reader",
 "category":"reader"
 },
 {
 "stepType":"ftp",// The plug-in name.
 "parameter":{
 "path":"","// The file path.
 "fileName": "",// The file name.
 "nullFormat": "null", // The string that represents
null.
 "dateFormat":"yyyy-MM-dd HH:mm:ss", // The time format
.
 "datasource": "", // The data source.
 "writeMode": "",// The writing method.
 "fieldDelimiter": ",", // The column delimiter.
 "encoding":"","// The encoding.
 "fileFormat": "",// The file type.
 },
 "name":"Writer",
 "category":"writer"
 }
],
 "setting":{
 "errorLimit":{
 "record":"0"// The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false,// The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent":"1",// The maximum number of concurrent
threads.
 "dmu":1// The number of DMUs.
 }
 },
 "order":{
```

```
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
}
```

#### 2.8.3.4.4 Configure HBase Writer

This topic describes the features, data types, and parameters supported by HBase Writer and how to configure it by using the code editor.

HBase Writer allows you to write data to HBase data stores. Specifically, HBase Writer connects to a remote HBase data store through the Java client of HBase. Then, HBase Writer uses the PUT method to write data to the HBase data store.

##### Features

- HBase 0.94.x and 1.1.x are supported.
  - If you use HBase 0.94.x, set the `hbaseVersion` parameter to `094x` for the writer.

```
"writer": {
 "hbaseVersion": "094x"
}
```

- If you use HBase 1.1.x, set the `hbaseVersion` parameter to `11x` for the writer.

```
"writer": {
 "hbaseVersion": "11x"
}
```



##### Note:

Currently, HBase Writer for HBase 1.1.x is compatible with HBase 2.0. If you have any issues in using HBase Writer with HBase 2.0, open a ticket.

- You can use concatenated fields as a rowkey.

Currently, HBase Writer supports concatenating multiple fields to generate the rowkey of an HBase table.

- You can use any of the following information as the version of each HBase cell.

The information that can be used as the version of an HBase cell includes:

- Current time
- Specified source column
- Specified time

## Data types

The following table lists the data types supported by HBase Writer.

**Note:**

- The types of the specified columns must be the same as those in the HBase table.
- Except for the data types listed in the following table, other types are not supported.

Category	HBase data type
Integer	Int, Long, and Short
Float	Float and Double
Boolean	Boolean
String	String

## Parameters

Parameter	Description	Required	Default value
haveKerberos	<p><b>Specifies whether Kerberos authentication is required. It is required if you specify haveKerberos as true.</b></p> <div> <b>Note:</b><ul style="list-style-type: none"><li>• If this value is true, the following five Kerberos-related parameters must be specified:<ul style="list-style-type: none"><li>- kerberosKeytabFilePath</li><li>- kerberosPrincipal</li><li>- hbaseMasterKerberosPrincipal</li><li>- hbaseRegionserverKerberosPrincipal</li><li>- hbaseRpcProtection</li></ul></li><li>• If the value is false, Kerberos authentication is not required and you do not need to specify the preceding parameters.</li></ul></div>	No	false

Parameter	Description	Required	Default value
hbaseConfig	The properties of the HBase cluster, in JSON format. This parameter must contain the <code>hbase.zookeeper.quorum</code> configuration option, which indicates the ZooKeeper ensemble servers. It can also contain optional configuration options such as those related to the cache and batch for scan operations.	Yes	None
mode	The mode in which data is written to the HBase data store. Currently, only the normal mode is supported. The dynamic column selection mode is coming soon.	Yes	None
table	The name of the HBase table to which data is written. The name is case sensitive.	Yes	None
encoding	The encoding, to which a string is converted through <code>byte[]</code> . Currently, UTF-8 and GBK are supported.	No	utf-8
column	The HBase columns to which data is written. <ul style="list-style-type: none"> <li>• <b>index</b>: the ID of the corresponding source table column, starting from 0.</li> <li>• <b>name</b>: the name of the column in the HBase table, in the <code>columnFamily:column</code> format.</li> <li>• <b>type</b>: the type of the data written, which is used by the <code>byte[]</code> constructor.</li> </ul>	Yes	None
maxVersion	The number of versions read by HBase Reader when multiple versions are available. Valid values: -1 and integers greater than 1. The value -1 indicates that all versions are read.	Required in multiVersionFixedColumn mode	None



Parameter	Description	Required	Default value
range	<p><b>The rowkey range that HBase Reader reads.</b></p> <ul style="list-style-type: none"> <li>• <b>startRowkey: the start rowkey.</b></li> <li>• <b>endRowkey: the end rowkey.</b></li> <li>• <b>isBinaryRowkey: the operation called by byte[] to convert the specified start and end rowkeys. Default value: false. If the value is true, Bytes.toBytesBinary(rowkey) is called. If the value is false, Bytes.toBytes(rowkey) is called.</b></li> </ul> <p><b>Example:</b></p> <pre>"range": {   "startRowkey": "aaa",   "endRowkey": "ccc",   "isBinaryRowkey": false }</pre> <p><b>Example:</b></p> <pre>"column": [   {     "index": 1,     "name": "cf1:q1",     "type": "string"   },   {     "index": 2,     "name": "cf1:q2",     "type": "string"   } ]</pre>	No	None
rowkeyColumn	<p><b>The rowkey of each HBase cell.</b></p> <ul style="list-style-type: none"> <li>• <b>index: the ID of the corresponding source table column, starting from 0. If the column is a constant, set the value to -1.</b></li> <li>• <b>type: the type of the data written, which is used by the byte[] constructor.</b></li> <li>• <b>value: a constant, which is usually used as the delimiter between fields. HBase Writer sequentially concatenates all columns specified in this parameter to a string, and uses the string as the rowkey. The specified columns cannot be all constants.</b></li> </ul> <p><b>Example:</b></p> <pre>"rowkeyColumn": [   {     "index": 0,     "type": "string"   },   {     "index": 1,     "type": "string"   } ]</pre>	Yes	None

Parameter	Description	Required	Default value
walFlag	<b>Specifies whether to enable write ahead logging (WAL) for HBase. If the value is true, all edits requested by an HBase client for all Regions carried by the RegionServer are recorded first in the WAL (that is, the HLog). After the edits are successfully recorded in the WAL, they are implemented to the Memstore and a success indication is sent to the HBase client. If edits fail to be recorded in the WAL, a failure indication is sent to the HBase client without implementing the edits. If the value is false, WAL is disabled but writing efficiency is improved.</b>	No	false
writeBufferSize	<b>The write buffer size, in bytes, of the HBase client. If you specify this parameter, you must also specify the autoflush parameter.</b>  <b>autoflush:</b> <ul style="list-style-type: none"><li>• If the value is true, the HBase client sends a PUT request each time it receives an edit.</li><li>• If the value is false, the HBase client sends a PUT request only when its write buffer is full.</li></ul>	No	8M

Configure HBase Writer by using the codeless UI

**Currently, the codeless UI is not supported for HBase Writer.**

Configure HBase Writer by using the code editor

**In the following code, a node is configured to write data to an HBase 1.1.x data store.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "hbase", // The writer type.
 "parameter": {
```

```

"mode":"normal",// The mode in which data is written
to the HBase data store.
"walFlag":"false",// WAL is disabled for HBase.
"hbaseVersion":"094x",// The HBase version.
"rowkeyColumn":[// The rowkey of each HBase cell.
{
 "index":"0",// The ID of the corresponding
source table column.
 "type":"string",// The data type.
},
{
 "index":"-1",
 "type":"string",
 "value":"_"
}
],
"nullMode":"skip",// The method of processing null
values.
"column":[// The HBase columns to which data is
written.
{
 "name":"columnFamilyName1:columnName1",// The
name of the HBase column.
 "index":"0",// The ID of the corresponding
source table column.
 "type":"string",// The data type.
},
{
 "name":"columnFamilyName2:columnName2",
 "index":"1",
 "type":"string"
},
{
 "name":"columnFamilyName3:columnName3",
 "index":"2",
 "type":"string"
}
],
"writeMode":"api",// The write mode.
"encoding":"utf-8",// The encoding.
"table":"","// The name of the table to be synchronized
.
"hbaseConfig":{"// The properties of the HBase cluster
, in JSON format.
 "hbase.zookeeper.quorum":"hostname",
 "hbase.rootdir":"hdfs://ip:port/database",
 "hbase.cluster.distributed":"true"
}
},
"name":"Writer",
"category":"writer"
},
],
"setting":{
 "errorLimit":{
 "record":"0",// The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 }
}

```

```
 "concurrent":1,// The maximum number of concurrent threads
 },
 "dmu":1// The DMU value.
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
}
```

### 2.8.3.4.5 Configure the HBase11xsq writer

The HBase11xsq writer allows you to batch import data into an SQL table (Phoenix ) in HBase. Phoenix encodes the rowkey using a specific method. Therefore, you need to manually convert the data when you use HBase APIs to write data. The conversion process requires lots of efforts and makes it easy to make mistakes. This writer enables you to separately import data into SQL tables.

It runs the UPSERT statement at the underlying level to write data into HBase based on the JDBC drive of Phoenix.

#### Features

**This writer supports importing tables with indexes and simultaneously updating all index tables.**

#### Restrictions



**The restrictions of the HBase11xsq writer are described as follows:**

- Supports only 1.x versions of Hbase.
- Supports only HBase tables that are created using Phoenix.
- Does not support importing data that is assigned with timestamps.

#### Implementation

**To batch import data into HBase, this writer runs the UPSERT statement through the JDBC drive of Phoenix. An upper-layer API enables the simultaneous update of indexed tables.**

## Parameters

Parameter	Description	Required	Default value
plugin	The name of this plug-in, which must be <code>hbase11xsq1</code> .	Yes	None
table	The name of the table to be imported. The name is case sensitive, and the name for each Phoenix table contains only uppercase letters in most cases.	Yes	None
column	<p>The name of the column, which is case sensitive. The column names of Phoenix tables contain only uppercase letters in most cases.</p> <div> <b>Note:</b><ul style="list-style-type: none"><li>• The column sequence must be consistent with the sequence of output columns from the reader.</li><li>• You do not need to specify the data type, and the column metadata is automatically retrieved from Phoenix.</li></ul></div>	Yes	None
hbaseConfig	<p>The IP address of the HBase cluster, in the format of <code>ip1,ip2,ip3</code>. The <code>zk</code> parameter is required.</p> <div> <b>Note:</b><ul style="list-style-type: none"><li>• Separate multiple IP addresses with commas (,).</li><li>• The <code>znode</code> parameter is optional, and its default value is <code>/hbase</code>.</li></ul></div>	Yes	None
batchSize	The number of rows to write per batch.	No	256

Parameter	Description	Required	Default value
<b>nullMode</b>	<p>The processing mode when the column value is null. The valid values for this parameter are as follows:</p> <ul style="list-style-type: none"><li>• <b>skip</b>: Skip this column. In this mode, the column is not inserted. If the column of the row already exists, the column is deleted.</li><li>• <b>empty</b>: Insert 0 or an empty string. In this mode, 0 is inserted for a numeric value , and an empty string is inserted for a varchar value.</li></ul>	No	skip

Configure the HBase11xsql writer in script mode

**An example is described as follows:**

```
{
 "type": "job",
 "version": "1.0",
 "configuration": {
 "setting": {
 "errorLimit": {
 "record": "0"
 },
 "speed": {
 "mbps": "1",
 "concurrent": "1"
 }
 },
 "reader": {
 "plugin": "odps",
 "parameter": {
 "datasource": "",
 "table": "",
 "column": [],
 "partition": ""
 }
 },
 "plugin": "hbase11xsql",
 "parameter": {
 "table": "The name of the target HBase table, which is case sensitive",
 "hbaseConfig": {
 "hbase.zookeeper.quorum": "The IP address of the ZooKeeper server for the target HBase cluster. Contact the PE for more information.",
 "zookeeper.znode.parent": "The znode of the target HBase cluster. Contact the PE for more information."
 },
 "column": [
 "columnName"
],
 },
 }
}
```

```
 "batchSize": 256,
 "nullMode": "skip"
 }
}
```

## Restrictions

The sequence of columns in the reader determines how columns for each row are organized. The column sequence in the writer must be consistent with that in the reader. The sequence of columns in the writer describes how the writer expects the received data in columns for each row to be organized. For example:

If the column sequence in the reader is c1, c2, c3, c4, and the column sequence in the writer is x1, x2, x3, x4, column c1 from the reader corresponds to column x1 in the writer. If the column sequence in the writer is x1, x2, x4, x3, columns c3 and c4 from the reader correspond to columns x4 and x3, respectively.

## FAQ

**Q: What is the proper number of concurrent threads? Can I increase the number of concurrent threads to improve the efficiency of importing data?**

**A: The default JVM heap size is 2 GB during the data import process. Concurrent tasks are executed by multiple threads. An excessively large number of threads may not improve the import efficiency, and may compromise the performance due to frequent garbage collection (GC). We recommend that you set the number of concurrent threads to 5 to 10.**

**Q: What is the proper value for the batchSize parameter?**

**A: The default value is 256. Set the batchSize parameter based on the data volume in each row. Typically, the data volume for each operation is about 2 MB to 4 MB. Divide this value by the data volume in the row, and set the batchSize parameter accordingly.**

### 2.8.3.4.6 Configure the HDFS writer

The HDFS writer is used to write TextFile, ORCFile, and ParquetFile to the specified path to HDFS. The files can be associated with Hive tables. You must configure the data source before configuring the HDFS Writer plug-in. For more information, see [Add HDFS data sources](#).

## Implementation

The implementation process for HDFS writer is shown below.

1. Create a temporary directory that does not exist in HDFS based on the path you specified.

Naming rule: path\_random

2. Write the files that have been read to this temporary directory.
3. When all the files are written to the temporary directory, move these files to the directory you specified. The file names should be unique.
4. Delete the temporary directory. If you are unable to connect to HDFS for reasons such as network interruption during the process, delete the temporary directory and the files written to it manually.



### Note:

For data synchronization, admin account and read/write permissions for the files are required.

### Usage:

- Create an admin user and home directory, specify a user group and additional group, and grant the permissions for the files.

```
useradd -m -G supergroup -g hadoop -p admin admin
```

- `-G supergroup`: Specifies the additional group to which the user belongs.
- `-g hadoop`: Specifies the user group to which the user belongs.
- `-p admin admin`: Add a password to the admin user.

- View the contents of the files in this directory.

```
hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
```

When using hadoop commands, the format is `hadoop fs -command`, where `command` represents the command.



- **Copies the file part-00000 to the local file system.**

```
hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
```

- **Edit the file you just copied.**

```
vim part-00000
```

- **Exits the current user.**

```
exit
```

- **Connect to the host from the list and create an admin account on each attached host.**

```
pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -g hadoop -p admin admin
```

- `pssh -h /home/hadoop/slave4pssh`: **connect to the host from the manifest file.**
- `useradd -m -G supergroup -g hadoop -p admin admin`: **Create admin account.**

#### Restrictions

- It only supports TextFile, ORCFile, and ParquetFile formats, and what is stored in the file must be a two-dimensional table in a logic sense.
- HDFS is a file system and has no schema. Therefore, it does not support writing data into specified columns.
- Only the following Hive data types are supported:
  - Numeric: TINYINT, SMALLINT, INT, BIGINT, FLOAT, and DOUBLE
  - String: STRING, VARCHAR, and CHAR
  - Boolean: BOOLEAN
  - Time type: date, timestamp
- Hive data types such as decimal, binary, arrays, maps, ovens, and union are not currently supported.
- For Hive partitioned tables, the data can only be written to one partition at a time.
- For the TextFile format, ensure the delimiters in the files written to HDFS are identical to the ones used in the tables created in Hive, so that the data written to HDFS is associated with the Hive table fields.

- In the current plug-in, the Hive version is 1.1.1, and the Hadoop version is 2.7.1. Apache is compatible with JDK1.7. Data can be written normally in the testing environments of Hadoop 2.5.0, Hadoop 2.6.0, and Hive 1.2.0. For other versions, further test is needed.

#### Data type conversion

Currently, HDFS Writer supports most data types in Hive. Check whether the Hive type you are using is supported.


HDFS Writer converts the data types in Hive as follows:

Data Integration data type	HDFS/Hive data type
Long	TINYINT, SMALLINT, INT, and BIGINT
Double	FLOAT,DOUBLE
String	STRING, VARCHAR, and CHAR
Boolean	Boolean
Date	DATE and TIMESTAMP


#### Parameters

Parameter	Description	Required	Default value
defaultFS	The namenode address in Hadoop HDFS, for example, <code>hdfs://127.0.0.1:9000</code> . The default resource group does not support the configuration of the advanced Hadoop parameter HA.	Yes	None
fileType	The file type. Currently, only text, orc, and parquet are supported. <ul style="list-style-type: none"><li>· text: Indicates TextFile.</li><li>· orc: Indicates ORCFile.</li><li>· parquet: Indicates ParquetFile.</li></ul>	Yes	None

Parameter	Description	Required	Default value
path	<p>The path under which the files are written to Hadoop HDFS. The HDFS writer writes multiple files under the path based on the concurrent writing configurations.</p> <p>For association with a Hive table, enter the path under the Hive table stored in HDFS. For example, if the path to the data warehouse set in Hive is <code>/user/hive/warehouse/</code> and you have created the database test table named hello, the path of the Hive table is <code>/user/hive/warehouse/test.db/hello</code>.</p>	Yes	None
FileName	The name of the file written by HDFS Writer. A random suffix is appended to the file name to form the actual name of the file written using each thread.	Yes	None

Parameter	Description	Required	Default value
column	<p>The fields of the written data. You cannot write data into specified columns.</p> <p>For association with a Hive table, you must specify all the field names and types in the table, with name and type specifying the field name and field type respectively.</p> <p>You can configure the column field as follows:</p> <pre>"column": [   {     "name": "userName",     "type": "string",   },   {     "name": "age",     "type": "long"   } ]</pre>	Yes. If the filetype is parquet, it is optional.	None
writeMode	<p>The mode in which HDFS Writer clears the existing data before data writing:</p> <ul style="list-style-type: none"> <li>• <b>append:</b> No processing is performed on the file before the HDFS writer imports data into this file. In the Data Integration service, the HDFS writer uses the original file name in the data source. No duplicate file names are allowed.</li> <li>• <b>nonConflict:</b> An error is reported if a file prefixed by fileName exists in the path.</li> </ul> <p> <b>Note:</b> Parquet files only support the nonConflict mode, and does not support the Append mode.</p>	Yes	None
fieldDelimiter	<p>The field delimiter used for the fields written by HDFS Writer. Ensure the field delimiter is identical to the one used in the Hive table created. Otherwise, you are unable to locate the data in the Hive table.</p>	Yes. If the filetype is parquet, it is optional.	None

Parameter	Description	Required	Default value
<b>compress</b>	<b>The compression option of HDFS files. It is left empty by default, which means no compression is performed.</b>  <b>Text files support gzip and bzip2 compression types. Orc files support SNAPPY compression. SnappyCodec is needed.</b>	<b>No</b>	<b>None</b>
<b>encoding</b>	<b>The encoding of the file to be written.</b>	<b>No</b>	<b>No compression</b>

Parameter	Description	Required	Default value
parquetSchema	<p>Required when the file is in parquet format. It is used to specify the structure of the target file, and takes effect only when the fileType is parquet. The format is as follows:</p> <pre>message MessageType {   Required, data type, column name;   .....; ; }</pre> <p>The configuration items are as follows:</p> <ul style="list-style-type: none"> <li>• <b>MessageType:</b> Any supported value</li> <li>• <b>Required:</b> Required or Optional. Optional is recommended.</li> <li>• <b>Data Type:</b> Parquet files support the following data types: boolean, int32, int64, int96, float, double, binary (select binary if the data type is string), and fixed_len_byte_array.</li> </ul> <p> <b>Note:</b> Each configuration row and column, including the last one, must end with a semicolon.</p> <p><b>Example:</b></p> <pre>message m {   optional int64 id;   optional int64 date_id;   optional binary datetimestring;   optional int32 dspId;   optional int32 advertiserId;   optional int32 status;   optional int64 bidding_req_num;   optional int64 imp;   optional int64 click_num; }</pre>	No	None

Configure the HDFS writer in wizard mode

**Development in wizard mode is not supported currently.**

Configure the HDFS writer in script mode

**An example is described as follows. For more information about parameters, see the corresponding section.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Reader ",
 "category": "reader"
 },
 {
 "stepType": "hdfs", // The plug-in name.
 "parameter": {
 "path": "", // The path information stored to Hadoop
HDFS File System.
 "fileName": "", // The file name when the HDFS writer
write data into the file.
 "compress": "", // The HDFS file compression option.
 "datasource": "", // The name of the data source.
 "column": [
 {
 "name": "col1", // The field name.
 "type": "string" // The field Type.
 },
 {
 "name": "col2",
 "type": "int"
 },
 {
 "name": "col3",
 "type": "double"
 },
 {
 "name": "col4",
 "type": "boolean"
 },
 {
 "name": "col5",
 "type": "date",
 }
],
 "writeMode": "", // The writing method.
 "fieldDelimiter": ",", // The column delimiter.
 "encoding": "", // The encoding.
 "fileType": "text" // text type
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "" // The maximum number of dirty data records
allowed.
 },
 "speed": {
```

```
 "concurrent":3, // The maximum number of concurrent
threads.
 "throttle":false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "dmu":1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}
```

### 2.8.3.4.7 Configure the MaxCompute writer

The MaxCompute writer is designed for ETL developers to insert or update data in MaxCompute. It allows you to import business data into MaxCompute and is suitable for TB and GB-level data transmission.



**Note:**

You must configure the data source before configuring the MaxCompute writer. For more information, see [Add MaxCompute data sources](#).

In terms of underlying implementation, it writes data into MaxCompute by using Tunnel based on the source project/table/partition/table field and other information you configured.

#### Data types

The following table lists data types supported by the MaxCompute writer.

Data Integration data type	MaxCompute data type
Integer	Bigint
Floating point	Double and decimal
String	String
Date and time	Datetime
Boolean	Boolean



## Parameters

Parameter	Description	Required	Default value
<b>datasource</b>	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
<b>table</b>	The name of the destination table . The name is case insensitive. Specify only one destination table.	Yes	None
<b>partition</b>	<p>The partition information of the destination table. Specify the parameter for each level. For example, if you want to write data into a table with three partition levels, specify this parameter for each level, for example, pt=20150101, type=1, biz=2.</p> <ul style="list-style-type: none"> <li>For non-partitioned tables, do not specify this parameter, indicating that data is directly imported into the destination table.</li> <li>The MaxCompute writer does not support writing data via routing. For a table with multiple partition levels, specify this parameter for each level.</li> </ul>	Required only for partitioned tables	None

Parameter	Description	Required	Default value
column	<p>The list of fields to be imported. If you want to import all the fields, enter <code>"column": ["*"]</code>. If you want to insert some columns, enter these columns, for example, <code>"column": ["id", "name"]</code>.</p> <ul style="list-style-type: none"><li>• The MaxCompute writer enables you to filter and switch columns. For example, a table includes three fields: a, b, and c. To synchronize data of fields c and b, specify this parameter to <code>"column": ["c, b"]</code>. In this case, the value null is automatically entered for field a.</li><li>• You must specify the columns to be imported.</li></ul>	Yes	None

Parameter	Description	Required	Default value
truncate	<p><b>The TRUNCATE TABLE operation.</b></p> <p><b>"truncate": "true" is configured to ensure the idempotence of write operations. When a retry is made after a failed write attempt, the MaxCompute writer cleans up existing data and imports new data. This ensures the data is consistent after each rerunning.</b></p> <p><b>The TRUNCATE TABLE statement is not an atomic operation. This is because MaxCompute SQL is used for data cleansing and SQL cannot achieve atomicity. Therefore, when multiple tasks are performed to clean up a table or partition at the same time, high concurrency may lead to potential problems.</b></p> <p><b>To prevent such problems, we do not recommend that you operate on one partition with multiple job DDLs at the same time. We recommend that you create partitions before starting multiple concurrent jobs.</b></p>	Yes	None

Configure the MaxCompute writer in wizard mode

### 1. Select data sources.

Configure the source and destination for the data synchronization task.

Parameter	Description
Data Source	The datasource parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.
Table	The table parameter provided in the preceding table. Select the table to be synchronized.
Partition	<p>To synchronize all columns in the table, set "column": [""]. The partition parameter supports wildcards and includes one or more partitions.</p> <ul style="list-style-type: none"><li>• "partition": "pt=20140501/ds=*" indicates that all partitions in the ds partition need to be synchronized.</li><li>• "partition": "pt=top?" indicates that the partitions with pt=top and pt=to need to be synchronized.</li></ul> <p>You can specify the partition columns to be synchronized, such as a partition column named pt. For example, assume that the partition column of a MaxCompute source table is pt=\${bdp.system.bizdate}. You can configure the pt column to be synchronized. Ignore it if the column is marked as unidentified. To synchronize all partitions, set pt=\${*}. To synchronize some of the partitions, specify the corresponding dates.</p>
Clearance Rule	<ul style="list-style-type: none"><li>• Clear Existing Data Before Writing (Insert Overwrite): All data in the table or partition is cleaned up before import.</li><li>• Retain Existing Data (Insert Into): No data is cleared before data importing. New data is always appended with each run.</li></ul>
Compression	No compression

Parameter	Description
Specify Empty String as Null	Yes

2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

- After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.
- After you click Auto Layout, fields are automatically sorted based on specific rules.

3. Configure the channel.

Parameter	Description
DMU	The data processing capabilities. A DMU represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.
Concurrent Jobs	The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.
Transmission Rate	Indicates whether to enable bandwidth throttling. You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Task Resource Group	The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.

Configure the MaxCompute writer in script mode

**An example is described as follows. For more information about parameters, see the corresponding section.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following script configures the Stream reader. Modify
 the script according to the real destination. For more information,
 see the corresponding section.
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "odps", // The plug-in name.
 "parameter": {
 "partition": "", // The partition information.
 "truncate": true, // The writing method.
 "compress": false, // The compression option.
 "datasource": "odps_first", // The data source name.
 "column": [// The column name.
 "*"
],
 "emptyAsNull": false, // Indicates whether to specify
an empty string as null?
 "table": "" // The table name.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // Indicates whether to enable bandwidth
throttling
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
 },
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
}
```

```
}
```

#### Additional information

- **Column filtering**

MaxCompute does not support column filtering, reordering, and null filling, but the MaxCompute writer does. For example, to import all fields in a list, set `"column": ["*"]`.

Assume that a MaxCompute table has three fields: a, b, and c. To synchronize data of fields c and b, set `"column": ["c", "b"]`. In this case, fields c and b in the MaxCompute table are imported into the first and second columns of the table in the reader. In the table of the reader, field a is set to null.

- **How to handle column configuration errors**

To ensure data is written in a reliable manner, data loss from redundant columns must be prevented to guarantee data quality. The MaxCompute writer reports an error if redundant columns are imported. For example, if a MaxCompute table has fields a, b, and c, but more than three fields are written into the destination table, the MaxCompute Writer produces an error.

- **Partition configuration**

**MaxCompute** The MaxCompute writer only supports writing data by specifying each partition level, and does not support partition routing of writing based on a specific field. To write data into a table with three partition levels, specify the partition parameter for each level, for example, `pt=20150101, type=1, biz=2` rather than `pt=20150101, type=1` or `pt=20150101`.

- **Task rerunning and failover**

**The TRUNCATE TABLE operation.** `"truncate": "true"` is configured to ensure the idempotence of write operations. When a retry is made after a failed write attempt, the MaxCompute writer cleans up existing data and imports new data. This ensures the data is consistent after each rerunning. If the task is interrupted by any exceptions during the running process, the atomicity of the data cannot be guaranteed, nor will the data be rolled back or rerun automatically. It is required that you use this idempotence to rerun the task to ensure data integrity.



**Note:**

Setting "truncate" to "true" cleans up all the data of the specified partition or table, so proceed with caution.

#### 2.8.3.4.8 Configure the Memcache (OCS) writer

KVStore for Memcache (formerly known as OCS) is a seamlessly scalable distributed memory database service with high performance and reliability. KVStore for Memcache has been developed based on high-performance storage technologies and the Apsara distributed compute system. It provides a complete database solution for hot standby, fault recovery, monitoring, and data migration.

KVStore for Memcache supports the out-of-the-box deployment mode, and relieves the database load for dynamic web applications using the cache service, thus accelerating the overall response of the website.

Similar to the local Memcache databases, KVStore for Memcache is compatible with the memcached protocol. You can use it directly in your operating environment. The difference is that the hardware and data of KVStore for Memcache are deployed in the cloud, providing complete infrastructure, network security, and system maintenance services. All these services are billed on a Pay-As-You-Go basis.

The Memcache writer writes data into Memcache channels based on the memcached protocol.

Currently, the Memcache writer supports only one write mode. Data types written in different modes are converted differently:


- **text:** The Memcache writer serializes source data to the string type, and uses the specified value of the fieldDelimiter parameter as the delimiter.
- **Binary:** currently not supported.

##### Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None



Parameter	Description	Required	Default value
writeMode	<p>The Memcache writer writes data in the following modes:</p> <ul style="list-style-type: none"><li>• <b>set:</b> Stores the data.</li><li>• <b>add:</b> Stores the data only when this key does not exist (not supported currently).</li><li>• <b>replace:</b> Stores the data only when this key exists (not supported currently).</li><li>• <b>append:</b> Stores data after the existing key, and ignores the exptime (not supported currently).</li><li>• <b>prepend:</b> Stores data before the existing key, and ignores the exptime (not supported currently).</li></ul>	Yes	None

Parameter	Description	Required	Default value
writeFormat	<p>Currently, the Memcache writer supports writing data in only one format:</p> <p><b>TEXT:</b> Serialize the source data to the text format with the first field being the key written into Memcache, and all subsequent fields to the string type. Use the specified value of the fieldDelimiter parameter as the delimiter to concatenate the text data into a complete string and write it into Memcache.</p> <p>For example, source data is:</p> <pre>   ID   NAME   COUNT    ---- ----- -----    23   "CDP"   100     </pre> <p>If the fieldDelimiter parameter is set to <code>\^</code>, the format of data written into Memcache is:</p> <pre>   KEY (OCS)   VALUE(OCS)    ----- :-----    23         CDP\^100     </pre>	No	None
expireTime	<p>The cache invalidation time for the Memcache value. Currently, Memcache supports two types of the invalidation time.</p> <ul style="list-style-type: none"> <li>• Unix time (the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970, minus leap seconds.) indicates that data is invalid at a certain time point in the future.</li> <li>• The relative time (in seconds) starting from the current time point. It specifies the time range during which data is valid.</li> </ul>	No	0. The value 0 indicates permanently valid.
940	 <b>Note:</b> If the invalidation time is larger than		Issue: 20200116

Parameter	Description	Required	Default value
batchSize	The quantity of records submitted in one operation. Setting this parameter can greatly reduce the interactions between CDP and Memcache over the network, and increase the overall throughput. However, an excessively large value may cause the running process of CDP to become out of memory. (Writing in batches is not supported for the current Memcache version.)	No	1024

Configure the Memcache writer in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the Memcache writer in script mode

**Use the data generated from memory and imported into Memcache.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "Parameter ": {},
 "name": "Reader ",
 "category": "reader"
 },
 {
 "stepType": "ocs", // The reader type.
 "parameter": {
 "writeFormat": "text", // The data format used when
the Memcache writer writes data.
 "expireTime": 1000, // The expiration time of the
memcache value.
 "indexes": 0,
 "datasource": "", // The data source.
 "writeMode": "set", // The writing method.
 "batchSize": "256", // The number of records to write
per batch.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 },
}
```

```

 "speed": {
 "throttle":false, // The value false means that the
 bandwidth is not throttled. The value true means that the bandwidth
 is throttled. The maximum transmission rate takes effect only if you
 specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
 threads.
 "dmu":1// The number of DMUs.
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
}

```

#### 2.8.3.4.9 Configure MongoDB Writer

This topic describes the data types and parameters supported by MongoDB Writer and how to configure it by using the code editor.

MongoDB Writer connects to a remote MongoDB database by using the Java client MongoClient and writes data to the database. The latest version of MongoDB has improved the locking feature from database locks to document locks. With the powerful index functionalities of MongoDB, MongoDB Writer can efficiently write data to MongoDB databases. If you need to update data, specify the primary key.



#### Note:

- You must configure the connection before configuring MongoDB Writer. For more information, see [Add a MongoDB connection](#).
- If you use ApsaraDB for MongoDB, the MongoDB database has a root account by default.
- For security concerns, Data Integration only supports access to a MongoDB database by using a MongoDB database account. When adding a MongoDB connection, do not use the root account for access.

MongoDB Writer obtains data from a Data Integration reader, and converts the data types to those supported by MongoDB. Data Integration does not support arrays. MongoDB supports arrays and the array index is useful.

To use MongoDB arrays, you can convert strings to MongoDB arrays by configuring a parameter and write the arrays to a MongoDB database.

## Data types

MongoDB Writer supports most MongoDB data types. Ensure that your data types are supported.

The following table lists the data types supported by MongoDB Writer.

Category	MongoDB data type
Integer	Int and Long
Float	Double
String	String and Array
Date and time	Date
Boolean	Boolean
Binary	Bytes


**Note:**

When data of the Date type is written to a MongoDB database, the type of the data is converted to Datetime.

## Parameters

Parameter	Description	Require	Default value
<b>datasource</b>	The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.	Yes	None
<b>collection Name</b>	The name of the MongoDB collection.	Yes	None
<b>column</b>	The columns in MongoDB. <ul style="list-style-type: none"><li>• <b>name:</b> the column name.</li><li>• <b>type:</b> the data type of the column.</li><li>• <b>splitter:</b> the delimiter. Specify this field only when you need to convert the string to an array. The string is split based on the specified delimiter, and the split strings are saved in a MongoDB array.</li></ul>	Yes	None

Parameter	Description	Require	Default value
writeMode	<p>Specifies whether to overwrite data.</p> <ul style="list-style-type: none"> <li>· <b>isReplace:</b> A value of true indicates that MongoDB Writer overwrites the data in the destination table with the same primary key. If you set this parameter to false, the data is not overwritten.</li> <li>· <b>replaceKey:</b> the primary key for each record. Data is overwritten based on this primary key. The primary key must be unique.</li> </ul>	No	None

Parameter	Description	Required	Default value
preSql	<p>The action to perform before the data synchronization node is run. For example, you can clear outdated data before data synchronization. If the preSql parameter is left empty, no action is performed before data synchronization. Ensure that the value of the preSql parameter complies with the JSON syntax. The format requirements for the preSql parameter are as follows:</p> <ul style="list-style-type: none"> <li>You need to configure the type field to specify the action type. Valid values: drop and remove. Example: <code>"preSql": {"type": "remove"}</code>.</li> <li>drop: deletes the collection specified by the collectionName parameter and the data in the collection.</li> <li>remove: deletes data based on conditions.</li> <li>json: the conditions for deleting data. Example: <code>"preSql":{"type":"remove", "json":"'operationTime':{'\$gte':ISODate('{\$last_day}T00:00:00.424+0800')}'"}</code>. In the preceding JSON string, <code>{\$last_day}</code> is a scheduling parameter of DataWorks. The format is <code>[\$yyyy-mm-dd]</code>. You can use comparison operators (such as \$gt, \$lt, \$gte, and \$lte), logical operators (such as \$and and \$or), and functions (such as max, min, sum, avg, and ISODate) supported by MongoDB as needed. For more information, see the MongoDB query syntax.</li> </ul> <p>Data Integration uses the following standard MongoDB API to query and delete the specified data:</p> <pre>query=(BasicDBObject) com.mongodb.util.JSON.parse(json); col.deleteMany(query);</pre> <div>  <b>Note:</b> If you need to delete data based on conditions, we recommend that you specify the conditions in JSON format preferentially. </div>	No	None
Issue: 20200116	<ul style="list-style-type: none"> <li>item: the name, condition, and value for filtering data. Example: <code>"preSql":{"type":"</code> </li> </ul>		945

Configure MongoDB Writer by using the codeless UI

**Currently, the codeless UI is not supported for MongoDB Writer.**

Configure MongoDB Writer by using the code editor

**In the following code, a node is configured to write data to a MongoDB database. For more information about the parameters, see the preceding parameter description.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "mongodb", // The writer type.
 "parameter": {
 "datasource": "", // The connection name.
 "column": [
 {
 "name": "name", // The name of the column to be
 "type": "string" // The data type.
 },
 {
 "name": "age",
 "type": "int"
 },
 {
 "name": "id",
 "type": "long"
 },
 {
 "name": "wealth",
 "type": "double"
 },
 {
 "name": "hobby",
 "type": "array",
 "splitter": " "
 },
 {
 "name": "valid",
 "type": "boolean"
 },
 {
 "name": "date_of_join",
 "format": "yyyy-MM-dd HH:mm:ss",
 "type": "date"
 }
]
 },
 "writeMode": { // The write mode.
 "isReplace": "true",
 "replaceKey": "id"
 }
 }
]
}
```



```

 "collectionName": "datax_test"// The name of the
MongoDB collection.
 },
 "name": "Writer",
 "category": "writer"
 }
],
"setting": {
 "errorLimit": {// The maximum number of dirty data records
allowed.
 "record": "0"
 },
 "speed": {
 "jvmOption": "-Xms1024m -Xmx1024m",
 "throttle": true,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "concurrent": 1,// The maximum number of concurrent
threads.
 "dmu": 1// The DMU value.
 "mbps": "1"// The maximum transmission rate.
 }
},
"order": {
 "hops": [// Synchronize data from the reader to the writer.
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

#### 2.8.3.4.10 Configure MySQL Writer

This topic describes the data types and parameters supported by MySQL Writer and how to configure it by using the codeless UI and code editor.

MySQL Writer allows you to write data to tables stored in MySQL databases.

Specifically, MySQL Writer connects to a remote MySQL database through

Java Database Connectivity (JDBC), and runs an `INSERT INTO` or `REPLACE INTO`

statements to write data to the MySQL database. MySQL uses the InnoDB engine so that data is written to the database in batches.



#### Note:

- You must configure the connection before configuring MySQL Writer. For more information, see [Add MySQL data sources](#).
- Currently, MySQL Writer does not support MySQL 8.0 or later.

MySQL Writer can be used as a data migration tool by users such as database administrators (DBAs). MySQL Writer obtains data from a Data Integration reader,

and writes the data to the destination database based on value of the `writeMode` parameter.

**Note:**

A data synchronization node that uses MySQL Writer must have at least the permission to run the `INSERT INTO` or `REPLACE INTO` statement. Whether other permissions are required depends on the SQL statements specified in the `preSql` and `postSql` parameters when you configure the node.

### Data types


MySQL Writer supports most MySQL data types. Ensure that your data types are supported.


The following table lists the data types supported by MySQL Writer.

Category	MySQL data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, and YEAR
Float	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

### Parameters

Parameter	Description	Require	Default value
<code>datasource</code>	The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.	Yes	None
<code>table</code>	The name of the table to be synchronized.	Yes	None

Parameter	Description	Required	Default value
writeMode	<p><b>The write mode. Valid values:</b> insert into, on duplicate key update, and replace into.</p> <ul style="list-style-type: none"> <li>insert into: <b>If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows and is regarded as dirty data.</b></li> <li>on duplicate key update: <b>If no primary key conflict or unique index conflict occurs, the action is the same as that of insert into. If a conflict occurs, specified fields in original rows are updated.</b></li> <li>replace into: <b>If no primary key conflict or unique index conflict occurs, the action is the same as that of insert into. If a conflict occurs, original rows are deleted and new rows are inserted. That is, all fields of original rows are replaced.</b></li> </ul>	No	insert
column	<p><b>The columns in the destination table to which data is written. Separate the columns with a comma (.). Example: "column": ["id", "name", "age"]. Set the value to an asterisk (*) if data is written to all the columns in the destination table. Example: "column": ["*"].</b></p>	Yes	None
preSql	<p><b>The SQL statement to run before the data synchronization node is run. For example, you can clear outdated data before data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor.</b></p> <div>  <b>Note:</b>  <b>If you specify multiple SQL statements in the code editor, the system does not ensure that they are run in the same transaction.</b> </div>	No	None

Parameter	Description	Required	Default value
postSql	<p>The SQL statement to run after the data synchronization node is run. For example, you can add a timestamp after data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor.</p> <div>  <b>Note:</b>            If you specify multiple SQL statements in the code editor, the system does not ensure that they are run in the same transaction.         </div>	No	None
batchSize	The number of data records to write at a time. Setting this parameter can greatly reduce the interactions between Data Integration and the MySQL database over the network, and increase the throughput. However, an excessively large value may lead to the out of memory (OOM) error during the data synchronization process.	No	1,024

Configure MySQL Writer by using the codeless UI

### 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.

Parameter	Description
Connection	The <code>datasource</code> parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks.
Table	The <code>table</code> parameter in the preceding parameter description.
Statement Run Before Sync	The <code>preSql</code> parameter in the preceding parameter description. Enter the SQL statement to run before the data synchronization node is run.
Statement Run After Sync	The <code>postSql</code> parameter in the preceding parameter description. Enter the SQL statement to run after the data synchronization node is run.


Parameter	Description
<b>Solution to Primary Key Violation</b>	The <code>writeMode</code> parameter in the preceding parameter description. Select the expected write mode.

2. Configure field mapping (the `column` parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

Configuration item	Description
<b>Map Fields with the Same Name</b>	Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.
<b>Map Fields in the Same Line</b>	Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.
<b>Delete All Mappings</b>	Click Delete All Mappings to remove mappings that have been established.
<b>Auto Layout</b>	The fields are automatically sorted based on specified rules .
<b>Change Fields</b>	You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, while other blank rows are ignored.
<b>Add</b>	<ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as <code>\${bizdate}</code>.</li><li>• You can enter functions supported by relational databases, such as <code>now()</code> and <code>count(1)</code>.</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul>

### 3. Configure the channel.

Parameter	Description
DMU	<p>The billing unit of Data Integration.</p> <div> <b>Note:</b> Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>
Concurrent Threads	<p>The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.</p>
Bandwidth Throttling	<p>Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.</p>
Dirty Data Records Allowed	<p>The maximum number of dirty data records allowed.</p>
Resource Group	<p>The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance.</p>

Configure MySQL Writer by using the code editor

In the following code, a node is configured to write data to a MySQL database. For more information about the parameters, see the preceding parameter description.

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "mysql", // The writer type.
 "parameter": {
```

```

 "postSql": [], // The SQL statement to run after the
data synchronization node is run.
 "datasource": "", // The connection name.
 "column": [// The columns to be synchronized.
 "id",
 "value"
],
 "writeMode": "insert", // The write mode.
 "batchSize": 1024, // The number of data records to
write at a time.
 "table": "", // The name of the table to be synchronized
 },
 "preSql": [] // The SQL statement to run before the data
synchronization node is run.
 },
 "name": "Writer",
 "category": "writer"
}
],
"setting": {
 "errorLimit": { // The maximum number of dirty data records
allowed.
 "record": "0"
 },
 "speed": {
 "throttle": false, // Specifies whether to enable bandwidth
throttling.
 "concurrent": 1, // The maximum number of concurrent threads
 },
 "dmu": 1 // The DMU value.
}
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}
}

```

### 2.8.3.4.11 Configure Oracle Writer

This topic describes the data types and parameters supported by Oracle Writer and how to configure it by using the codeless UI and code editor.

Oracle Writer allows you to write data to tables stored in primary Oracle databases. Specifically, Oracle Writer connects to a remote Oracle database through Java Database Connectivity (JDBC), and runs an `INSERT INTO` statement to write data to the Oracle database.



#### Note:

**You must configure the connection before configuring Oracle Writer. For more information, see [Add Oracle data sources](#).**

Oracle Writer is designed for extract-transform-load (ETL) developers to import data from data warehouses to Oracle databases. Oracle Writer can also be used as a data migration tool by users such as database administrators (DBAs).

Oracle Writer obtains data from a Data Integration reader, connects to a remote Oracle database through JDBC, and runs an INSERT INTO statement to write data to the Oracle database.

#### Data types

Oracle Writer supports most Oracle data types. Ensure that your data types are supported.

The following table lists the data types supported by Oracle Writer.

Category	Oracle data type
Integer	NUMBER, ROWID, INTEGER, INT, and SMALLINT
Float	NUMERIC, DECIMAL, FLOAT, DOUBLE PRECISION, and REAL
String	LONG, CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, CLOB, NCLOB, CHARACTER, CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, and NCHAR VARYING
Date and time	TIMESTAMP and DATE
Boolean	BIT and BOOLEAN
Binary	BLOB, BFILE, RAW, and LONG RAW

#### Parameters

Parameter	Description	Require	Default value
datasource	The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.	Yes	None
table	The destination table name.	Yes	None



Parameter	Description	Required	Default value
writeMode	<p><b>The write mode. Valid values:</b> insert into, on duplicate key update, and replace into.</p> <ul style="list-style-type: none"> <li>insert into: <b>If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows and is regarded as dirty data.</b></li> <li>on duplicate key update: <b>If no primary key conflict or unique index conflict occurs, the action is the same as that of insert into. If a conflict occurs, specified fields in original rows are updated.</b></li> <li>replace into: <b>If no primary key conflict or unique index conflict occurs, the action is the same as that of insert into. If a conflict occurs, original rows are deleted and new rows are inserted. That is, all fields of original rows are replaced.</b></li> </ul>	No	insert
column	<p><b>The columns in the destination table to which data is written. Separate the columns with a comma (,). Example: "column": ["id", "name", "age"]. Set the value to an asterisk (*) if data is written to all the columns in the destination table. Example: "column": ["*"].</b></p>	Yes	None
preSql	<p><b>The SQL statement to run before the data synchronization node is run. For example, you can clear outdated data before data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor.</b></p>	No	None
postSql	<p><b>The SQL statement to run after the data synchronization node is run. For example, you can add a timestamp after data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor.</b></p>	No	None

Parameter	Description	Required	Default value
batchSize	The number of data records to write at a time. Setting this parameter can greatly reduce the interactions between Data Integration and the Oracle database over the network, and increase the throughput. However, an excessively large value may lead to the out of memory (OOM) error during the data synchronization process.	No	1,024

Configure Oracle Writer by using the codeless UI

### 1. Configure the connections.

Configure the source and destination connections for the data synchronization node.


Parameter	Description
Connection	The <code>datasource</code> parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks.
Table	The <code>table</code> parameter in the preceding parameter description.
Statement Run Before Sync	The <code>preSql</code> parameter in the preceding parameter description. Enter the SQL statement to run before the data synchronization node is run.
Statement Run After Sync	The <code>postSql</code> parameter in the preceding parameter description. Enter the SQL statement to run after the data synchronization node is run.
Solution to Primary Key Violation	The <code>writeMode</code> parameter in the preceding parameter description. Select the expected write mode.

## 2. Configure field mapping (the column parameter in the preceding parameter description).

Fields in the source table (left) have a one-to-one mapping with fields in the destination table (right). You can click Add to add a field or move the pointer over a field and click the Delete icon to delete a field.

Configuration item	Description
Map Fields with the Same Name	Click Map Fields with the Same Name to establish a mapping between fields with the same name. Note that the data types of the fields must match.
Map Fields in the Same Line	Click Map Fields in the Same Line to establish a mapping for the same row. Note that the data types of the fields must match.
Delete All Mappings	Click Delete All Mappings to remove mappings that have been established.
Auto Layout	The fields are automatically sorted based on specified rules .
Change Fields	You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, while other blank rows are ignored.
Add	<ul style="list-style-type: none"><li>• You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'.</li><li>• You can use scheduling parameters, such as \${bizdate}.</li><li>• You can enter functions supported by relational databases, such as now() and count(1).</li><li>• If the value you entered cannot be parsed, the type is displayed as Unidentified.</li></ul>

## 3. Configure the channel.

Parameter	Description
DMU	<p>The billing unit of Data Integration.</p> <div> <b>Note:</b> Use caution while setting the DMU parameter. Its value limits the maximum number of concurrent threads.</div>

Parameter	Description
Concurrent Threads	The maximum number of concurrent data synchronization threads. If you specify this parameter, the data records are split based on the shard key specified for the reader so that they are synchronized in multiple threads concurrently. This improves the transmission rate.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Resource Group	The resource group used for running the data synchronization node. By default, the node runs on the default resource group. If resources are insufficient, you can add a custom resource group and run the data synchronization node on the custom resource group. Set the resource group properly based on network conditions of the data stores, resource group usage, and business importance.

Configure Oracle Writer by using the code editor

**In the following code, a node is configured to write data to an Oracle database.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 {
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "oracle", // The writer type.
 "parameter": {
 "postSql": [], // The SQL statement to run after the
data synchronization node is run.
 "datasource": "",
 "session": [], // The settings of the session to the
database.
 "column": [// The columns to be synchronized.
 "id",
 "name"
],
 "encoding": "UTF-8", // The encoding.
 "batchSize": 1024, // The number of data records to
write at a time.
 }
 }
]
}
```

```

 "table":"","// The name of the table to be synchronized
 .
 "preSql":[]// The SQL statement to run before the data
synchronization node is run.
 },
 "name":"Writer",
 "category":"writer"
 }
],
 "setting":{
 "errorLimit":{
 "record":"0"// The maximum number of dirty data records
allowed.
 },
 "speed":{
 "throttle":false,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "concurrent":1, // The maximum number of concurrent
threads.
 "dmu":1// The DMU value.
 }
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
}

```

#### 2.8.3.4.12 Configure the OSS writer

The OSS writer provides the ability to write one or more table files in CSV-like format into OSS.



##### Note:

You must configure the data source before configuring the OSS writer. For more information, see [Add OSS data sources](#).

What is written and saved to the OSS file is a two-dimensional table in a logic sense, for example, text information in a CSV format.

The OSS writer provides the ability to convert the data synchronization protocol to a text file in OSS, which is a non-structured data storage. Currently, the OSS writer supports the following features:

- Only supports writing text files. The schema in the text file must be a two-dimensional table.
- Supports CSV-like format files with custom delimiters.

- Supports multi-thread writing, with each thread performing write operations on a subfile.
- Supports file rollover. A file exceeding a specific size must be switched. A file that contains rows exceeding a specific number of rows must be switched.

The OSS writer currently does not support the following features:

- Concurrent writing is not supported for a single file.
- OSS only supports the string type. The OSS writer writes data of the string type to OSS.

OSS only supports the string type. The following table lists the data types supported by the DataX OSS writer.

Data Integration data type	OSS data type
Integer	Long
Floating point	Double
String	String
Boolean	Bool
Date and time	Date

#### Parameters

Parameter	Description	Require	Default value
datasource	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None

Parameter	Description	Required	Default value
Object	<p>The file name written by the OSS writer. It is specified by the simulated path. If the bucket in the OSS data source for data synchronization is the test folder of test118, only test needs to be specified for object, without the bucket name. The file name synchronized to the OSS end is identical to the one entered in the source end.</p> <p>The format is "object": "test/DI", in which test is a folder, DI is the prefix of the file name (the suffix is a random string), and a forward slash (/) is used as the delimiter of the simulated OSS directory.</p>	Yes	None
writeMode	<p>The mode in which the OSS writer clears the existing data before writing data.</p> <ul style="list-style-type: none"> <li>• <b>truncate:</b> All objects with matched object name prefixes are cleared before writing. For example, if "object": "abc" is specified, all objects beginning with abc are cleared.</li> <li>• <b>append:</b> No processing is performed before writing. Data Integration OSS Writer writes data directly using the object name, and appends a random UUID suffix name to ensure no conflict of file names. For example, if the object name you specified is Data Integration, the name is actually entered as DI_XXXXXX_XXXX_XXXX.</li> <li>• <b>nonConflict:</b> If an object with the matched prefix exists in a specified path, an error is reported directly. For example, if "object": "abc", is specified, when an object beginning with abc123 exists, an error is reported.</li> </ul>	Yes	None
fileFormat	The format in which a file is written, including both CSV and text. Supported formats are CSV and text. If the data to be written contains column delimiters, the column delimiters are escaped with double quotation marks (") in the CSV escape syntax. For text format, the data to be written is separated by column delimiters without being escaped.	No	text

Parameter	Description	Required	Default value
fieldDelimiter	The delimiter used to separate the read fields.	No	,
encoding	The encoding of the file to be written.	No	UTF-8
nullFormat	The string that represents null. Since no standard strings can represent null in text files, Data Integration provides the nullFormat parameter to define which string represents a null pointer. For example, if you specify nullFormat="null", then Data Integration considers "null" as a null pointer.	No	None
header (advanced configuration, which is not supported in wizard mode)	The header used when a file is written in OSS. For example, ['id', 'name', 'age'].	No	None
maxFileSize (advanced configuration, which is not supported in wizard mode)	<p>The maximum size of a single object file written in OSS, which defaults to 10,000 x 10 MB. It is similar to the log rotation based on the log size in log4j log printing. For multipart upload in OSS, the size of each part is 10 MB (which is the minimum file granularity for log rotation, and maxFileSize smaller than 10 MB is also considered as 10 MB), and the maximum number of parts supported for each OSS InitiateMultipartUploadRequest is 10,000.</p> <p>When rotation occurs, the naming rule for object is the original object prefix+a random UUID+a suffix, such as _1, _2, _3.</p>	No	100,000 MB



Configure the OSS writer in wizard mode

### 1. Select data sources.

Configure the source and destination of the data for the synchronization task.

Parameter	Description
Data Source	The datasource parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.
Object prefix	The Object parameter provided in the preceding table. Enter a path to the OSS folder without the bucket name.
Column Delimiter	The fieldDelimiter parameter provided in the preceding table. Default value: a comma (,)
Encoding	The encoding parameter provided in the preceding table. Default value: UTF-8.
Null String	The nullFormat parameter provided in the preceding table, and defines a string that represents null.

### 2. Configure field mappings.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.

### 3. Configure the channel.

Parameter	Description
DMU	The data processing capabilities. A DMU represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.
Concurrent Jobs	The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.

Parameter	Description
Transmission Rate	Indicates whether to enable bandwidth throttling. You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Task Resource Group	The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.

Configure the OSS writer in script mode

An example is described as follows. For more information about parameters, see the corresponding section.

```
{
 "type": "job",
 "version": "2.0",
 "steps": [
 {
 //The following is a reader template. You can find the
 corresponding reader plug-in documentations.
 "stepType": "stream",
 "Parameter": {},
 "name": "Reader ",
 "category": "reader"
 },
 {
 "stepType": "oss", // The reader type.
 "parameter": {
 "nullFormat": "", // The string that represents null.
 "dateFormat": "", // The date format.
 "datasource": "", // The data source.
 "writeMode": "", // The writing method.
 "encoding": "", // The encoding.
 "fieldDelimiter": ",", // The column delimiter.
 "fileFormat": "", // The file type.
 "object": "" // The object name prefix,
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
 bandwidth is not throttled. The value true means that the bandwidth
```

```

is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
threads.
 "dmu":1// The number of DMUs.
 }
},
"order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
}
}

```

### 2.8.3.4.13 Configure the PostgreSQL writer

The PostgreSQL writer writes data into PostgreSQL. At the underlying level, the PostgreSQL writer connects to a remote PostgreSQL database over JDBC and runs SELECT statements to extract data from the database.



#### Note:

You must configure the data source before configuring the PostgreSQL writer. For more information, see [Add a PostgreSQL connection](#).

Specifically, the PostgreSQL writer connects to a remote PostgreSQL database over JDBC. Then, it generates SELECT statements based on your configurations, and sends the statements to the remote PostgreSQL database. After the database successfully runs the statements, the PostgreSQL reader retrieves the results, formats the results based on the custom data types of the CDP, and sends the formatted results to the writer.

- The PostgreSQL writer generates SQL statements based on the table, column, and where parameters, and sends the generated SQL statements to the PostgreSQL database.
- The PostgreSQL writer directly sends the `querySql` parameter setting to the PostgreSQL database.

#### Data types

The PostgreSQL writer supports most PostgreSQL data types. Since still some of the PostgreSQL data types are not supported, verify that your data types are supported.

The following table lists data types supported by the PostgreSQL writer.

Data Integration data type	PostgreSQL data type
Long	BIGINT, BIGSERIAL, INTEGER, SMALLINT, and SERIAL
Double	DOUBLE, PRECISION, MONEY, NUMERIC, and REAL
String	VARCHAR, CHAR, TEXT, BIT, and INET
Date	DATE, TIME, and TIMESTAMP
Boolean	BOOL
Bytes	BYTEA

**Note:**

- Data types that are not listed in the table are not supported.
- You need to convert the MONEY, INET, or BIT type to another by using a statement such as `a_inet::varchar`.

## Parameters

Parameter	Description	Require	Default value
<b>datasource</b>	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
<b>table</b>	The name of the destination table.	Yes	None
<b>writeMode</b>	The writing method. Valid value: insert.  insert: When a data record violates the primary key or unique index constraint, Data Integration considers it dirty and retains the original data.	No	insert
<b>column</b>	The columns in the destination table to which data is written. Separate the columns with a comma (.). Example: "column": ["id", "name", "age"]. Set the value to an asterisk (*) if data is written to all the columns in the destination table. Example: "column": ["*"].	Yes	None

Parameter	Description	Require	Default value
preSql	The SQL statement runs before the data synchronization task starts. Currently, you can run only one SQL statement in wizard mode but multiple SQL statements in script mode. For example, you can run a statement to clear outdated data.	No	None
postSql	The SQL statement runs after the data synchronization task ends. Currently, you can run only one SQL statement in wizard mode but multiple SQL statements in script mode. For example, you can run a statement to add a timestamp.	No	None
batchSize	The number of data records to write per batch . Setting this parameter can greatly reduce the interactions between the data synchronization task and the PostgreSQL database over the network , and increase the throughput. However, an excessively large value may cause the running process of Data Integration to become out of memory (OOM).	No	1024

Configure the PostgreSQL writer in wizard mode

### 1. Select data sources.

Configure the source and destination of the data for the synchronization task.

Field	Description
Data Source	The datasource parameter provided in the preceding table. Select a data source type, and enter the name of a data source that has been configured in DataWorks.
Table	The table parameter provided in the preceding table.
Statements Run Before Import	The preSql parameter provided in the preceding table. Enter a SQL statement to be run before the data synchronization task starts.
Statements Run After Import	The postSql parameter provided in the preceding table. Enter a SQL statement to be run after the data synchronization task is run.

2. Configure field mappings. It is equivalent to setting the column parameter provided in the preceding table.

You can map the left-side source table fields to the right-side destination table fields. You can also click Add to add a field or click the Delete icon to delete a field in the source table.

- After you click Map Fields in the Same Line, each source table field is mapped to the destination table field in the same line if exists. Ensure that the conversion between data types is feasible.
- After you click Auto Layout, fields are automatically sorted based on specific rules.

3. Configure the channel.

Parameter	Description
DMU	The data processing capabilities. A DMU represents the data processing capabilities for data integration, given a minimum configuration of the CPU, memory, network, and other resources.
Concurrent Jobs	The maximum number of concurrent threads to read and write data to data storage within the data synchronization task.
Transmission Rate	Indicates whether to enable bandwidth throttling. You can throttle the bandwidth and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Task Resource Group	The servers on which tasks are run. If an excessively large number of tasks are run on the default resource group, some tasks may be delayed due to insufficient resources. In this case, you can configure additional servers.

Configure the PostgreSQL writer in script mode

An example is described as follows. For more information about parameters, see the corresponding section.

```
{
 "type": "job",
 "version": "2.0 ", // The version number.
```

```

"steps": [// The following template is used to configure the
reader. For more information, see the corresponding section.
{
 "stepType": "stream",
 "parameter": {},
 "name": "Reader ",
 "category": "reader"
},
{
 "stepType": "postgresql", // The plug-in name.
 "parameter": {
 "postSql": [], // The SQL statement runs after the
data synchronization task ends.
 "datasource": "// The data source.
 "col1",
 "col2"
],
 "table": "", // The table name.
 "preSql": [], // The SQL statement runs before the
data synchronization task starts.
 },
 "name": "Writer",
 "category": "writer"
}
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
}
}

```

#### 2.8.3.4.14 Configure the Redis writer


**Remote Dictionary Server (Redis)** is a high-performance, key-value, and in-memory data structure store. In order to achieve its outstanding performance, Redis works with an in-memory dataset. You can also persist data if necessary. It can be used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries.

The Redis writer is a writer plug-in based on the Data Integration framework. It can import data from a data warehouse or other data sources to a Redis instance. The Redis writer interacts with a Redis server through Jedis. As a preferred Java client development kit provided by Redis, Jedis supports almost all the features of Redis.


**Note:**


- You must configure the data source before configuring the OSS writer. For more information, see [Add Redis data sources](#).
- If you write data into Redis using the Redis writer and the value type is list, the result of the re-run synchronization task is not idempotent. If the value type is list, you must manually clear the corresponding data on Redis when re-running the synchronization task.

## Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
keyIndexes	<p>The columns of the source table that are used as key (starts with 0 for the first column). If the key is the combination of the first and second columns, the value of keyIndexes is [0,1].</p> <div> <b>Note:</b> After you specify the keyIndexes parameter, the Redis writer considers the remaining columns as the value. If you do not need to synchronize all the fields, filter columns at the reader side.</div>	Yes	None
keyFieldDelimiter	The key delimiter used when writing data to Redis. Take key=key1\u0001id as an example. If multiple keys need to be concatenated, the value is required. If only one key exists, this configuration item can be ignored.	No	\u0001



Parameter	Description	Require	Default value
batchSize	The number of data records to write per batch . Setting this parameter can greatly reduce the interactions between Data Integration and the PostgreSQL database over the network, and increase the throughput. However, an excessively large value may cause the running process of Data Integration to become out of memory (OOM).	No	1,000
expireTime	<p>The expiration time of Redis value cache. The value is valid permanently if this parameter is not specified.</p> <ul style="list-style-type: none"> <li>seconds: The relative time (in seconds) starting from the current time point. It specifies the time range during which data is valid.</li> <li>unixtime: Unix time (the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970, minus leap seconds.) indicates that data is invalid at a certain time point in the future.</li> </ul> <div>  <p><b>Note:</b> If the expiration time is larger than 60*60*24*30 (30 days), the server identifies the invalidation time as the Unix time.</p> </div>	No	0 (0 indicates permanent validity)
timeout	The time-out of writing data into Redis, measured in milliseconds.	No	30000 (30 seconds of network disconnection)
dateFormat	The time format when data is written into Redis: yyyy-MM-dd HH:mm:ss.	No	None

Parameter	Description	Required	Default value
writeMode	<p>Redis supports diverse types of values, such as strings, lists, sets, sorted sets, and hashes. The Redis writes allows you to writes data of these five types. The value of the writeMode parameter varies depending on the data type.</p> <div>  <b>Note:</b>  When configuring the Redis writer, you can only choose one data type from the following five options: </div> <ul style="list-style-type: none"> <li>String <div> <pre>"writeMode":{   "type": "string",   "mode": "set",   "valueFieldDelimiter": "\u0001" }</pre> </div> <div> <b>Parameters</b> <ul style="list-style-type: none"> <li>type <div> <b>Description:</b> The value type (string)  <b>Required:</b> Yes </div> </li> <li>mode <div> <b>Description:</b> The writing method when the value type is string.  <b>Required:</b> Yes. Valid value: set (stores the data, or overwrites the existing data.) </div> </li> <li>valueFieldDelimiter <div> <b>Description:</b> The delimiter between values when values are strings, such as value1\u0001value2\u0001value3. If there are more than two columns for each row in the source data, specify this parameter. If only two columns of source data exist: key and value, do not specify this parameter.  <b>Required:</b> No  <b>Default value:</b> \u0001 </div> </li> </ul> </div> </li> <li>List <div> <pre>"writeMode":{   "type": "list",   "mode": "lpush rpush",   "valueFieldDelimiter": "\u0001" }</pre> </div> <div> <b>Parameters</b> </div> </li> </ul>	No	string

Configure the Redis writer in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the Redis writer in script mode

**In the following script, a task is configured to write data to Redis. For more information about parameters, see the corresponding section.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "Parameter ": {},
 "name": "Reader ",
 "category": "reader"
 },
 {
 "stepType": "redis", // The reader type.
 "parameter": {
 "expireTime": { // The expiration time of the redis
value cache.
 "seconds": 1000
 },
 "keyFieldDelimiter": "u0001", // The key delimiter when
data is written to Redis.
 "dateFormat": "yyyy-MM-dd HH:mm:ss", // The time
format when data is written to Redis.
 "datasource": "", // The data source.
 "writeMode": { // The writing method.
 "mode": " ", // The writing method when the data
type is specified.
 "valueFieldDelimiter": " ", // The value delimiter.
 "type": " // The value type.
 },
 "keyIndexes": [// The primary key index.
 0,
 1
],
 "batchSize": "1000", // The number of records to write
per batch.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 }
 }
}
```

```
 "dmu":1// The number of DMUs.
 }
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
 }
}Writer"
}
}
```

#### 2.8.3.4.15 Configure Elasticsearch Writer

This topic describes the data types and parameters supported by Elasticsearch Writer and how to configure it by using the code editor.


Elasticsearch is an open-source product that complies with the Apache open standards. It is the mainstream search engine for enterprise data. Elasticsearch is a distributed search and data analysis tool based on Lucene. The mappings between Elasticsearch core concepts and database core concepts are as follows:

```
Relational database (instance) -> database -> table -> row -> column
Elasticsearch -> index -> type -> document -> field
```

Elasticsearch can contain multiple indexes (databases). Each index can contain multiple types (tables). Each type can contain multiple documents (rows). Each document can contain multiple fields (columns). Elasticsearch Writer uses the RESTful API of Elasticsearch to write multiple data records retrieved by a reader to Elasticsearch at a time.


##### Parameters

Parameter	Description	Require	Default value
endpoint	The endpoint of Elasticsearch, in the format of <code>http://xxxx.com:9999</code> .	No	None

Parameter	Description	Require	Default value
accessId	<p>The username for accessing Elasticsearch, which is used for authorization when a connection with Elasticsearch is established.</p> <div>  <b>Note:</b>                      The accessId and accessKey parameters are required. If you do not set the parameters, an error is returned. If you use on-premises Elasticsearch for which basic authentication is not configured, the username and password are not required. In this case, you can set the accessId and accessKey parameters to random values.                 </div>	No	None
accessKey	The password for accessing Elasticsearch.	No	None
index	The index name in Elasticsearch.	No	None
indexType	The type name in the index of Elasticsearch.	No	Elasticsearch
cleanup	Specifies whether to clear existing data in the index. The method used to clear the data is to delete and rebuild the corresponding index. The default value false indicates that the existing data in the index is retained.	No	false
batchSize	The number of data records to write at a time.	No	1,000
trySize	The number of retries after a failure.	No	30
timeout	The client timeout.	No	600,000
discovery	Specifies whether to enable Node Discovery. When Node Discovery is enabled, the server list in the client is polled and regularly updated.	No	false
compression	Specifies whether to enable compression for an HTTP request.	No	true
multiThread	Specifies whether to use multiple threads for an HTTP request.	No	true

Parameter	Description	Require	Default value
ignoreWriteError	<b>Specifies whether to ignore write errors and proceed with writing without retries.</b>	No	false
ignoreParseError	<b>Specifies whether to ignore format parsing errors and proceed with writing.</b>	No	true
alias	<p><b>The alias of the index. The alias feature of Elasticsearch is similar to the view feature of a traditional database. For example, if you create an alias named my_index_alias for the index my_index, the operations on my_index_alias also take effect on my_index.</b></p> <p><b>Configuring alias means that after the data import is completed, an alias is created for the specified index.</b></p>	No	None
aliasMode	<b>The mode in which an alias is added after the data is imported. Valid values: append and exclusive.</b>	No	append
settings	<p><b>The delimiter (-,-) for splitting the source data if you are inserting an array to Elasticsearch. Example:</b></p> <p><b>The source column stores data a-,-b-,-c-,-d of the String type. Elasticsearch Writer uses the delimiter (-,-) to split the source data and obtains the array ["a", "b", "c", "d"]. Then, Elasticsearch Writer writes the array to the corresponding field in Elasticsearch.</b></p>	No	-,-

Parameter	Description	Require	Default value
column	<p><b>The fields of the document. The parameters for each field include basic parameters such as name and type and advanced parameters such as analyzer, format, and array.</b></p> <p><b>The field types supported by Elasticsearch are as follows:</b></p> <ul style="list-style-type: none"> <li>- id // The id type corresponds to the _id type in Elasticsearch, and can be considered as the unique primary key. Data with the same ID will be overwritten and not indexed</li> <li>- string</li> <li>- text</li> <li>- keyword</li> <li>- long</li> <li>- integer</li> <li>- short</li> <li>- byte</li> <li>- double</li> <li>- float</li> <li>- date</li> <li>- boolean</li> <li>- binary</li> <li>- integer_range</li> <li>- float_range</li> <li>- long_range</li> <li>- double_range</li> <li>- date_range</li> <li>- geo_point</li> <li>- geo_shape</li> <li>- ip</li> <li>- token_count</li> <li>- array</li> <li>- object</li> <li>- nested</li> </ul> <ul style="list-style-type: none"> <li>• <b>When the field type is text, you can specify the analyzer, norms, and index_options parameters. Example:</b> <pre>{   "name": "col_text",   "type": "text",   "analyzer": "ik_max_word" }</pre> </li> <li>• <b>When the field type is date, you can specify the format and timezone parameters, indicating the date serialization format and the time zone, respectively. Example:</b> <pre>{   "name": "col_date",   "type": "date",   "format": "MM-dd-yyyy HH:mm:ss" }</pre> </li> </ul>	Yes	None

Parameter	Description	Require	Default value
actionType	<p><b>The type of the action for writing data to Elasticsearch. Currently, Data Integration supports only the following action types: index and update. Default value: index.</b></p> <ul style="list-style-type: none"><li>• <b>index: Data Integration uses Index . Builder of the Elasticsearch SDK to construct a request for writing multiple data records at a time. In index mode, Elasticsearch first checks whether an ID is specified for the document to be inserted.</b><ul style="list-style-type: none"><li>- If the ID is not specified, Elasticsearch generates a unique ID by default. In this case, the document is directly inserted to Elasticsearch.</li><li>- If the ID is specified, Elasticsearch replaces the existing document with the document to be inserted.</li></ul></li></ul> <div> <b>Note:</b> <b>In this case, you cannot modify specific fields in the document.</b></div> <ul style="list-style-type: none"><li>• <b>update: Data Integration uses Update . Builder of the Elasticsearch SDK to construct a request for writing multiple data records at a time. In update mode, Elasticsearch calls the get method of InternalEngine to obtain the information of the original document for each update. In this way, you can modify specific fields. In update mode, you need to obtain the information of the original document for each update, which greatly affects the performance. However, you can modify specific fields in this mode. If the original document does not exist, the new document is directly inserted.</b></li></ul>	No	index



Configure Elasticsearch Writer by using the code editor

**In the following code, a node is configured to write data to Elasticsearch. For more information about the parameters, see the preceding parameter description.**

```
{
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 },
 "setting": {
 "errorLimit": {
 "record": "0"
 },
 "speed": {
 "concurrent": 1,
 "throttle": false
 }
 },
 "steps": [
 {
 "category": "reader",
 "name": "Reader",
 "parameter": {
 },
 "stepType": "stream"
 },
 {
 "category": "writer",
 "name": "Writer",
 "parameter": {
 "endpoint": "http://xxxx.com:9999",
 "accessId": "xxxx",
 "accessKey": "yyyy",
 "index": "test-1",
 "type": "default",
 "cleanup": true,
 "settings": {
 "index": {
 "number_of_shards": 1,
 "number_of_replicas": 0
 }
 },
 "discovery": false,
 "batchSize": 1000,
 "splitter": ",",
 "column": [
 {
 "name": "pk",
 "type": "id"
 },
 {
 "name": "col_ip",
 "type": "ip"
 },
 {
 "name": "col_double",
```

```
 "type": "double"
 },
 {
 "name": "col_long",
 "type": "long"
 },
 {
 "name": "col_integer",
 "type": "integer"
 },
 {
 "name": "col_keyword",
 "type": "keyword"
 },
 {
 "name": "col_text",
 "type": "text",
 "analyzer": "ik_max_word"
 },
 {
 "name": "col_geo_point",
 "type": "geo_point"
 },
 {
 "name": "col_date",
 "type": "date",
 "format": "yyyy-MM-dd HH:mm:ss"
 },
 {
 "name": "col_nested1",
 "type": "nested"
 },
 {
 "name": "col_nested2",
 "type": "nested"
 },
 {
 "name": "col_object1",
 "type": "object"
 },
 {
 "name": "col_object2",
 "type": "object"
 },
 {
 "name": "col_integer_array",
 "type": "integer",
 "array": true
 },
 {
 "name": "col_geo_shape",
 "type": "geo_shape",
 "tree": "quadtree",
 "precision": "10m"
 }
]
 },
 "stepType": "elasticsearch"
}
],
"type": "job",
"version": "2.0"
```

}

**Note:**

Currently, Elasticsearch that is deployed in a Virtual Private Cloud (VPC) supports only custom resource groups. A data synchronization node that is run on the default resource group may fail to connect to Elasticsearch.

### 2.8.3.4.16 Configure LogHub Writer

This topic describes the data types and parameters supported by LogHub Writer and how to configure it by using the code editor.

LogHub Writer allows you to transfer data from a Data Integration reader to LogHub through Log Service Java SDK.

**Note:**

LogHub does not guarantee idempotence. Rerunning a node after the node fails may result in redundant data.

LogHub Writer retrieves data from a Data Integration reader and converts the data types supported by Data Integration to String. When the number of the data records reaches the value specified for the batchSize parameter, LogHub Writer sends the data records to LogHub at a time through Log Service Java SDK. LogHub Writer sends 1,024 data records at a time by default. The batchSize parameter can be set to 4,096 at most.

#### Data types

The following table lists the data types supported by LogHub Writer.

Data Integration data type	LogHub data type
Long	String
Double	String
String	String
Date	String
Boolean	String
Bytes	String

## Parameters

Parameter	Description	Require	Default value
endpoint	The endpoint for accessing Log Service.	Yes	None
accessKeyId	The AccessKey ID for accessing Log Service.	Yes	None
accessKeySecret	The AccessKey secret for accessing Log Service.	Yes	None
project	The name of the destination Log Service project.	Yes	None
logstore	The name of the destination Logstore in Log Service.	Yes	None
topic	The selected topic.	No	Empty string
batchSize	The number of data records to write at a time.	No	1,024
column	The column name in each data record.	Yes	None

Configure LogHub Writer by using the codeless UI

**Currently, the codeless UI is not supported for LogHub Writer.**

Configure LogHub Writer by using the code editor

**In the following code, a node is configured to write data to LogHub. For more information about the parameters, see the preceding parameter description.**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { //
 "stepType": "stream",
 "parameter": {},
 "name": "Reader",
 "category": "reader"
 },
 {
 "stepType": "loghub", // The writer type.
 "parameter": {
 "datasource": "", // The connection name.
 "column": [// The columns to be synchronized.
 "col0",
 "col1",
 "col2",
 "col3",
 "col4",
 "col5"
],
 }
 }
]
}
```

```

 "topic": "",// The selected topic.
 "batchSize": "1024",// The number of data records to
write at a time.
 "logstore": ""// The name of the destination Logstore
in Log Service.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": ""// The maximum number of dirty data records
allowed.
 },
 "speed": {
 "concurrent": 3,// The maximum number of concurrent
threads.
 "throttle": false,// A value of false indicates that
the bandwidth is not throttled. A value of true indicates that the
bandwidth is throttled. The maximum transmission rate takes effect
only if you set this parameter to true.
 "dmu": 1// The DMU value.
 }
 },
 "order": {
 "hops": [
 {
 "from": "Reader",
 "to": "Writer"
 }
]
 }
}

```

#### 2.8.3.4.17 Configure the OpenSearch writer

The OpenSearch writer is designed for developers to insert or update data into OpenSearch tables. It allows developers to import processed data into OpenSearch tables and provides search services. The data transmission rate depends on the QPS of the account that manages the OpenSearch table.

##### Implementation

At the underlying level, the OpenSearch writer provides the openly available OpenSearch APIs.



##### Note:

- OpenSearch V3 uses internal dependent databases, with POM of com.aliyun.opensearch aliyun-sdk-opensearch 2.1.3.
- To use the OpenSearch writer, you must use JDK 1.6-32 or later versions. You can use the java-version command to view the JDK version.

- **Currently, the default resource group does not support connections to a VPC due to potential network problems.**

## Features

### Column order

The columns in OpenSearch are unordered, so you must use the OpenSearch writer to write data in accordance with the order of the specified columns. If the number of specified columns is less than that in OpenSearch, redundant columns are set to the default value or null.

For example, if the field list to be imported contains fields b and c but the OpenSearch table contains fields a, b, and c, enter "column": ["c","b"]. The first two columns in the reader are imported to fields c and b in OpenSearch, and the field a, into which new records are inserted, is set to the default value or null.

- **How to handle column configuration errors**

To ensure data is written in a reliable manner, data loss from redundant columns must be prevented to guarantee data quality. When redundant columns are written, the OpenSearch writer reports an error. For an OpenSearch table that contains fields a, b, and c, the OpenSearch writer reports an error if more than three fields are written by the OpenSearch writer.

- **Table configuration**

The OpenSearch writer can only write data to one table at a time.

- **Task rerunning and failover**

After one task is rerun, data is automatically overwritten based on IDs. Therefore , OpenSearch must contain one ID column. The ID uniquely identifies a row in OpenSearch. The data with the unique ID will be overwritten.

- **Task rerunning and failover**

After one task is rerun, data is automatically overwritten based on IDs.

The OpenSearch writer supports most OpenSearch data types. Since still some of the OpenSearch data types are not supported, verify that your data types are supported. The following table lists data types supported by the OpenSearch writer

.


Data Integration data type	OpenSearch data type
Integer	Int
Floating point	Double and float
String	TEXT, Literal, and SHORT_TEXT
Date and time	Int
Boolean	Literal

## Parameters

Parameter	Description	Required	Default value
accessId	The AccessKey ID for accessing the Alibaba Cloud system.	Yes	None
accessKey	The AccessKey Secret for accessing the Alibaba Cloud system.	Yes	None
host	The endpoint of OpenSearch. You can view the endpoint information on the details page. Generally, the address for the production environment is <code>http://opensearch-cn-internal.aliyuncs.com/</code> . The address for the test environment is <code>http://opensearch-cn-corp.aliyuncs.com/</code> .	Yes	None
indexName	The name of the OpenSearch project.	Yes	None
table	The table to which the data is written . You cannot enter more than one table, because DataX does not support importing multiple tables at a time.	Yes	None

Parameter	Description	Required	Default value
<b>column</b>	The columns to which data is written. If you need to write data to all the columns, set to "column": ["*"]. If you need to insert data into some OpenSearch columns, enter these columns: "column": ["id", "name"]. OpenSearch supports column filtering and switching. For example, if a table has three fields: a, b, and c, and you can synchronize only fields c and b, you can configure it to ["c", "b"]. During the import process, field a is automatically inserted and set to null.	Yes	None
<b>batchSize</b>	The number of data records to write per batch. Data is written into OpenSearch in batches. In general, the advantage of OpenSearch is searching, and its write performance (TPS) is not impressive. Proceed with the configuration based on the resources that your account have applied for. For OpenSearch, generally, the size of a data record is less than 1 MB, and the size of the data to write at a time is less than 2 MB.	This parameter is required for a partitioned table. Do not specify this parameter if the target table is a non-partitioned table.	300



Parameter	Description	Required	Default value
writeMode	<p>In the OpenSearch writer, "writeMode": "add/update" is configured to ensure the idempotence of write operations.</p> <ul style="list-style-type: none"><li>• "add": When a reattempt is made after a failed write attempt, the OpenSearch writer cleans up this data and imports the new data (atomic operation).</li><li>• "update": It indicates that data is inserted by using the modify operation (atomic operation).</li></ul> <div> <b>Note:</b>  In OpenSearch, batch insert is not an atomic operation, which may be partially successful. Therefore, writeMode is a critical option. OpenSearch V3 does not support the update operation currently.</div>	Yes	None
ignoreWriteError	<p>Ignores write errors.</p> <p>Example: "ignoreWriteError": true</p> <p>. OpenSearch write operations are performed in batches. It indicates whether to ignore the write errors of the current batch. If yes, other write operations keep going. If not, the current task ends, and an error is returned. We recommend that you keep the default setting.</p>	No	false
version	<p>The version of OpenSearch. Example : "version": "v3". OpenSearch V2 has multiple limitations on push operations, so OpenSearch V3 is preferable.</p>	No	v2

Configure the OpenSearch writer in script mode

**In the following script, a task is configured to write data to a OpenSearch data source.**

```
{
 "type": "job",
 "version": "1.0",
 "configuration": {
 "reader": {},
 "writer": {
 "plugin": "opensearch",
 "parameter": {
 "accessId": "*****",
 "accessKey": "*****",
 "host": "http://yyyy.aliyuncs.com",
 "indexName": "datax_xxx",
 "table": "datax_yyy",
 "column": [
 "appkey",
 "id",
 "title",
 "gmt_create",
 "pic_default"
],
 "batchSize": 500,
 "writeMode": add,
 "version": "v2",
 "ignoreWriteError": false
 }
 }
 }
}
```

#### 2.8.3.4.18 Configure the Table Store (OTS) writer

Table Store is a NoSQL database service built on the Apsara distributed operating system that allows you to store and access large amounts of structured data in real time. Table Store organizes data into instances and tables. Using data partition and server load balancing technologies, it provides seamless scaling.

The Table Store writer connects to a Table Store server by using the official Java SDK and writes data to the Table Store server by using the SDK. The Table Store writer has greatly optimized the write process, including retry upon write timeout, retry upon exceptions, and batch submission.

Currently, the Table Store writer supports all Table Store data types and supports the following two writing methods:

- **PutRow:** PutRow API for Table Store, which is used to insert data to a specified row. If this row does not exist, a new row is added. Otherwise, the original row is overwritten. If the row exists, overwrite the original row.

- **UpdateRow:** UpdateRow API for Table Store, which is used to update the data of a specified row. If the row does not exist, a new row is added. Otherwise, the values of the specified columns are added, modified, or deleted as requested.

Currently, the Table Store writer supports all Table Store data types:

Data Integration data type	Table Store data type
Integer	Integer
Floating point	Double
String	String
Boolean	Boolean
Binary	Binary




**Note:**

You must configure the integer type to Int in script mode so that it can be converted to the integer type for Table Store. If you configure it to the integer type for Table Store, an error is reported in the log and a task failure occurs.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. You can add a data source either in wizard or script mode. The value of this parameter must be identical to the real data source name.	Yes	None
endPoint	The endpoint of the Table Store server.	Yes	None
accessId	The AccessKey ID used to access Table Store.	Yes	None
accessKey	The AccessKey Secret used to access Table Store.	Yes	None

Parameter	Description	Require	Default value
<b>instanceName</b>	<b>The instance name for accessing Table Store. An instance is used for managing Table Store services. After enabling the Table Store service , you need to create an instance on the console before creating and managing tables. Instances are the basic unit for Table Store resource management. Table Store performs application access control and resource usage metering at the instance level.</b>	<b>Yes</b>	<b>None</b>
<b>table</b>	<b>The name of the table to be extracted. You can enter only one table name. Multi-table synchronization is not required for Table Store.</b>	<b>Yes</b>	<b>None</b>

Parameter	Description	Require	Default value
primaryKey	<p>The primary key information of Table Store. Field information is described using the JSON array. Table Store is a NoSQL system, so the corresponding field name must be specified when the Table Store writer imports data.</p> <div> <b>Note:</b> The primary key of Table Store only supports STRING and INT types, so only data of these two types can be entered using the Table Store writer.</div> <p>The data synchronization system supports data type conversion, so the Table Store writer can convert the non-string and non-int source data. An example is described as follows:</p> <pre>"primaryKey" : [   {"name":"pk1", "type":"string"},   {"name":"pk2", "type":"int"} ],</pre>	Yes	None
column	<p>The source table columns to be synchronized. Arrange the column names in a JSON array.</p> <p>The format is as follows:</p> <pre>{"name":"col2", "type":"INT"},</pre> <p>"name" specifies the name of Table Store column to be written, and "type" specifies the type of data to be written. Data types supported by Table Store include STRING, INT, DOUBLE, BOOL, and BINARY.</p>	Yes	None

Parameter	Description	Require	Default value
writeMode	<p>Constants, functions, or custom statements are not supported during the write process. The writing method. The following three methods are supported:</p> <ul style="list-style-type: none"> <li>• <b>Single row operations</b> <ul style="list-style-type: none"> <li>- <b>GetRow:</b> Reads a single row from the table.</li> <li>- <b>PutRow:</b> PutRow API for Table Store, which is used to insert data to a specified row. If this row does not exist, a new row is added. Otherwise, the original row is overwritten. If the row exists, overwrite the original row.</li> <li>- <b>UpdateRow:</b> UpdateRow API for Table Store, which is used to update the data of a specified row. If the row does not exist, a new row is added. If the row exists, the values of the specified columns are added, modified, or deleted as requested.</li> <li>- <b>DeleteRow:</b> Deletes a row from the table.</li> </ul> </li> <li>• <b>Batch operation</b> <p>BatchGetRow: Reads data from multiple rows.</p> </li> <li>• <b>Read range</b> <p>GetRange: Reads data from a table within the specified range.</p> </li> </ul>	Yes	None

Configure the Table Store writer in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the Table Store writer in script mode

**In the following script, a task is configured to write data to Table Store:**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "Parameter": {},
 "name": "Reader ",
 }
]
}
```

```

 "category": "reader"
 },
 {
 "stepType": "ots", // The plug-in name.
 "parameter": {
 "datasource": "", // The data source.
 "column": [// The column.
 {
 "name": "columnName1", // The column name.
 "type": "INT" // The data type.
 },
 {
 "name": "columnName2",
 "type": "STRING"
 },
 {
 "name": "columnName3",
 "type": "DOUBLE"
 },
 {
 "name": "columnName4",
 "type": "BOOLEAN"
 },
 {
 "name": "columnName5",
 "type": "BINARY"
 }
],
 "writeMode": "insert", // The writing method.
 "table": "", // The table name
 "primaryKey": [// The primary key of Table Store.
 {
 "name": "pk1",
 "type": "STRING"
 },
 {
 "name": "pk2",
 "type": "INT"
 }
]
 },
 "name": "Writer",
 "category": "writer"
 }
],
"setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent": 1, // The maximum number of concurrent
threads.
 "dmu": 1 // The number of DMUs.
 }
},
"order": {
 "hops": [
 {
 "from": "Reader",

```

```
 "to": "Writer"
 }
]
}
```

#### 2.8.3.4.19 Configure RDBMS Writer

This topic describes the data types and parameters supported by RDBMS Writer and how to configure it by using the code editor.

RDBMS Writer allows you to write data to tables stored in primary relational database management system (RDBMS) databases. Specifically, RDBMS Writer obtains data from a Data Integration reader, connects to a remote RDBMS database through Java Database Connectivity (JDBC), and runs an `INSERT INTO` statement to write data to the RDBMS database. RDBMS Writer is a common writer for relational databases. To enable RDBMS Writer to support a new relational database, register the driver for the relational database.


RDBMS Writer is designed for extract-transform-load (ETL) developers to import data from data warehouses to RDBMS databases. RDBMS Writer can also be used as a data migration tool by users such as database administrators (DBAs).



##### Data types

RDBMS Writer supports most data types in relational databases, such as numbers and characters. Ensure that your data types are supported.



## Parameters

Parameter	Description	Required	Default value
jdbcUrl	<p>The JDBC connectivity URL, used to connect to the database. The format must be in accordance with official specifications. You can also specify the information of the attachment facility. The format varies with the database type. Data Integration selects an appropriate driver for data reading based on the format.</p> <ul style="list-style-type: none"><li>• <b>Format for DM databases:</b> jdbc:dm://ip:port/database</li><li>• <b>Format for DB2 databases:</b> jdbc:db2://ip:port/database</li><li>• <b>Format for PPAS databases:</b> jdbc:edb://ip:port/database</li></ul>	Yes	None
username	The username used to connect to the database.	Yes	None
password	The password used to connect to the database.	Yes	None
table	The destination table name.	Yes	None
column	<p>The columns in the destination table to which data is written. Separate the columns with a comma (,).</p> <div> <b>Note:</b> We recommend that you do not use the default setting.</div>	Yes	None

Parameter	Description	Required	Default value
preSql	<p>The SQL statement to run before the data synchronization node is run. Currently, you can run only one SQL statement. For example, you can clear outdated data before data synchronization.</p> <div> <b>Note:</b> If you specify multiple SQL statements in the code editor, the system does not ensure that they are run in the same transaction.</div>	No	None
postSql	<p>The SQL statement to run after the data synchronization node is run. Currently, you can run only one SQL statement. For example, you can add a timestamp after data synchronization.</p> <div> <b>Note:</b> If you specify multiple SQL statements in the code editor, the system does not ensure that they are run in the same transaction.</div>	No	None
batchSize	<p>The number of data records to write at a time. Setting this parameter can greatly reduce the interactions between Data Integration and the RDBMS database over the network, and increase the throughput. However, an excessively large value may lead to the out of memory (OOM) error during the data synchronization process.</p>	No	1024

Configure RDBMS Writer by using the code editor

In the following code, a node is configured to write data to an RDBMS database.

```
{
 "job": {
 "setting": {
 "speed": {
 "channel": 1
 }
 }
 }
}
```

```

 },
 "content": [
 {
 "reader": {
 "name": "streamreader",
 "parameter": {
 "column": [
 {
 "value": "DataX",
 "type": "string"
 },
 {
 "value": 19880808,
 "type": "long"
 },
 {
 "value": "1988-08-08 08:08:08",
 "type": "date"
 },
 {
 "value": true,
 "type": "bool"
 },
 {
 "value": "test",
 "type": "bytes"
 }
]
 },
 "sliceRecordCount": 1000
 }
 },
 {
 "writer": {
 "name": "RDBMS Writer",
 "parameter": {
 "connection": [
 {
 "jdbcUrl": "jdbc:dm://ip:port/database",
 "table": [
 "table"
]
 }
],
 "username": "username",
 "password": "password",
 "table": "table",
 "column": [
 "*"
],
 "preSql": [
 "delete from XXX;"
]
 }
 }
 }
]
 }
}

```

**You can enable RDBMS Writer to support a new database as follows:**

1. Go to the directory of RDBMS Writer, `${DATAAX_HOME}/plugin/writer/RDBMS Writer`. In the preceding directory, `${DATAAX_HOME}` indicates the main directory of Data Integration.
2. Add the driver of your database to the drivers array in the `plugin.json` file in the RDBMS Writer directory. RDBMS Writer automatically selects an appropriate driver for connecting to a database.

```
{
 "name": "RDBMS Writer",
 "class": "com.alibaba.datax.plugin.reader.RDBMS writer.RDBMS
writer",
 "description": "useScene: prod. mechanism: Jdbc connection using
the database, execute select sql, retrieve data from the ResultSet
. warn: The more you know about the database, the less problems you
encounter.",
 "developer": "alibaba",
 "drivers": [
 "dm.jdbc.driver.DmDriver",
 "com.ibm.db2.jcc.DB2Driver",
 "com.sybase.jdbc3.jdbc.SybDriver",
 "com.edb.Driver"
]
}
```

3. Add the package of the driver to the libs directory in the RDBMS Writer directory.

```
$tree
.
|-- libs
| |-- Dm7JdbcDriver16.jar
| |-- commons-collections-3.0.jar
| |-- commons-io-2.4.jar
| |-- commons-lang3-3.3.2.jar
| |-- commons-math3-3.1.1.jar
| |-- datax-common-0.0.1-SNAPSHOT.jar
| |-- datax-service-face-1.0.23-20160120.024328-1.jar
| |-- db2jcc4.jar
| |-- druid-1.0.15.jar
| |-- edb-jdbc16.jar
| |-- fastjson-1.1.46.sec01.jar
| |-- guava-r05.jar
| |-- hamcrest-core-1.3.jar
| |-- jconn3-1.0.0-SNAPSHOT.jar
| |-- logback-classic-1.0.13.jar
| |-- logback-core-1.0.13.jar
| |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
| |-- slf4j-api-1.7.10.jar
|-- plugin.json
-- plugin_job_template.json
```

```
-- RDBMS Writer-0.0.1-SNAPSHOT.jar
```

### 2.8.3.4.20 Configure the Stream writer

The Stream writer provides the ability to read data from the reader and print data on the screen or directly discard data. It is mainly applicable to performance testing for data synchronization and basic functional testing.

Parameter description

#### print

- **Description:** Specifies whether to print the output data on the screen.
- **Required:** No
- **Default value:** true.

Configure the Stream writer in wizard mode

**Currently, development in wizard mode is not supported.**

Configure the Stream writer in script mode

**In the following script, a task is configured to read data from the reader and print the data on the screen:**

```
{
 "type": "job",
 "version": "2.0", // The version number.
 "steps": [
 { // The following template is used to configure the reader.
 For more information, see the corresponding section.
 "stepType": "stream",
 "Parameter ": {},
 "name": "Reader ",
 "category": "reader"
 },
 {
 "stepType": "otdsstream", // The plug-in name.
 "parameter": {
 "print": false, // Specifies whether to print output
on the screen.
 "fieldDelimiter": ",", // The column delimiter.
 },
 "name": "Writer",
 "category": "writer"
 }
],
 "setting": {
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
allowed.
 },
 "speed": {
 "throttle": false, // The value false means that the
bandwidth is not throttled. The value true means that the bandwidth
```

```
is throttled. The maximum transmission rate takes effect only if you
specify this parameter as true.
 "concurrent":1, // The maximum number of concurrent
threads.
 "dmu":1// The number of DMUs.
 },
 "order":{
 "hops":[
 {
 "from":"Reader",
 "to":"Writer"
 }
]
 }
}
```

### 2.8.3.5 Optimize synchronization performance

This topic describes how to maximize the synchronization speed by adjusting the concurrency configuration, the difference between nodes that are configured with bandwidth throttling and those that are not, and precautions for custom resource groups.

Data Integration supports real-time and offline data synchronization between any data stores in any location and in any network environment. Data Integration allows you to synchronize dozens of TBs of data between various cloud and local data storage platforms.

Data Integration provides fast data transmission and synchronization between over 400 pairs of heterogeneous data stores. The service can be used to design advanced analysis solutions.

Factors affecting the speed of data synchronization

The factors that affect the speed of data synchronization are listed as follows:

- **Source**
  - **Database performance:** the performance of the CPU, memory module, SSD, network, and hard disk.
  - **Concurrency:** A high concurrency results in a high database workload.
  - **Network:** the bandwidth (throughput) and speed of the network. Generally, a database with better performance can support more concurrent nodes and a larger concurrency value can be set for data synchronization nodes.

- **Synchronization nodes**
  - **Synchronization speed:** whether an upper limit is set for the synchronization speed.
  - **Concurrency:** a maximum number of concurrent threads to read data from source and write data to destination data storage within a single synchronization node.
  - **Nodes that are waiting for resources.**
  - **Bandwidth throttling:** The bandwidth of a single thread is 1,048,576 bit/s . Timeout occurs when the business is sensitive to the network speed. We recommend that you set a smaller value.
  - **Whether to create an index for query statements.**
- **Destination**
  - **Performance:** the performance of the CPU, memory module, SSD, network, and hard disk.
  - **Load:** Excessive load in the destination database affects the write efficiency within the synchronization nodes.
  - **Network:** the bandwidth (throughput) and speed of the network.

You need to monitor and optimize the performance, load, and network of the source and destination databases. The following describes the core settings of a synchronization node.

#### Concurrency

You can configure the concurrency for a node on the codeless UI. The following is an example of how to configure the concurrency in the code editor:

```
"setting": {
 "speed": {
 "concurrent": 10
 }
}
```

#### Bandwidth throttling

By default, bandwidth throttling is disabled. In a synchronization node, data is synchronized at the maximum speed given the concurrency configured for the node . Considering that excessively fast synchronization may overstress the database and thus affect the production, Data Integration allows you to limit the synchronization speed and optimize the configuration as required. If bandwidth throttling

is enabled, we recommend that you limit the maximum speed to 30 Mbit/s. The following is an example for configuring an upper limit for synchronization speed in the code editor, in which the transmission bandwidth is 1 Mbit/s:

```
"setting": {
 "speed": {
 "throttle": true // Indicates whether to throttle the
 transmission rate.
 "mbps": 1, // The synchronization speed.
 }
}
```

**Note:**

- When the throttle parameter is set to false, throttling is disabled, and you do not need to configure the mbps parameter.
- The bandwidth value is a Data Integration metric and does not represent the actual network interface card (NIC) traffic. Generally, the NIC traffic is two to three times of the channel traffic, which depends on the serialization of the data storage system.
- A semi-structured file does not have shard keys. If multiple files exist, you can set the maximum job speed to increase the synchronization speed. However, the maximum job speed is limited by the number of files. For example, the maximum job speed limit is set to  $n$  Mbit/s for  $n$  files. If you set the limit to  $n + 1$  Mbit/s, the synchronization speed remains at  $n$  Mbit/s. If you set the limit to  $n - 1$  Mbit/s, the synchronization is performed at  $n - 1$  Mbit/s.
- A table can be partitioned according to the set maximum job speed only when a maximum job speed and a shard key are configured for a relational database. Relational databases only support numeric shard keys, while Oracle databases support both numeric and string shard keys.

#### Scenarios of slow data synchronization

**Resolve the issue that data synchronization nodes to be run on the default resource group remain waiting for resources.**



- **Example**

When you test the synchronization nodes in DataWorks, one or more nodes remain waiting for resources and an internal system error occurs.

For example, a synchronization node is configured to synchronize data from RDS to MaxCompute by using the default resource group. The node has waited for about 800 seconds before it is run successfully. However, the log shows that the node runs for only 18 seconds and then stops. When you run other nodes for synchronizing hundreds of data records from RDS to MaxCompute by using the default resource group, the nodes remain waiting for resources.

The log is displayed as follows:

```
2017-01-03 07:16:54 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
```

- **Resolution**

The default resource group is not exclusively used by a single user. If resources are insufficient after you start to run a task, the task needs to wait for resources. In this case, the task is completed 800 seconds after you start running the task, but it only takes 10 seconds for the task to be executed.

To improve the synchronization speed and reduce the waiting time, we recommend that you run synchronization nodes during off-peak hours. Typically, most synchronization nodes are run between 00:00 and 03:00.

Accelerate nodes that synchronize data from multiple source tables to the same destination table.

- **Example**

To synchronize data from tables of multiple data stores to a table, you configure multiple synchronization nodes to run in sequence. However, the synchronization takes a long time.

- **Resolution**

To launch multiple concurrent nodes that write data to the same destination database, pay attention to the following points:

- Ensure that the destination database can support the execution of all the concurrent nodes.
- You can configure a synchronization node that synchronizes multiple source tables to the same destination table. Alternatively, you can configure multiple nodes to run concurrently in the same workflow.
- If resources are insufficient, you can configure synchronization nodes to run during off-peak hours.

If no index is added when the WHERE clause is used, an entire table scan slows down the data synchronization.

- **Example**

SQL statement:

```
select bid,inviter,uid,createTime from `relatives` where createTime
>='2016-10-23 00:00:00'and reateTime<'2016-10-24 00:00:00';
```

The synchronization node started to run at 11:01:24.875 on October 25, 2016 and started to return results from 11:11:05.489 on October 25, 2016. The synchronization program is waiting for the database to return SQL query results. However, it takes a long time before MaxCompute can respond.

- **Cause**

When the WHERE clause is used for a query, the createTime column is not indexed, resulting in an entire table scan.

- **Resolution**

We recommend that you use an indexed column or add an index to the column that you want to scan if you use the WHERE clause.

## 2.8.4 Full-database migration

### 2.8.4.1 Overview

This section describes the full-database migration feature in terms of its functions and limits.

Full-database migration is an easy-to-use tool that helps you to improve cost-efficiency. It can quickly upload all the tables in a MySQL database to MaxCompute at a time, saving time that is spent on creating batch tasks for initial data migration to the cloud.

For example, if a database contains 100 tables, you must configure 100 data synchronization tasks in a traditional way. With the full-database migration, you can upload all the tables at a time. However, an upload failure might occur due to the issues that involve the principles of designing database tables.

#### Task generation rules

After the configuration is completed, MaxCompute tables are created and data synchronization tasks are generated based on the selected tables to be synchronized.

The table names, field names, and field types of the MaxCompute tables are generated according to the advanced settings. If no advanced settings are configured, the structure of MaxCompute tables is identical to that of MySQL tables. The partition of these tables is pt, and its format is yyyyymmdd.

The generated data synchronization tasks are daily scheduled tasks and run automatically on the early morning of the next day. The typical transmission rate is 1 Mbit/s, but it varies depending on the synchronization method and concurrency configurations. To customize a data synchronization task, locate the task by choosing clone\_database > Data Source Name > mysql2odps\_table name, and then specify its settings.



#### Note:

We recommend that you perform smoke testing on a data synchronization task on the day when it is generated. To perform smoke testing, choose Administration Center > Task Management > project\_etl\_start > Upload Database > Data Source Name, find the synchronization task, right-click the task, and then test the task.

## Limits

**Full-database migration has the following limits due to the issues that involve the principles of designing database tables.**

- **Currently, only the full-database migration from a MySQL data source to MaxCompute is supported. We are working on support for full-database migration from a Hadoop or Hive data source to Oracle.**
- **Only the daily incremental and daily full upload modes are available.**

**If you want to synchronize historical data at a time, this feature cannot meet your needs. We recommend that:**

- **You configure daily tasks instead of synchronizing historical data at a time . You trace the historical data with the provided retrospective data import feature. This eliminates the need to run temporary SQL tasks to split data after all the historical data is synchronized.**
- **To synchronize historical data at a time, configure a task on the task development page and click Run. Then, data is converted by using SQL statements. They are both one-time operations.**

**If your daily incremental upload task uses a special business logic and cannot be identified by a date field, this feature cannot meet your needs. We provide the following suggestions:**

- **The incremental data upload can be achieved by using two methods: binlog provided by the DTS product and the date field for data changes provided by databases.**

**Currently, Data Integration supports the second method. Therefore, your database must contain the date field for data changes. The system determines whether your data is changed on the same day as the business date by using this field. If yes, all the changed data is synchronized.**

- **To facilitate the incremental data uploading, we recommend that you include the `gmt_create` and `gmt_modify` fields when creating any database tables. Additionally, you can set the `id` field as the primary key to improve efficiency.**

- Full-database migration supports batch upload and full upload modes.

Batch upload is configured with time intervals. Currently, the connection pool protection feature for data sources is not supported, but will be available later.

- To prevent overloads on the database, the full-database migration feature provides the batch upload mode. This mode enables you to upload tables in batches at a specified time interval and prevents compromised service functionality. We provide the following suggestions:

- If you have master and slave databases, we recommend that you synchronize the data of the slave database.

- In a batch upload task, each table has a database connection with a maximum transmission rate of 1 Mbit/s. For example, if you run a synchronization task for 100 tables at a time, 100 database connections are established. We recommend that you specify proper concurrency settings based on your business needs.

- If you have special requirements for transmission efficiency, this feature cannot meet your needs. The maximum transmission of each generated task is 1 Mbit/s.

- Only the mapping of all table names, field names, and field types are supported.

During the full-database migration process, MaxCompute tables are created automatically, where the partition field is pt, the field type is string, and the format is yyyyymmdd.



**Note:**

When you select tables for synchronization, all fields must be synchronized and none of these fields can be edited.

## 2.8.4.2 Migrate a MySQL database

This topic describes how to migrate a MySQL database to MaxCompute.

The database migration feature improves efficiency and reduces costs. It can quickly upload all tables in a MySQL database to MaxCompute. For more information, see [Overview](#).

### Procedure

1. Log on to the DataWorks console.

2. Move the pointer over the DataWorks icon in the upper-left corner, and select Data Integration.
3. In the left-side navigation pane, choose Sync Resources > Connections. On the Connections page, click Add Connection.
4. In the Add Connection dialog box, select MySQL.
5. Add a MySQL connection named clone\_database for database migration.
6. Click Test Connection and verify that the database can be accessed. Click Complete.
7. The added MySQL connection named clone\_database appears in the connection list. Find the added connection and click Migrate Database in the Actions column.

The database migration settings page consists of three functional modules.

Functional module	Description
Tables to migrate	This module lists all the tables in the MySQL database named clone_database. Selected tables will be migrated.
Advanced settings	You can configure the rules for converting the table name, column names, and data types.
Basic settings	You can select whether to synchronize full or incremental data, whether to upload data in one or more batches, and the synchronization efficiency. You can also view the migration progress and results.

8. Click Advanced Settings and configure conversion rules based on your needs. For example, you can add an ods\_ prefix to the name of each MaxCompute table.
9. Specify basic settings. Set Sync Method to Synchronize Incremental Data Daily, and configure the incremental data to be determined based on the gmt\_modified column. Data Integration will generate WHERE clauses based on the specified column and DataWorks scheduling parameters such as \${bdp.system.bizdate}.

Data Integration reads data from MySQL tables by connecting to a remote MySQL database over JDBC and running SELECT statements. Data Integration uses

standard SQL statements, and therefore you can configure WHERE clauses to filter data. The WHERE clause used in this example is provided as follows:

```
STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d') <= gmt_modified AND
gmt_modified < DATE_ADD(STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d'), interval 1 day)
```

Select data upload in batches to protect the MySQL database from being overloaded. Let Data Integration start data synchronization for three tables every one hour from 00:00 each day.

Click Commit. Then, you can view the migration progress and results of each table.

10 Find Table a1 and click View Node to view the migration results.

You have configured a node for migrating a MySQL database named clone\_database to MaxCompute. This node is run based on the specified schedule, daily by default. You can also create retroactive node instances to transmit historical data. The database migration feature of Data Integration significantly simplifies the initial configurations for migrating your data to the cloud and reduces data migration costs.

You can view the migration success logs of Table a1.

### 2.8.4.3 Migrate an Oracle database

This topic describes how to migrate an Oracle database to MaxCompute.

The database migration feature improves efficiency and reduces costs. It can quickly upload all tables in an Oracle database to MaxCompute. For more information, see [Overview](#).

#### Procedure

1. Log on to the DataWorks console.
2. Move the pointer over the DataWorks icon in the upper-left corner, and select Data Integration.
3. In the left-side navigation pane, choose Sync Resources > Connections. On the Connections page, click Add Connection.
4. In the Add Connection dialog box, select Oracle.
5. Add an Oracle connection named clone\_databae for database migration.
6. Click Test Connection and verify that the database can be accessed. Click Complete.

7. The added Oracle connection named `clone_databae` appears in the connection list. Find the added connection and click **Migrate Database** in the **Actions** column.

The database migration settings page consists of three functional modules.

Functional module	Description
Tables to migrate	This module lists all the tables in the Oracle database named <code>clone_databae</code> . Selected tables will be migrated.
Advanced settings	You can configure the rules for converting the table name, column names, and data types.
Basic settings	You can select whether to synchronize full or incremental data, whether to upload data in one or more batches, and the synchronization efficiency. You can also view the migration progress and results.

8. Click **Advanced Settings** and configure conversion rules based on your needs.
9. Set **Sync Method** to **Synchronize All Data Daily**.



**Note:**

If a date column exists in your table, you can select incremental migration and configure the incremental data to be determined based on the date column. Data Integration will generate WHERE clauses based on the specified column and DataWorks scheduling parameters such as `${bdp.system.bizdate}`.

Select data upload in batches to protect the Oracle database from being overloaded. Let Data Integration start data synchronization for three tables every one hour from 00:00 each day.

Click **Commit**. Then, you can view the migration progress and results of each table.

10. Find a related table and click **View Node** to view the node details.

You have configured a node for migrating an Oracle database named `clone_databae` to MaxCompute. This node is run based on the specified schedule, daily by default. You can also create retroactive node instances to transmit historical data. The database migration feature of Data Integration significantly



**simplifies the initial configurations for migrating your data to the cloud and reduces data migration costs.**

## 2.8.5 Best practices

### 2.8.5.1 Synchronize data when the source or destination is deployed on a private network

**This topic describes how to migrate a MySQL database to MaxCompute.**

#### Scenarios

**The complex network environment is divided into the following two scenarios:**

- **Either the source or destination is deployed on a private network.**
  - **VPC (except ApsaraDB for MySQL) <-> Public network**
  - **Finance Cloud <-> Public network**
  - **Data stores hosted on the premises without public IP addresses <-> Public network**
- **Both the source and destination are deployed on private networks.**
  - **VPC (except ApsaraDB for MySQL) <-> VPC (except ApsaraDB for MySQL)**
  - **Finance Cloud <-> Finance Cloud**
  - **Data stores hosted on the premises without public IP addresses <-> Data stores hosted on the premises without public IP addresses**
  - **Data stores hosted on the premises without public IP addresses <-> VPC (except ApsaraDB for MySQL)**
  - **Data stores hosted on the premises without public IP addresses <-> Finance Cloud**

**You can install a Data Integration agent to connect to any network environment and complete data transmission and synchronization in a complex network environment. This topic describes how to synchronize data on a private network. The following sections describe the mechanism and procedure.**

#### Mechanism

**If either the source or destination is located on a private network, install the Data Integration agent on an instance on the same private network. The data stores on the private network are connected to public networks through the agent. Data stores on private networks include:**

- **ECS-hosted data stores without public IP addresses or elastic IP addresses**
- **User-created data stores hosted on the premises without public IP addresses**

ECS-hosted data stores without public IP addresses or elastic IP addresses

- **The ECS2 instance cannot access public networks. Deploy the agent on the ECS1 instance which is located in the same CIDR block as the ECS2 instance and is able to access public networks.**
- **Specify ECS1 as a custom resource group and run the data synchronization node on this resource group.**



**Note:**

**You must authorize the ECS2 instance to access the required data store so that ECS1 can read data from the database. Run the following command to grant permissions to the ECS2 instance:**

```
grant all privileges on *.* to 'demo_test'@'%' identified by 'Password'; -- The percent sign (%) indicates that the specified permissions are granted to all IP addresses.
```

**Authorization is required if a data synchronization node is run on a custom resource group. In this example, create a security group rule on the ECS1 instance for the private and public IP addresses and corresponding ports of ECS2.**

User-created data stores hosted on the premises without public IP addresses

- **Server 1 cannot access public networks. Deploy the agent on server 2 which is located in the same CIDR block as server 1 and is able to access public networks.**
- **Specify server 2 as a custom resource group and run the data synchronization node on this resource group.**

Configure connections

1. **Log on to the DataWorks console as a developer.**
2. **Move the pointer over the DataWorks icon in the upper-left corner, and select Data Integration.**
3. **In the left-side navigation pane, choose Sync Resources > Connections. On the Connections page, click Add Connection.**
4. **In the Add Connection dialog box, select MySQL.**

## 5. Set Connect To to User-Created Data Store without Public IP Addresses in the Add MySQL Connection dialog box.

- Source without public IP addresses

Parameter	Description
Connect To	The connection type. In this example, the MySQL connection type is User-Created Data Store without Public IP Addresses.
Connection Name	The name of the connection. The connection name can contain letters, digits, and underscores (_). It must begin with a letter or an underscore (_), and can be up to 60 characters in length.
Description	The description of the connection. The description can be up to 80 characters in length.
Resource Group	The resource group on which data synchronization nodes that involve this type of connection are run. Each resource group consists of one or more servers.
JDBC URL	The JDBC URL, in the format of jdbc:mysql://ServerIP:Port/Database.
Username and Password	The username and password used to connect to the database.
Test Connection	The connectivity test, which is not supported for data stores without public IP addresses. For these data stores, ignore this feature and click Complete.

- Destination (MaxCompute)

Parameter	Description
Connection Name	The name of the connection. The connection name can contain letters, digits, and underscores (_). It must start with a letter.
Description	The description of the connection. The description can be up to 80 characters in length.
MaxCompute Endpoint	The endpoint of the MaxCompute project. This parameter is read-only, which is automatically obtained from system configurations.
MaxCompute Project Name	The name of the MaxCompute project.

Parameter	Description
AccessKey ID and AccessKey Secret	The logon credentials, which are the account name and the password.
Test Connection	The connectivity test.

Configure a synchronization node

### 1. Select a source.

You must use the code editor to configure data synchronization nodes where the source is located on a network without public IP addresses. You can click **Switch to Code Editor** to switch to the code editor.

### 2. Apply a template.

The parameters are described as follows:

- **Source Connection Type:** The source type is automatically selected based on the connection you have specified on the codeless UI.
- **Target Connection Type:** Select a destination type from the drop-down list.



#### Note:

If the connection to be specified can be added by using the Data Integration service, select the connection from the drop-down list. If not, click **Add Connection** to edit the connection information in JSON.

### 3. Configure the node in the code editor.

**Resource Group:** You can view and change the resource group used to run the data synchronization node. By default, this configuration item is collapsed.

```
{
 "type": "job",
 "configuration": {
 "setting": {
 "speed": {
 "concurrent": "1", // The maximum number of concurrent threads.
 "mbps": "1" // The maximum transmission rate.
 },
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 }
 },
 "reader": {
 "parameter": {
 "splitPk": "id", // The shard key.
 "column": [// The names of the source columns.
 "name",
 "tag",
```

```
 "age",
 "balance",
 "gender",
 "birthday"
],
 "table": "source", // The name of the source table.
 "where": "ds = '20171218'", // The WHERE clause.
 "datasource": "private_source" // The name of the connection,
 which must be identical to the name of the connection you added.
 },
 "plugin": "mysql"
},
"writer": {
 "parameter": {
 "partition": "ds='${bdp.system.bizdate}'", // The partition.
 "truncate": true,
 "column": [// The names of the destination columns.
 "name",
 "tag",
 "age",
 "balance",
 "gender",
 "birthday"
],
 "table": "random_generated_data", // The name of the destination
 table.
 "datasource": "odps_mrtest2222" // The name of the connection,
 which must be identical to the name of the connection you added.
 },
 "plugin": "odps"
}
},
"version": "1.0"
}
```

Run the data synchronization node

**You can run the node by using either of the following methods:**

- Click **Run** on the **Data Integration** page.
- **Schedule the node.**

## 2.8.5.2 Data integration when the networks of both data sources at the source and destination ends are disconnected

Scenarios

**A complex network satisfies either of the following two conditions:**

- **Either the data source at the source or destination end is connected to a private network. For example, data sources at the source and destination end are connected to:**
  - **VPC (except for the RDS) and public network**
  - **Finance Cloud network and public network**
  - **User-created private network and public network**
- **Both data sources at the source and destination ends are connected to private networks. For example, data sources at the source and destination end are connected to:**
  - **VPC (except for the RDS) and VPC (except for the RDS)**
  - **Finance Cloud network and Finance Cloud network**
  - **User-created private network and user-created private network**
  - **User-created private network and VPC (except for the RDS)**
  - **User-created private network and Finance Cloud network**

**The Data Integration service implements stable and efficient data synchronization between homogeneous or heterogeneous data sources even in complicated networks. The following section describes the specific implementation logic and procedures, and assumes that the networks of both data sources are disconnected.**

#### **Implementation logic**

**For the complex networks where both ends are connected to private networks, deploy the Data Integration agents at both ends, with the source agent pushing data to the Data Integration server and the destination agent saving the data to the local device. During data transmission, transmission timeliness and security are ensured by data blocking, compression, and encryption.**

## Procedure

**1. Configure a data source.**

- a. Log on to the DataWorks console as a developer.
- b. In the top navigation bar, hover over the DataWorks icon and click Data Integration. In the left-side navigation pane, choose Sync Resources > Data Source.
- c. Click New Source to show the supported data source types.
- d. Select the data source without a public IP address from the semi-structured FTP data sources.

Add a data source at the source end

Parameter	Description
Data Source Type	The type of the data source. Select Public IP Address Unavailable.
Data Source Name	The data source name. It can contain letters, digits, and underscores (_). It must start with a letter or an underscore (_), and must not exceed 60 characters in length.
Description	The description of the data source, which cannot exceed 80 characters in length.
Resource Group	The resource group. It specifies the machine on which the agent is deployed. The source agent pushes data to the Data Integration server.
Protocol	The protocol. Valid values: ftp   sftp
Host	The host.
Port	The port number. The default FTP port number is 21, and the default SFTP port number is 22.
Username and Password	The username and password used to connect to the database.
Test Connectivity	The connectivity test. Data sources with public IP addresses do not support connectivity tests. Click Finish to complete the configuration.

Add a data source at the destination end by following the similar steps. The resource group specifies the machine on which the agent is deployed. The destination agent saves data to the local device.

## 2. Select the script mode.

- a. In the top navigation bar, click **Data Studio**.
- b. Choose **Create Data Integration Node > Data Synchronization**, and enter a name for the synchronization task.
- c. After selecting the source end and destination end, click **Switch to Script Mode**.

On the script mode page, select an appropriate template that contains key parameters of synchronization tasks, and enter the required information. Note that you cannot switch to wizard mode from the script mode.

### d. Select the ftp-to-ftp import template.

- **Source Type:** The data source name is automatically selected based on the data source selected in the wizard mode.
- **Destination Type:** You select a destination data source from the drop-down list.



#### Note:

If adding data sources on the page is supported by the database, you can select data sources from the template. If not, you must edit relevant data source information in the JSON code section of the template, and then click **Add Data Source**.

### e. Configure the synchronization task.

Configure the resource groups. You can change and view the resource groups for the synchronization task. The default source and destination groups are the resource groups that you selected when adding the data source.

```
{
 "configuration": {
 "setting": {
 "speed": {
 "concurrent": "1", // The number of concurrent threads.
 "mbps": "1" // The maximum transmission rate for the job.
 },
 "errorLimit": {
 "record": "0" // The maximum number of dirty data records
 allowed.
 }
 },
 "reader": {
 "parameter": {
 "fieldDelimiter": ",", // The column delimiter.
 "encoding": "UTF-8", // The encoding.
 "column": // The data source column.
 }
 }
 }
}
```



```

 {
 "index": 0,
 "type": "string",
 },
 {
 "index": 1,
 "type": "string",
 }
],
 "path": "// The file path.
 "/home/wb-zww354475/ww.txt"
],
 "datasource": "lzz_test3"// The data source name, which must
be identical to the name of the added data source.
},
 "plugin": "ftp"
},
 "writer": {
 "parameter": {
 "writeMode": "truncate",// The writing method.
 "fieldDelimiter": ",", // The column delimiter.
 "fileName": "ww",// The file name.
 "path": "/home/wb-zww354475/ww_test",// The file path.
 "dateFormat": "yyyy-MM-dd HH:mm:ss",
 "datasource": "lzz_test4",// The data source name, which must
be identical to the name of the added data source.
 "fileFormat": "csv"// The file type.
 },
 "plugin": "ftp"
 }
},
 "type": "job",
 "version": "1.0",
}

```

### 3. Run a synchronization task.

You can use either of the following two methods to run a synchronization task:

- Click **Run** on the **Data Integration** page.
- **Schedule the task.**

### 2.8.5.3 Incremental data synchronization

Two types of data to be synchronized

Based on whether the data is changed after being written, the data to be synchronized is classified as **unchanged data** (generally log data) and **changing data** (such as the personnel table where the status may change).

Examples

You must specify varying synchronization policies for each type of data. The following example shows how to synchronize the data between an RDS database and MaxCompute.

According to the idempotence (multiple operations of tasks produce the same result. In this way, the task supports re-running scheduling and can easily clear dirty data when an error occurs), data is imported to a separate table or partition, or directly overwrites the historical data in the existing table or partition.

In this example, the test date is November 14, 2016, full data synchronization is performed on the same day, and historical data is synchronized to the partition ds=20161113. In the incremental data synchronization scenario, automatic scheduling is configured to synchronize the incremental data to the partition ds=20161114 on November 15, 2016. The time field optime indicates the time when the data is modified. It helps to determine whether the data is incremental or not.

#### Incremental synchronization of unchanged data

This scenario allows you to perform partitioning easily based on the data generation pattern, because the data remains unchanged after being generated. Typically, you can perform partitioning by date, such as creating one partition on a daily basis.

#### Prepare data

```
drop table if exists oplog;
create table if not exists oplog(
 optime DATETIME,
 uname varchar(50),
 action varchar(50),
 status varchar(10)
);
Insert into oplog values(str_to_date('2016-11-11','%Y-%m-%d'),'LiLei',
', 'SELECT', 'SUCCESS');
Insert into oplog values(str_to_date('2016-11-12','%Y-%m-%d'),'HanMM',
', 'DESC', 'SUCCESS');
```

The two data entries are historical data. Perform full data synchronization first to add the historical data to the partition created yesterday.

#### Procedure

##### 1. Create a MaxCompute table.

```
-- Create a MaxCompute table and partition the table by day.
create table if not exists ods_oplog(
 optime datetime,
 uname string,
 action string,
 status string
```

```
) partitioned by (ds string);
```

## 2. Configure a task to synchronize historical data.

Given that the task is performed only once, only one test is required. After the test is completed, go to the Data Studio page, change the status of the task to suspended (in the rightmost scheduling configuration), and submit and release the task again. The aim is to prevent the task from being scheduled automatically.

View the result of the MaxCompute table.

## 3. Write additional data to the RDS source table as incremental data.

```
insert into oplog values(CURRENT_DATE, 'Jim', 'Update', 'SUCCESS');
insert into oplog values(CURRENT_DATE, 'Kate', 'Delete', 'Failed');
insert into oplog values(CURRENT_DATE, 'Lily', 'Drop', 'Failed');
```

## 4. Configure a task to synchronize the incremental data.



### Note:

If you configure the data filtering parameters, all the data added to the source table on November 14 is retrieved and synchronized to the incremental partition in the target table during the synchronization on the early morning of the next day (November 15).

## 5. View synchronization results.

If you set the task scheduling cycle as daily, the task is scheduled automatically the next day after the task is submitted and released. The data in the MaxCompute destination table is changed as follows once the task runs successfully.

### Incremental synchronization of changing data

For personnel or order tables that stores changing data, full data synchronization on a daily basis is recommended based on the time variable of the data warehouse. In other words, you store full data on a daily basis. In this way, both historical and current data can be retrieved easily.

In actual scenarios, daily incremental data synchronization is required. Because MaxCompute does not support changing data by using the UPDATE statement, you must take other measures to implement the synchronization. The following section describes how to implement full and incremental data synchronization.

## Prepare data

```

drop table if exists user ;
create table if not exists user(
 uid int,
 uname varchar(50),
 deptno int,
 gender VARCHAR(1),
 optime DATETIME
);
-- Historical data
insert into user values (1,'LiLei',100,'M',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (2,'HanMM',null,'F',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (3,'Jim',102,'M',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (4,'Kate',103,'F',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (5,'Lily',104,'F',str_to_date('2016-11-11','%Y-%m-%d'));
-- Incremental data
update user set deptno=101,optime=CURRENT_TIME where uid = 2; --
Change null to non-null.
update user set deptno=104,optime=CURRENT_TIME where uid = 3; --
Change non-null to non-null.
update user set deptno=null,optime=CURRENT_TIME where uid = 4; --
Change non-null to null.
delete from user where uid = 5;
insert into user(uid,uname,deptno,gender,optime) values (6,'Lucy',105,'F',CURRENT_TIME);

```

## Daily full data synchronization

### 1. Create a MaxCompute table.

```

-- Full data
create table ods_user_full(
 uid bigint,
 uname string,
 deptno bigint,
 gender string,
 optime DATETIME
) partitioned by (ds string);ring);

```

### 2. Configure a full data synchronization task.

**Note:**

**Set the scheduling cycle of the task as daily, because daily full data synchronization is required.**

### 3. Test the task, and view the MaxCompute destination table after synchronization.

Because full data synchronization is performed on a daily basis and no incremental synchronization is performed in this case, you can see the following results after the task is automatically scheduled on the next day:

To query the results, set `where ds = '20161114'` to retrieve the full data.

#### Daily incremental data synchronization

This mode is not recommended. You can use this method in the scenarios where DELETE statements are not supported. This is because deleted data cannot be retrieved by filtering conditions of SQL statements. Generally, enterprises' code is deleted logically, in which case UPDATE statements are applied instead of DELETE statements. Using this mode might cause data inconsistency in special scenarios. Another disadvantage is that you must merge new data and historical data after the synchronization.

#### Prepare data

Create two tables, one of which is for writing latest data and the other is for writing incremental data.

```
-- Destination table
create table dw_user_inc(
 uid bigint,
 uname string,
 deptno bigint,
 gender string,
 optime DATETIME
);
-- Incremental data
create table ods_user_inc(
 uid bigint,
 uname string,
 deptno bigint,
 gender string,
 optime DATETIME
)
```

### 1. Configure a task to write full data to the destination table.



#### Note:

Run this task only once and set the task as suspended on the Data Studio page after the task runs successfully.

### 2. Configure a task to write incremental data to the incremental data table.

### 3. Merge the data.

```
insert overwrite table dw_user_inc
select
-- Lists all the SELECT operations. If the ODS table stores certain
data, changes have been made. In this case, make decisions based on
the ODS table.
case when b.uid is not null then b.uid else a.uid end as uid,
case when b.uid is not null then b.uname else a.uname end as uname,
case when b.uid is not null then b.deptno else a.deptno end as
deptno,
case when b.uid is not null then b.gender else a.gender end as
gender,
case when b.uid is not null then b.optime else a.optime end as
optime
from
dw_user_inc a
full outer join ods_user_inc b
on a.uid = b.uid ;
```

**The daily incremental synchronization is different from the daily full synchronization in that the daily incremental synchronization mode synchronizes only a small amount of incremental data with the risk of data inconsistency, and requires extra computing workload for data merging.**

**In most cases, you only need to perform daily full synchronization for changing data. You can also set the lifecycle for historical data to delete the data automatically after the specified period.**

## 2.8.6 FAQs

### 2.8.6.1 What can I do if the status of the node is Pending (Resources)?

#### Symptom

**The node is not functioning properly, and the current instance has no log information recorded. The status of the node is Pending (Resources).**

#### Cause

**The node is configured to use a custom resource group but no custom resource group is available.**

#### Resolution

- 1. Move the pointer over the DataWorks icon and select Operation Center. In the left-side navigation pane, choose Nodes > Recurring. In the DAG, right-click the**

node that is not scheduled as expected and select **View Node Details** to check the resource group used by the node.

2. Move the pointer over the DataWorks icon and select **Project Management**. In the left-side navigation pane, click **Schedule Resources**. On the **Schedule Resources** page, click **Manage Servers**. Check whether the server is stopped or occupied by other nodes.

3. If the issue persists, restart the service by running the following command:

```
su - admin
/home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart
```

### 2.8.6.2 RDS data synchronization fails

#### Description

When data is synchronized from ApsaraDB RDS for MySQL to user-created MySQL databases, error message "DataX cannot connect to the corresponding database" appears.

#### Solution

Take data synchronization from ApsaraDB RDS for MySQL to user-created database as an example. You must complete the following operations:

1. Add a MySQL data source that supports JDBC.
2. Use the new data source to configure and run a synchronization task.

### 2.8.6.3 How do I troubleshoot data integration issues?

If an error occurs with operations performed in Data Integration, locate the fault first. You can check the information related to the error, such as the running resources, connections, and the region where node instances reside.

Check running resources

- If nodes are run on the default resource group, the following information appears in logs:

```
running in Pipeline[basecommon_ group_xxxxxxxxx]
```

- If nodes are run on a custom resource group of Data Integration, the following information appears in logs:

```
running in Pipeline[basecommon_xxxxxxxxx]
```

- If nodes are run on a dedicated resource group of Data Integration, the following information appears in logs:

```
running in Pipeline[basecommon_S_res_group_xxx]
```

Check connection information

**You need to check the following configurations of connections:**

1. Check the names and types of the source and destination connections.
2. Check the network environment.

**Example:** ApsaraDB, connections in connection string mode where Data Integration networks can be directly connected, connections in connection string mode where Data Integration networks cannot be directly connected, RDS or other connections in Virtual Private Cloud (VPC), and connections in Finance Cloud (VPC and the classic network).



### 3. Check whether each data source has passed the connectivity test.

For more information about how to check whether the configurations of connections are correct, see [Data sources](#). Some examples of invalid configurations are as follows:

- Multiple database names are incorrect.
- The entered information contains spaces or special characters.
- Connectivity testing is not supported for the connections, such as the connections in connection string mode where Data Integration networks cannot be directly connected and non-RDS connections in VPCs.

Check the region where node instances reside

Go to DataWorks console, and view the corresponding region, such as China (Shanghai), China (Shenzhen), China (Hong Kong), Singapore, Germany (Frankfurt), and Australia (Sydney). By default, the China (Shanghai) region is selected.



**Note:**

You can view the region only after you have purchased the MaxCompute service.

Copy the error message that appears on the page

If an error occurs, copy the error message and send it to engineers.

Analyze errors in logs

- The data store has failed the connectivity test.

An error message returned because the database cannot be accessed. Database

URL: jdbc:mysql://xx.xx.xx.x:3306/t\_demo. Username: fn\_test. Error

message: Access denied for user 'fn\_test'@'%' to database 't\_demo'.

Check whether the RDS whitelist is properly configured.

Troubleshooting method:

- An error message starting with "Access denied for" returned because the information you specified is incorrect. Check the configurations you specified.
- For example, check whether the RDS whitelist is properly configured and whether the specified account has required database permissions. You can configure the whitelist and permissions in the RDS console.

- The routing policy is incorrect. The node is run on an OXS cluster and an ECS cluster.

```
2017-08-08 15:58:55 : Start Job[xxxxxxx], traceId **running
in Pipeline[basecommon_group_xxx_cdp_oxs]**ErrorMessage:Code:[
DBUtilErrorCode-10]
```

#### Analysis:

The database connection failed. Check the account, password, database name, IP address, port, and the network environment or ask the database administrator for help. The database connection failed because no available JDBC URL can be found in the jdbc:oracle:thin:@xxx.xxxxx.x.xx:xxxx:prod configuration you have made.

- java.lang.Exception: DataX cannot connect to the database.

#### Analysis:

Possible reasons are described as follows:

- The ip, port, database, or jdbc parameter setting is incorrect.
- The authorization failed because the username or password parameter setting is incorrect. Confirm with the database administrator that the configurations of the database are correct.

#### Troubleshooting method:

##### Scenario 1:

- If you need to synchronize data from an Oracle database to an ApsaraDB for PostgreSQL instance, you can only click the Run icon. Recurring tasks are not supported for such synchronization because different resource pools are required.
- When adding an RDS connection, use a normal JDBC URL. Then, Oracle data can be successfully synchronized to ApsaraDB for PostgreSQL.

##### Scenario 2:

- You cannot run nodes related to an ApsaraDB for PostgreSQL instance located in a VPC on custom resource groups. This is because RDS instances use the reverse proxy feature, which can lead to network issues between the RDS instance and your custom resource group. We recommend that you run such nodes on the default resource group. If you need to run such nodes on your

custom resource group, configure the RDS instance as a JDBC data store and create an ECS instance in the same Classless Inter-Domain Routing (CIDR) block.

- The URL of an RDS instance located in a VPC contains an IP address. For example, `jdbc:mysql://100.100.70.1:4309/xxx` where `100.100.70.1` is an IP address. In contrast, the URL of an RDS instance that is not located in a VPC contains a domain name.
- The HBase writer is configured to write data of the Date type.

Synchronize data from an HBase database to another HBase database: 2017-08-15 11:19:29 : State: 4(FAIL) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0 % ErrorMessage:Code:[Hbasewriter-01]

**Analysis:**

The error message returned because you specified an invalid data type.

The HBase writer does not support writing data of the Date type. Currently, it only supports the following data types: String, Boolean, Short, Int, Long, Float, and Double.

**Troubleshooting method:**

- Change the data type. Do not configure the HBase writer to write data of the Date type.
- Change the data type to String. This is because HBase does not support typed values and stores all data as Byte arrays.

- The configurations you specified are not in the correct JSON format.

The column configurations are incorrect.

The intelligent analysis results of DataX show that the most possible cause is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework-02]
```

**Analysis:**

The DataX engine encountered an error when running. For more information, see the diagnostic information prompted when DataX stops running.

```
java.lang.ClassCastException:com.alibaba.fastjson.JSONObject cannot be cast to java.lang.String
```

**Troubleshooting method:**

The configurations you specified are not in the correct JSON format.

```
For the writer:
"column":[
{
"name":"busino",
"type":"string"
}
]
The correct format is provided as follows:
"column":[
{
"busino"
}
]
```

- Square brackets ([]) are missing in the JSON list.

The intelligent analysis results of DataX show that the most possible cause is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework-02]
```

**Analysis:**

The DataX engine encountered an error when running. For more information, see the diagnostic information prompted when DataX stops running.

```
java.lang.String cannot be cast to java.util.List - java.lang.String
cannot be cast to java.util.List
```

```
at com.alibaba.datax.common.exception.DataXException.asDataXException(DataXException.java:41)
```

**Troubleshooting method:**

If square brackets ([]) are missing, a list will be recognized as another data type.

Add square brackets ([]) if necessary.

- You are not authorized to perform related operations.
- You do not have the permission to delete tables.

An error message returned when data is synchronized from MaxCompute to ApsaraDB for MySQL. The error message is as follows:

```
ErrorMessage:Code:[DBUtilErrorCode-07]
```

**Analysis:**

The error message returned because the data cannot be read from the database. Check the column, table, where, and querySql parameters you have configured or ask the database administrator for help.

**SQL statement:**

```
delete from fact_xxx_d where sy_date=20170903
```

**Error message:**

```
DELETE command denied to user 'xxx_odps'@xx.xxx.xxx.xxx' for table 'fact_xxx_d' - com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: DELETE command denied
```

```
to user 'xxx_odps'@xx.xxx.xxx.xxx' for
table 'fact_xxx_d'
```

**Troubleshooting method:**

The error message starting with "DELETE command denied to" indicates that you are not authorized to delete the table. You must be granted the required permission in the database console.

- You do not have the permission to drop tables.

```
Code: [DBUtilErrorCode-07]
```

**Analysis:**

The error message returned because the data cannot be read from the database. Check the column, table, where, and querySql parameters you have configured or ask the database administrator for help.

**SQL statement:** truncate table be\_xx\_ch

**Error message:**

```
DROP command denied to user 'xxx'@[xxx.xx.xxx.xxx](http://
xxx.xx.xxx.xxx)' for table 'be_xx_ch' - com.mysql.jdbc.exceptions
.jdbc4.MySQLSyntaxErrorException: DROP command denied to user 'xxx
'@xxx.xx.xxx.xxx' for table 'be_xx_ch'
```

**Troubleshooting method:**

The error message returned because the TRUNCATE statement is specified in the preSql parameter and you are not authorized to drop the table.

- The whitelist is not properly configured.
- The data store has failed the connectivity test because its whitelist is not properly configured.

**An error occurs because the data store has failed the connectivity test.**

```
error message: **Timed out after 5000** ms while waiting for a
server that matches ReadPreferenceServerSelector{readPreference=
primary}. Client view of cluster state is {type=UNKNOWN, servers
=[{[address:3717=dds-bp1afb47fc7e8e41.mongodb.rds.aliyuncs.com
(http://address:3717=dds-bp1afb47fc7e8e41.mongodb.rds.aliyuncs
.com), type=UNKNOWN, state=CONNECTING, exception={com.mongodb.
MongoSocketReadException: Prematurely reached end of stream}},
{[address:3717=dds-bp1afb47fc7e8e42.mongodb.rds.aliyuncs.com](
http://address:3717=dds-bp1afb47fc7e8e42.mongodb.rds.aliyuncs.
```

```
com), type=UNKNOWN, state=CONNECTING,** exception={com.mongodb.
MongoSocketReadException: Prematurely reached end of stream**}}]
```

**Troubleshooting method:**

The error message starting with "Timed out after 5000" returned when you add MongoDB connections that are not in the VPC because the whitelist is not properly configured.

**Note:**

If you use ApsaraDB for MongoDB, the MongoDB database has a root account by default. For security concerns, Data Integration only supports access to a MongoDB database by using a MongoDB database account. When adding a MongoDB connection, do not use the root account for access.

- The whitelist is incomplete.

```
for Code:[DBUtilErrorCode-10]
```

**Analysis:**

The database connection failed. Check the account, password, database name, IP address, port, and the network environment or ask the database administrator for help.

**Error message:**

```
java.sql.SQLException: Invalid authorization specification,
message from server: "#**28000ip not in whitelist, client ip is xx
.xx.xx.xx". **
2017-10-17 11:03:00.673 [job-xxxx] ERROR RetryUtil - Exception
when calling callable
```

**Troubleshooting method:**

The whitelist is incomplete. You have not added your server IP address to the whitelist.

- The connection configurations are incorrect.
- The connection name is not specified in the code editor.

```
2017-09-06 12:47:05 [INFO] Success to fetch meta data for table
with **projectId [43501]** **project ID **and instanceId **[
mongodb]connection name. **
2017-09-06 12:47:05 [INFO] Data transport tunnel is CDP.
2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info
for 3DES encrypt with parameter account: [zz_683cdbcefb143b7b
709067b362d4385].
```

```
2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info
for 3DES encrypt with parameter account: [zz_683cdbcefba143b7b
709067b362d4385].
[Error] Exception when running task, message:** Configuration
property [accessId]Parameter could not be blank! **
```

#### Troubleshooting method:

If the error message does not contain any **AccessKey**, the data synchronization node is usually configured in the code editor. View the JSON code to check whether you have configured the connection information.

- The connection configurations are incorrect.

```
2017-10-10 10:30:08 INFO
=====
File "/home/admin/synccenter/src/Validate.py", line 16, in notNone
raise Exception("Configuration property [%s] could not be blank!"
% (context))
**Exception: Configuration property [username] could not be blank!
**
```

#### Troubleshooting method:

##### ■ Check with normal logs:

```
[56810] and instanceId(instanceName) [spfee_test_mysql]...
2017-10-09 21:09:44 [INFO] Success to fetch meta data for table
with projectId [56810] and instanceId [spfee_test_mysql].
```

- The logs of ApsaraDB for MySQL show that an error occurs while loading data from data stores and the username parameter returns an empty value. The connection configurations are incorrect.

- The connection to a Distributed Relational Database Service data store times out.

When you synchronize data from MaxCompute to other database, the following error may occur:

```
[2017-09-11 16:17:01.729 [49892464-0-0-writer] WARN CommonRdbm
sWriter$Task
```

Roll back the synchronization, and enable the writer to write only one row at each time.

```
com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: **
Communications link failure **
```



```
The last packet successfully received from the server was 529
milliseconds ago. The last packet sent successfully to the server
was** 528 milliseconds ago**.
```

**Troubleshooting method:**

**The error occurs because the connection to the DataX client times out. When you add connections, add the `? useUnicode=true&characterEncoding=utf-8&socketTimeout=3600000` parameter.**

**Example:**

```
jdbc:mysql://10.183.80.46:3307/ae_coupon? useUnicode=true&
characterEncoding=utf-8&socketTimeout=3600000
```

- An internal system error occurs.

**Troubleshooting method:**

**An internal system error occurs usually because the configurations are not in the correct JSON format in the development environment. If the interface is displayed as blank, you can directly provide the workspace name and the node name to the technical support engineers to fix this problem.**

- Dirty data occurs.
- Empty strings (String[""]) cannot be converted into the Long data type.

```
2017-09-21 16:25:46.125 [51659198-0-26-writer] ERROR WriterRunner
- Writer Runner Received Exceptions:
com.alibaba.datax.common.exception.DataXException: Code:[Common-01
]
```

**Analysis:**

**Dirty data occurs during data synchronization because of infeasible data type conversion. Empty strings (String[""]) cannot be converted into the Long data type.**

**Troubleshooting method:**

**Empty strings (String[""]) cannot be converted into the Long data type. The two tables use the same CREATE TABLE statement. The error is reported because the empty strings cannot be converted into the Long data type. Configure the data type as String.**

- Data is out of valid value range.

```
2017-11-07 13:58:33.897 [503-0-0-writer] ERROR StdoutPlug
inCollector
```

```
Dirty data:
{"exception":"Data truncation: Out of range value for column '
id' at row 1","record":[{"byteSize":2,"index":0,"rawData":-3,"
type":"LONG"}, {"byteSize":2,"index":1,"rawData":-2,"type":"LONG
"}, {"byteSize":2,"index":2,"rawData":"other","type":"STRING"}, {"
byteSize":2,"index":3,"rawData":"other","type":"STRING"}], "type":"
writer"}
```

**Troubleshooting method:**

**The SMALLINT(5) data type allows negative values while the unsigned INT(11) data type does not. For data synchronization between MySQL data stores, dirty data occurs if the source table has a field of the SMALLINT(5) type and the destination table has a field of the unsigned INT(11) data type.**

- **Emoji characters are synchronized.**

**Dirty data occurs during data synchronization that involves a table with emoji characters.**

**Troubleshooting method:**

**Dirty data occurs during data synchronization that involves a table with emoji characters. Change the encoding mode.**

**■ Add connections through JDBC URL.**

```
jdbc:mysql://xxx.x.x.x:3306/database? characterEncoding=utf8&
com.mysql.jdbc.faultInjection.serverCharsetIndex=45
```

**■ Add connections through the ID of the instance.**

**Add ? characterEncoding=utf8&com.mysql.jdbc.faultInjection.**

**serverCharsetIndex=45 after the database name you specified.**

- **Dirty data occurs because of empty columns.**

```
{"exception":"Column 'xxx_id' cannot be null","record":[{"byteSize
":0,"index":0,"type":"LONG"}, {"byteSize":8,"index":1,"rawData":-1
```

```
, "type": "LONG"}, {"byteSize": 8, "index": 2, "rawData": 641, "type": "LONG"}]
```

**The intelligent analysis results of DataX show that the most possible cause is as follows:**

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework-14]
```

#### **Analysis:**

**DataX reports more dirty data than expected. For example, you limit the number of dirty data records to one but seven dirty data records are found. In this case, check the dirty data log information or increase the limit.**

**DataX reports more dirty data than expected. For example, you limit the number of dirty data records to one but seven dirty data records are found.**

#### **Troubleshooting method:**

**According to the code Column 'xxx\_id' cannot be null, the xxx\_id field cannot be left blank. Dirty data occurs if an xxx\_id field value is unspecified. Modify the value or the code.**

- **The data length exceeds the limit imposed by the field.**

```
2017-01-02 17:01:19.308 [16963484-0-0-writer] ERROR StdoutPlug
inCollector
Dirty data:
{"exception":"Data truncation: Data too long for column 'flash' at
row 1","record":[{"byteSize":8,"index":0,"rawData":1,"type":"LONG"},
{"byteSize":8,"index":3,"rawData":2,"type":"LONG"}, {"byteSize":
8,"index":4,"rawData":1,"type":"LONG"}, {"byteSize":8,"index":5
,"rawData":1,"type":"LONG"}, {"byteSize":8,"index":6,"rawData":1,"
type":"LONG"}]
```

#### **Troubleshooting method:**

**According to the code Data too long for column 'flash', the flash field imposes a limit on the data length and a field value exceeds the limit. Modify the data or the field.**

- **The data store is read-only.**

```
2016-11-02 17:27:38.288 [12354052-0-8-writer] ERROR StdoutPlug
inCollector
Dirty data:
{"exception":"The MySQL server is running with the --read-only
option so it cannot execute this statement","record":[{"byteSize":
3,"index":0,"rawData":201,"type":"LONG"}, {"byteSize":8,"index":
1,"rawData":1474603200000,"type":"DATE"}, {"byteSize":8,"index":2,"
```

```
rawData":"12:00 on September 23","type":"STRING"},{"byteSize":5,"index":3,"rawData":"12:00","type":"STRING"}
```

**Troubleshooting method:**

When the data store is read-only, all the data to be synchronized is dirty data.

Change the read-only mode of the data store to read/write.

- An error occurs with the partition.

The setting of the `$[yyyymm]` parameter is invalid. The log is provided as follows:

```
[2016-09-13 17:00:43]2016-09-13 16:21:35.689 [job-10055875] ERROR Engine
```

The intelligent analysis results of DataX show that the most possible cause is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[OdpsWriter-13]
```

**Analysis:**

If an error occurs while running a MaxCompute SQL statement, you can try again. If an error occurs while running SQL statements in the destination MaxCompute table, contact the MaxCompute administrator. The SQL statement is provided as follows:

```
alter table db_rich_gift_record add IF NOT EXISTS
partition(pt='${thismonth}');
```

**Troubleshooting method:**

The relative time parameter `${thismonth}` becomes invalid because it is included in a pair of single quotation marks (`'`). Remove the single quotation marks (`'`).

- The column parameter is not organized in a JSON array.

```
Run command failed.
com.alibaba.cdp.sdk.exception.CDPEException: com.alibaba.fastjson.
JSONException: syntax error, **expect {,** actual error, pos 0
at com.alibaba.cdp.sdk.exception.CDPEException.asCDPEException(
CDPEException.java:23)
```

**Troubleshooting method:**

The configurations are not specified in the correct format. Example:

```
"plugin": "mysql",**
```

```
"parameter": {
 "datasource": "xxxxx",
 ** "column": "uid",**
 "where": "",
 "splitPk": "",
 "table": "xxx"
}
"column": "uid",-----Not organized in an array
```

- **The JDBC URL is not in the correct format.**

**Troubleshooting method:**

**The JDBC URL is not in the correct format. The correct format is `jdbc:mysql://ServerIP:Port/Database`.**

- **A data store fails the connectivity test.**

**Troubleshooting method:**

■ **Check whether the firewall limits the IP address and port in use.**

■ **Check the security group of the port.**

- **An issue related to uid[xxxxxxxx] is logged.**

```
Run command failed.
com.alibaba.cdp.sdk.exception.CDPException: RequestId[F9FD049B-
xxxx-xxxx-xxx-xxxx] Error: CDP server encounter problems, please
contact us, reason: An error occurs while retrieving the network
information of an instance. Check the account which purchases the
RDS instance and the RDS instance name.,uid[xxxxxxxx],instance[rm-
bplcwz5886rmzio92]ServiceUnavailable : The request has failed due
to a temporary failure of the server.
```

```
RequestId : F9FD049B-xxxx-xxxx-xxx-xxxx
```

**Troubleshooting method:**

If the preceding error occurs when you synchronize data from ApsaraDB for RDS to MaxCompute, you can copy RequestId: F9FD049B-xxxx-xxxx-xxx-xxxx to the ApsaraDB for RDS engineers.

- The query parameter is invalid for MongoDB.

The following error message returned when you synchronize data from a MongoDB database to a MySQL database. The reason is that the query parameter is not in the correct JSON format.

```
Exception in thread "taskGroup-0" com.alibaba.datax.common.
exception.DataXException: Code:[Framework-13]
```

**Analysis:**

The DataX engine encountered an error when running. For more information, see the diagnostic information prompted when DataX stops running.

```
org.bson.json.JsonParseException: Invalid JSON input. Position: 34
. Character: '.'.
```

**Troubleshooting method:**

■ **Invalid example:** "query": "{ 'update\_date': { '\$gte': new Date().valueOf()  
( )/1000 } } }". Parameters such as new Date() are not supported.

■ **Valid example:** "query": "{ 'operationTime' { '\$gte': ISODate(' \${last\_day}  
}T00:00:00.424+0800' ) } } }".

- The memory is insufficient.

```
2017-10-11 20:45:46.544 [taskGroup-0] INFO TaskGroupContainer -
taskGroup[0] taskId[358] attemptCount[1] is started
```

```
Java HotSpot™ 64-Bit Server VM warning: INFO: os::commit_memory
(0x000007f15ceaeb000, 12288, 0) failed; error='**Cannot allocate
memory'** (errno=12)
```

**Troubleshooting method:**

**The memory is insufficient. If you run a node on custom resources, you need to add memory. If you run a node on the resources provided by Alibaba Cloud , open a ticket.**

- **The max\_allowed\_packet parameter is set to an improper value.**

**Error message:**

```
Packet for query is too large (70 > -1). You can change this
value on the server by setting the max_allowed_packet' variable
. - **com.mysql.jdbc.PacketTooBigException: Packet for query is
too large (70 > -1). You can change this value on the server by
setting the max_allowed_packet' variable. **
```

**Troubleshooting method:**

- **The max\_allowed\_packet parameter defines the maximum length of the communication buffer. A MySQL data store drops packets whose size is larger than the value of this parameter. Therefore, large inserts and updates will fail.**
- **If the value of the max\_allowed\_packet parameter is excessively large, change it to a smaller value. Usually, set it to 10 MB (10 × 1024 × 1024 Bytes ).**
- **HTTP status code 500 is logged because logs cannot be retrieved.**

```
Unexpected Error:
Response is com.alibaba.cdp.sdk.util.http.Response@382db087[proxy
=HTTP/1.1 500 Internal Server Error [Server: Tengine, Date: Fri,
27 Oct 2017 16:43:34 GMT, Content-Type: text/html;charset=utf-8,
Transfer-Encoding: chunked, Connection: close,
HTTP Status 500 - Read timed out**type** Exception report**
message***+Read timed out***description***+The server encountered
an internal error that prevented it from fulfilling this request.
***exception**
java.net.SocketTimeoutException: Read timed out
```

**Troubleshooting method:**

**HTTP status code 500 is logged while your nodes are running on resources provided by Alibaba Cloud. Data Integration fails to retrieve logs. In this case,**

contact Alibaba Cloud technical support. If the nodes are running on custom resources provided by other vendors, rerun the alisa command.

**Note:**

If you refresh the page and the node is still stopped, you can switch to the admin account and rerun the following alisa command: `/home/admin/alisa/tasknode/target/alisa/tasknode/bin/serverctl restart`.

- The `hbase.zookeeper.quorum` parameter setting for the HBase writer is invalid.

```
2017-11-08 09:29:28.173 [61401062-0-0-writer] INFO ZooKeeper -
Initiating client connection, connectString=xxx-2:2181,xxx-4:2181
,xxx-5:2181,xxxx-3:2181,xxx-6:2181 sessionTimeout=90000 watcher=
hconnection-0x528825f50x0, quorum=node-2:2181,node-4:2181,node-5:
2181,node-3:2181,node-6:2181, baseZNode=/hbase
Nov 08, 2017 9:29:28 AM org.apache.hadoop.hbase.zookeeper.
RecoverableZooKeeper checkZk
WARNING: **Unable to create ZooKeeper Connection**
```

**Troubleshooting method:**

■ Invalid example: `"hbase.zookeeper.quorum":"xxx-2,xxx-4,xxx-5,xxxx-3,xxx-6"`

■ Valid example: `"hbase.zookeeper.quorum":"Your ZooKeeper IP address"`

- The specified directory is empty.

The intelligent analysis results of DataX show that the most possible cause is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[
HdfsReader-08]
```

**Analysis:**

The specified directory is empty. The files to be read cannot be found. Check your configurations.

```
path:/user/hive/warehouse/tmp_test_map/*
```



```
at com.alibaba.datax.common.exception.DataXException.asDataXException(DataXException.java:26)
```

**Troubleshooting method:**

Check the files based on the directory provided. If files still cannot be found, configure the files.

- The table does not exist.

The intelligent analysis results of DataX show that the most possible cause is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[MySQLErrCode-04]
```

**Analysis:**

The table does not exist. Check the table name or contact the database administrator to check whether the table exists.

Table name: xxxx.

Run the following SQL statement: `select * from xxxx where 1=2;`

**Error message:**

```
Table 'darkseer-test.xxxx' doesn't exist - com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Table 'darkseer-test.xxxx' doesn't exist
```

**Troubleshooting method:**

Run the `select * from xxxx where 1=2` SQL statement to check whether the table has any errors. Do appropriate operations on the table if any errors exist.

## 2.8.6.4 Data synchronization task failure when the column name of the synchronized table is a keyword

**Problem**

When you perform a synchronization task, the task fails because the column name of the synchronized table is a keyword.

**Solution**

Take MySQL data source as an example:

**1. Create a new table with the name of aliyun, and the statement is as follows:**

```
create table aliyun (`table` int ,msg varchar(10));
```

**2. Create a view and assign an alias to the table column.**

```
create view v_aliyun as select `table` as col1,msg as col2 from aliyun;
```

**Note:**

- The word **table** is a MySQL keyword. In this case, a code error is reported when data is synchronized. To prevent such errors, create a view and assign an alias to the table column.
- We do not recommend that you use a keyword as a column name for a table.

**3. The preceding statement assigns an alias to a column that has a keyword.**

Therefore, when configuring a data synchronization task, you can replace the **aliyun** table with the **v\_aliyun** view.

**Note:**

- The escape character for MySQL is 'key'.
- The escape characters for Oracle and PostgreSQL are "keywords".

## 2.8.6.5 Customize a table name for the data synchronization task

### Background information

The tables are identified by days (such as **orders\_20170310**, **orders\_20170311**, and **orders\_20170312**) on a one-table-for-one-day basis with the same table structure.

### Purpose

To create only one data synchronization task with a custom table name to read and write the table data of the previous day from the source database into MaxCompute at the early morning of each day. For example, on March 15, 2017, data in the **orders\_20170314** table is read automatically from the source database and imported.

### Implementation

1. Log on to the DataWorks console and navigate to the Data Integration page.

2. Create a data synchronization task in wizard mode, and specify a table name when configuring the data source table, such as orders\_20170310. Configure and save the synchronization task by following the normal procedure.
3. Click Switch to Script Mode to switch to the script mode.
4. Use a variable as the name of the source table in the script mode, such as orders\_`\${tablename}`.

Assign the variable "tablename" a value in parameter settings of the task. The table names are identified by days, and the purpose is to read the table of the previous day. Therefore, the assigned value is \$yyyymmdd-1.



**Note:**

You can also use orders\_`\${bdp.system.bizdate}` as the variable to name the source table.

After completing the configuration above, save and submit the task before proceeding.

### 2.8.6.6 The specified encoding is incorrect

A data synchronization task may fail with dirty data generated, or it succeed with data garbled. This can be caused by an incorrect encoding setting.

A data synchronization task fails with dirty data generated

#### Symptom

A data synchronization task fails and dirty data is generated because the specified encoding is incorrect. The error log is shown as follows:

```
016-11-18 14:50:50.766 13350975-0-0-writer ERROR StdoutPluginCollector
- Dirty data:

{"exception":"Incorrect string value: '\\xF0\\x9F\\x98\\x82\\xE8\\xA2...' for column 'introduction' at row 1","record":[{"byteSize":8,"index":0,"rawData":9642,"type":"LONG"}, {"byteSize":33,"index":1,"rawData":" Hello world! (http://docs.aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/pic/56134/cn_zh/1498728641169/%E5%9B%BE%E7%89%877.png) ","type":"STRING"}, {"byteSize":8,"index":4,"rawData":0,"type":"LONG"}],"type":"writer"}
2016-11-18 14:50:51. 265 [13350975-0-0-writer] warn maid $ task-roll back this write, commit by writing one row at a time. Because: Java. SQL. batchupdateexception: incorrect string value: '\\ xq0 \\ x9f \\ x88 \\ xB6 \\ XeF \\ xB8...' For column' introduction 'at Row 1
```

#### Cause

**The encoding specified for the data source is not utf8mb4. Only the utf8mb4 encoding supports emoji characters.**

#### **Solution**

- **When you add a data source over a JDBC connection, you need to select utf8mb4 as the encoding. An example setting is `jdbc:mysql://xxx.x.x.x:3306/database?com.mysql.jdbc.faultInjection.serverCharsetIndex=45`. Then, emoji characters can be properly synchronized.**
- **Change the data source encoding to utf8mb4. For example, modify the encoding of an RDS instance in the RDS console.**

A data synchronization task succeed with data garbled

#### **Symptom**

**A data synchronization task succeeds but data is garbled.**

#### **Cause**

**Three possible causes are listed as follows:**

- **The source data is garbled.**
- **The specified encoding is different between the reader and the writer.**
- **The browser encoding is different from the data source encoding, and therefore the preview fails or the preview is garbled.**

#### **Solution**

**The solution varies with the cause.**

- **If the source data is garbled, re-process the source data and then start a data synchronization task.**
- **If the data source encoding is different from the client encoding, correct the settings so that the data source encoding is the same as the client encoding.**
- **If the browse encoding is different from the data source encoding, correct the settings so that the browser encoding is the same as the data source encoding.**

### **2.8.6.7 The specified data types are supported for full database migration**

**Currently, full database migration can only be performed from MySQL databases including ApsaraDB RDS for MySQL databases to MaxCompute. You can enter the**

full database migration page by clicking Migrate Database in the Actions column of the data source in the Data Integration service.

The data types supported in Advanced Settings for full database migration are described as follows:

The data types supported by source MySQL database for full database migration include TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, YEAR, FLOAT, DOUBLE, DECIMAL, DATE, DATETIME, TIMESTAMP, TIME, and BOOL.

The data types supported by the destination MaxCompute database are BIGINT, STRING, DOUBLE, DATETIME, and BOOLEAN.

All the data types that are supported by MySQL databases can be converted to data types supported by MaxCompute.



**Note:**

Binary numbers with more than 2 bits in MySQL databases cannot be converted to BIGINT, STRING, DOUBLE, DATETIME, and BOOLEAN. The 1-bit binary numbers are converted to BOOLEAN values.

## 2.9 Data Quality

### 2.9.1 Overview

DataWorks provides a Data Quality service for you to control the data quality of disparate connections. In Data Quality, you can check data quality, configure alert notifications, and manage connections.

Relying on DataWorks, Data Quality provides a comprehensive data quality solution that has various features. For example, you can detect data, compare data, monitor data quality, and use intelligent alerting.

Data Quality monitors data in datasets. Currently, it allows you to monitor MaxCompute tables and DataHub topics. When offline MaxCompute data changes, Data Quality checks data and blocks nodes if it detects exceptions. This prevents nodes from being affected. In addition, Data Quality allows you to manage the check result history so that you can analyze and evaluate the data quality.

For streaming data, Data Quality uses DataHub to monitor data streams and sends alert notifications to subscribers if it detects stream discontinuity. You can also set the alert severity such as warning and error alerts, and the alert frequency to minimize repeated alerts.



**Note:**

Data Quality monitors the quality of data from MaxCompute and DataHub datasets. To use Data Quality features, you need to create tables and write data to the tables. You can create MaxCompute tables and write data to the tables in the MaxCompute console or in the DataWorks console.

To go to the Data Quality page, do as follows: Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner, and click Data Quality.

## 2.9.2 Features

### 2.9.2.1 Dashboard

As the homepage of Data Quality, the Dashboard page displays an overview of alerts and blocks for subscribed nodes. You can set filter conditions to view the required alerts and blocks.

Card	Description
My MaxCompute Partition Subscriptions	Displays the number of MaxCompute partitions with alerts or blocks and the number of normal MaxCompute partitions on the current day. You can click this card to go to the Search by Node page for the MaxCompute connection and view alert details.
My DataHub Topic Subscriptions	Displays the number of DataHub topics with alerts and the number of normal DataHub topics on the current day . You can click this card to go to the Search by Node page for the DataHub connection and view alert details.
Current task alarm condition	Displays alerts for MaxCompute and DataHub connections of the current workspace on the current day.
Current task blocking situation	Displays blocks for the MaxCompute connection of the current workspace on the current day.
Task Alarm Situation Trend	Displays the trend chart of alerts for MaxCompute and DataHub connections. You can view the alert trend in the past 7 or 30 days, or a custom time period within the past three months.

Card	Description
Task Blocking Situation Trend Graph	Displays the trend chart of blocks for MaxCompute and DataHub connections. You can view the block trend in the past 7 or 30 days, or a custom time period within the past three months.

### 2.9.2.2 My Subscriptions

The My Subscriptions page displays all nodes subscribed by your account.

Currently, Data Quality allows you to monitor MaxCompute tables and DataHub topics. You can select a connection on the My Subscriptions page and search for subscribed nodes of the connection.

You can search for nodes of the following two connections:

- Select MaxCompute.
  - You can click a partition expression on the right to go to the Rules page.
  - You can click View Check Results in the Actions column for a partition expression to go to the Search by Node page.
  - Data Quality supports the following four notification methods: Email, Email and SMS, DingTalk Chatbot, and DingTalk Chatbot @ALL.
  - You can click Cancel Subscription in the Actions column for a partition expression to unsubscribe from the partition expression.
- Select DataHub.
  - You can click View Monitoring Rules in the Actions column for a topic to go to the Rules page.
  - You can click Cancel Subscription in the Actions column for a topic to unsubscribe from the topic.

### 2.9.2.3 Rules

Currently, Data Quality allows you to monitor MaxCompute tables and DataHub topics. This topic describes how to configure rules for MaxCompute monitoring.

- On the Rules page, select MaxCompute from the drop-down list in the upper-left corner. All tables of the MaxCompute connection are listed on the right. You can also search for tables in the search box.

- On the Rules page, select DataHub from the drop-down list in the upper-left corner. All topics of the DataHub connection are listed on the right. You can also search for topics in the search box.

After selecting MaxCompute, click View Monitoring Rules for a table to go to the Rules page. On the Rules page, you can configure rules for the table.

Currently, Data Quality allows you to configure template rules and custom rules for a table.


**Note:**

Before configuring a template rule for a table, you need to configure a partition expression. For more information, see [Configure partition expressions](#).

### Template rules


You can click Create Monitoring Rule or Quick Create to create a template rule.

- Create Monitoring Rule

Parameter	Description
Rule Name	The name of the rule.
Rule Type	<p>The type of the rule. Valid values: Strong or Weak.</p> <ul style="list-style-type: none"><li>- If you select Strong, error alerts are reported and descendant nodes are blocked, whereas warning alerts are reported but descendant nodes are not blocked.</li><li>- If you select Weak, error alerts are reported but descendant nodes are not blocked, whereas warning alerts are not reported and descendant nodes are not blocked.</li></ul>
Field	The fields to be monitored. You can select All Fields in Table or specify fields. If you specify fields, you can specify fields of a numeric type or non-numeric type.
Template	<p>The template to apply to the rule. Currently, Data Quality supports 37 rule templates.</p> <div> <b>Note:</b> You can set field-specific rules of the average value, accumulated value, minimum value, and maximum value only for numeric fields.</div>



Parameter	Description
Comparison Method	The comparison method of the rule. Valid values: Absolute Value, Raise, and Drop.

Parameter	Description
Thresholds	<ul style="list-style-type: none"> <li>- You can calculate the fluctuation by using the following formula:   <math display="block">\text{Fluctuation} = (\text{Sample} - \text{Baseline}) / \text{Baseline}</math> </li> <li>- You can calculate the fluctuation variance only for numeric fields such as Bigint and Double fields. The formula is as follows:   <math display="block">\text{Fluctuation variance} = (\text{Sample} - \text{Baseline}) / \text{Standard deviation}</math> </li> </ul> <div data-bbox="560 696 1433 1400" style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p> <b>Note:</b> The sample and baseline are defined as follows:</p> <ul style="list-style-type: none"> <li>- <b>Sample:</b> the sample value for the current N days. For example, if you want to check the fluctuation of table rows on an SQL node in a day, the sample is the number of table rows on that day.</li> <li>- <b>Baseline:</b> the comparison value from the previous N days. For example: <ul style="list-style-type: none"> <li>■ If you want to check the fluctuation of table rows on an SQL node in a day, the baseline is the number of table rows on the previous day.</li> <li>■ If you want to check the fluctuation of the average number of table rows on an SQL node in seven days, the baseline is the average number of table rows in the previous seven days.</li> </ul> </li> </ul> </div> <p>You can set Warning Threshold and Error Threshold to monitor data at different severities:</p> <ul style="list-style-type: none"> <li>- If the fluctuation does not exceed the warning threshold , Data Quality determines that data is normal.</li> <li>- If the fluctuation exceeds the warning threshold but does not exceed the error threshold, Data Quality reports a warning alert.</li> <li>- If the fluctuation exceeds the error threshold, Data Quality reports an error alert.</li> <li>- If you do not specify the warning threshold, Data Quality reports error alerts or normal based on the check result .</li> <li>- If you do not specify the error threshold, Data Quality reports warning alerts or normal based on the check result.</li> </ul>
1052	<ul style="list-style-type: none"> <li>- If you specify neither the warning threshold nor the error threshold, Data Quality reports error alerts if it detects exceptions. However, you must specify at least</li> </ul>

- Quick Create

Parameter	Description
Rule Name	The name of the rule.
Field	The fields to be monitored. You can select All Fields in Table or specify fields. If you specify fields, you can specify fields of a numeric type or non-numeric type.
Trigger	The trigger condition of the rule. If you select All Fields in Table for Field, you can select only The number of rows is greater than 0.

#### Custom rules

If template rules do not meet your requirements for monitoring the data quality, you can create custom rules.

You can click **Create Monitoring Rule** or **Quick Create** to create a custom rule.

- Create Monitoring Rule

Parameter	Description
Rule Name	The name of the rule.
Field	<p>The fields to be monitored. Valid values: All Fields in Table, SQL Statement, and specific fields.</p> <ul style="list-style-type: none"><li>- If you select All Fields in Table or specify fields, you can specify the WHERE clause to customize filter conditions based on business requirements.</li><li>- If you select SQL Statement, you can customize the SQL logic to set a rule. The return value is the value in a row of a column.</li></ul>
Rule Type	<p>The type of the rule. Valid values: Strong or Weak.</p> <ul style="list-style-type: none"><li>- If you select Strong, error alerts are reported and descendant nodes are blocked, whereas warning alerts are reported but descendant nodes are not blocked.</li><li>- If you select Weak, error alerts are reported but descendant nodes are not blocked, whereas warning alerts are not reported and descendant nodes are not blocked.</li></ul>

Parameter	Description
Sampling Method	The sampling method of the rule. Valid values: sum, max, min, and avg.
Filter	The filter condition of the rule. For example, if you want to query partitions of the table based on a specific data timestamp, you can specify the WHERE clause as follows: <code>pt=\${yyyymmdd-1}</code> .
Threshold Type	The threshold type of the rule. Valid values: Value and Fluctuation.
Comparison Method	The comparison method of the rule. Valid values: Greater Than, Greater Than or Equal To, Equal To, Unequal To, Less Than, and Less Than or Equal To.
Compare To	The comparison value of the rule. Currently, the comparison value must be a constant.
Expected Value	The expected value of the rule.
Description	The description of the rule.

- Quick Create

Parameter	Description
Rule Name	The name of the rule.
Rule Type	The type of the rule. You can select only Values Duplicated in Multiple Fields.
Field	The fields to be monitored.

## 2.9.2.4 Search by Node

The Search by Node page displays the monitoring results of MaxCompute partitions and DataHub topics. After monitoring rules are run, you can view monitoring results on this page.

View MaxCompute monitoring results

On the Search by Node page, you can search for MaxCompute monitoring results by metrics such as Status, Table Name, and Node ID. You can click an action in the Actions column to go to the corresponding page and perform operations as required.

- **Node ID:** the ID of the node that triggers rules.

- **Run At:** the runtime of rules.
- **Status:** the monitoring result of rules. Pay attention to partitions that trigger alerts or blocks.
- **Actions:**
  - **Details**
    - **More:** Click it to view more information about the node instance, such as the connection, workspace name, node ID, and owner.
    - **View History Check Results:** Click it for a rule to view the check result history of the rule.
  - **Rules:** Click it for a table to go to the Rules page. On this page, you can view partition expressions and rules created for the table, and modify rules as required. For more information, see [MaxCompute monitoring](#).
  - **View Log:** Click it to view the runtime logs of rules.
  - **View Statistics:** Click it for a node to view the data volume and the number of table rows each time the node is run after it was created.

View DataHub monitoring results

On the **Search by Node** page, you can search for DataHub monitoring results by metrics such as Status, Table Name, and Node ID. You can click an action in the **Actions** column to go to the corresponding page and perform operations as required.

**Actions:**

- **View Log:** Click it to view the runtime logs of rules.
- **Alerts:** Click it for a topic to view alert details. You can cancel alerts for the topic on the Alerts page.

## 2.9.3 User guide

### 2.9.3.1 MaxCompute monitoring

The Rules page is the most important part of Data Quality, where you can configure monitoring rules. Currently, Data Quality allows you to monitor MaxCompute tables

and DataHub topics. This topic describes how to configure rules for MaxCompute monitoring.

#### Add a connection

Before configuring monitoring rules, you need to go to the Data Integration page to add a connection. For more information, see [Add a MaxCompute connection](#). After the connection is added, you can go to the Data Quality page to configure monitoring rules.

#### Select a connection

1. In the left-side navigation pane, click Rules.
2. On the Rules page, select MaxCompute from the drop-down list in the upper-left corner. All tables of the MaxCompute connection are listed on the right.

On the Tables page, you can search for a table by table name. Fuzzy search based on the initial letter is supported.

3. Find the target table and click View Monitoring Rules in the Actions column.

#### Configure partition expressions

In Data Quality, you need to configure rules based on a partition expression.



#### Note:

- To configure rules for a non-partitioned table, you can specify NOTAPARTITIONTABLE as the partition expression.
- To configure rules for a partitioned table, you can specify a data timestamp expression, such as `$[yyyymmdd]`, or a regular expression as the partition expression.

On the Rules page of the target table, click + in the upper-left corner to add a partition expression.

- Add partition expressions: Click + in the upper-left corner. In the Add Partition dialog box that appears, specify a partition expression as needed. For a non-

partitioned table, select NOTAPARTITIONTABLE from the recommended partition expressions.

- For a table with only one partition, follow the format: Partition key=Partition value. The partition value can be either a constant or a system parameter. You must configure partition expressions through the last partition.
- For a table with multiple partitions, follow the format: Partition key 1=Partition value/Partition key 2=Partition value/Partition key N=Partition value. Each partition value can be either a constant or a system parameter. You must use brackets ([]) to indicate a parameter, such as \$[yyyymmdd-N].

The data timestamp configured in a partition expression also determines the recurrence of the partition expression. For example, if the data timestamp is the date of five days ago, the partition expression is triggered every five days. The following table lists supported partition expressions.

Partition expression	Description
dt=\$[yyyymmdd-N]	Indicates N days before the date specified by the data timestamp.
dt=\$[yyyymm01-1]	Indicates the first day of each month.
dt=\$[yyyymm01-Nm]	Indicates the first day of the month that is N months before the month specified by the data timestamp.
dt=\$[yyyymmld-1]	Indicates the last day of each month.
dt=\$[yyyymmld-1m]	Indicates the last day of the month that is N months before the month specified by the data timestamp.
dt=\$[hh24miss-1/24]	Indicates one hour before the hour specified by the data timestamp.
dt=\$[hh24miss-30/24/60]	Indicates half an hour before the hour specified by the data timestamp.
\$[yyyymmdd]	Indicates the data timestamp of recurrence.
\$[yyyymmdd-1]	Indicates one day before the data timestamp of the current instance.

Partition expression	Description
<code>\$[yyyymmddhh24miss]</code>	Indicates the data timestamp of the current instance. The format <code>yyyymmddhh24miss</code> is described as follows: <ul style="list-style-type: none"><li>- <code>yyyy</code> indicates a four-digit year.</li><li>- <code>mm</code> indicates a two-digit month.</li><li>- <code>dd</code> indicates a two-digit day.</li><li>- <code>hh24</code> indicates a two-digit hour (24-hour clock).</li><li>- <code>mi</code> indicates two-digit minutes.</li><li>- <code>ss</code> indicates two-digit seconds.</li></ul>
<code>NOTAPARTITIONTABLE</code>	Indicates the partition expression of a non-partitioned table.

- **Select recommended partition expressions:** For example, the following procedure describes how to select a recommended partition expression for the partition key `dt`. We recommend that you do not specify a regular expression as the partition expression for a dynamic partitioned table.
  1. In the Add Partition dialog box, click the Partition Expression field. A drop-down list appears to show you the partition expressions recommended by Data Quality.
    - Select a recommended partition expression if it meets your expectation.
    - Customize a partition expression if no recommended partition expressions meet your expectation.
  2. After you enter a partition expression, click Verify. Data Quality uses the current time, that is, the data timestamp, to calculate data and verify the partition expression.
  3. Click OK.
- **Delete partition expressions:** You can delete a partition expression as required. When you delete a partition expression, all rules configured based on the partition expression are also deleted.

Manage linked nodes

To monitor the quality of data involved in a node, you need to link a partition expression to the node.



**Note:**

- The Manage Linked Nodes dialog box lists all committed nodes. Data Quality allows you to link a partition expression to a node in another workspace.
- Before linking a partition expression to a node in another workspace, ensure that you are an administrator, a developer, or an administration expert in two workspace.

You can link a partition expression to one or more nodes. After nodes are linked, Data Quality can automatically monitor linked nodes.

**Note:**

Data Quality allows you to flexibly link a partition expression to a node. You can select a node that is not related to your table.

### Create rules

The Rules page is the most important part of Data Quality, where you can create rules for your tables.

Currently, Data Quality allows you to create template rules and custom rules as needed. To create a template rule or a custom rule, you can click **Create Monitoring Rule** or **Quick Create**.

After rules are configured, you can click **Save** to save all the configured rules for the current partition expression.

Configuration method	Parameter	Description
Create Monitoring Rule	Rule Name	The name of the rule.
	Field	The fields to be monitored . You can select <b>All Fields in Table</b> or specify fields . If you specify fields, you can apply the rule to the specified fields in the table. For example, select <b>All Fields in Table</b> and set other parameters for the table-specific rule.
	Template	The built-in template to apply to the rule.

Configuration method	Parameter	Description
	Comparison Method	The comparison method of the rule. Valid values: Absolute Value, Raise, and Drop.
	Rule Type	<p>The type of the rule. Valid values: Strong or Weak.</p> <ul style="list-style-type: none"> <li>• Strong: If a node reaches the error threshold, Data Quality reports an error alert and determines that the node fails. If a node reaches the warning threshold, Data Quality reports a warning alert and determines that the node is successful.</li> <li>• Weak: If a node reaches the error threshold, Data Quality reports an error alert and determines that the node is successful. If a node reaches the warning threshold, Data Quality does not report a warning alert and determines that the node is successful.</li> </ul>
	Thresholds	The warning threshold and error threshold of the fluctuation. You can adjust the slider to specify thresholds or directly enter thresholds.
	Description	The description of the rule.
Quick Create	Rule Name	The name of the rule.

Configuration method	Parameter	Description
	Field	The fields to be monitored . You can select All Fields in Table or specify fields . If you specify fields, you can apply the rule to the specified fields in the table.
	Trigger	The trigger condition of the rule. <ul style="list-style-type: none"><li>• If you select All Fields in Table for Field, you can select only The number of rows is greater than 0.</li><li>• If you specify fields for Field, you can select Field Values Duplicated or Null Strings.</li></ul>

## Test rules

After rules are configured for a partition expression, you can test all these rules and view the test results.



### Note:

By testing monitoring rules, you can manually run these rules to test their configuration and notification methods. We recommend that you test rules as required.

1. On the Rules page, click Test in the upper-right corner. In the Test dialog box, specify the time to test rules, and click Test.

Parameter	Description
Partition	The partition expression for which rules are run. The actual partition key varies with the data timestamp. For a non-partitioned table, use NOPARTITIONTABLE as the partition expression.
Data Timestamp	The time to test rules. The default value is the current time.

2. Click The test run is complete. Click to view the result. to view the test results on the Search by Node page.

#### Manage subscriptions

By default, Data Quality sends notifications to the user who created a partition expression. You can add other subscribers so that Data Quality sends notifications to these users. Data Quality supports four notification methods: Email, Email and SMS, DingTalk Chatbot, and DingTalk Chatbot @ALL.

#### Change the owner

As the owner of a partition expression, you can change the owner to another member in the same workspace before you resign or are transferred. The default owner is the user who created the partition expression.

Move the pointer over the owner of a partition expression, and wait until a button appears. Click the button, enter the name of the new owner, and then click OK.

#### More actions

Action	Description
View Operation Log	Click it to view the records of all rules configured for the current partition expression.
View Check Results	Click it to view the last check result and the check result history of rules for the current partition expression on the Search by Node page.
Clone Rules	Click it to duplicate configured rules and subscribers to another partition expression.

### 2.9.3.2 DataHub monitoring

The Rules page is the most important part of Data Quality, where you can configure monitoring rules. Currently, Data Quality allows you to monitor MaxCompute tables and DataHub topics. This topic describes how to configure rules for DataHub monitoring.

DataHub monitoring supports the following features:

- Templates for monitoring stream discontinuity and data latency
- Stream processing features, such as custom Flink SQL, dimension table join, multi-stream join, and analytic functions

## Add a connection


Before configuring monitoring rules, you need to go to the Data Integration page to add a connection. For more information, see [Add a DataHub connection](#). After the connection is added, you can go to the Data Quality page to configure monitoring rules.

## Select a connection

1. In the left-side navigation pane, click Rules.
2. On the Rules page, select DataHub from the drop-down list in the upper-left corner. All topics of the DataHub connection are listed on the right.

You can also search for topics in the search box.

Configuration item	Description
Configure Flink Resources	After you add a connection, click Configure Flink Resources to configure Flink and Log Service resources related to the connection.
Topics	<p>The Topics tab lists all topics of the DataHub connection. You can click the following actions in the Actions column for a topic:</p> <ul style="list-style-type: none"><li>• <b>View Monitoring Rules:</b> Click it to create rules for the topic. You can create template rules and custom rules as needed.</li><li>• <b>Manage Subscriptions:</b> Click it to view and modify subscribers to the current topic, and change the notification method. You can configure DingTalk Chatbot notification methods. The changed notification method takes effect for all subscribers of the topic.</li></ul>

Configuration item	Description
Dimension Tables	<p>When you create custom rules for a topic, you can create dimension tables and use the JOIN clause to join dimension tables. If the collected data streams lack some fields for a dimension table, you need to supplement fields to data streams before data analysis and declare the dimension table in Data Quality.</p> <p>DataHub supports the dimension tables of ApsaraDB for HBase, Lindorm, ApsaraDB for RDS, Table Store, Taobao Distributed Data Layer (TDDL), and MaxCompute.</p> <p>Flink SQL does not design data definition language (DDL) syntax for dimension tables. You can use the standard CREATE TABLE statement. However, you need to add <code>period for system_time</code> to specify the period of a dimension table and declare that the dimension table stores time-varying data.</p> <div> <b>Note:</b> When you declare a dimension table, you must specify the unique key. When you use the JOIN clause to join dimension tables, the ON clause must contain the equivalent conditions of all unique keys.</div>

3. On the Topics tab, click View Monitoring Rules in the Actions column for a topic.



Configure monitoring rules

1. On the Monitoring Rules page, click Create Rule in the upper-right corner. Data Quality allows you to create template rules and custom rules.

- Click Create Template Rule. Two templates are available: Data Delay and Stream Discontinuity.

For example, you can select Data Delay for Template Type.

Parameter	Description
Rule Name	The name of the rule. The name can be up to 255 characters in length.

Parameter	Description
Field Type	The fields to be monitored. By default, the field type is All Fields in Table.
Template Type	<p>The template to apply to the rule. Valid values: Data Delay and Stream Discontinuity.</p> <ul style="list-style-type: none"> <li>- <b>Data Delay:</b> monitors the interval between the time when data is generated and the time when data is written to DataHub based on the data timestamp field. If the interval exceeds a specified threshold, an alert is generated.</li> </ul> <div>  <b>Note:</b> The data timestamp field supports two data types: Timestamp and String (YYYY-MM-DD hh:mm:ss).         </div> <ul style="list-style-type: none"> <li>- <b>Stream Discontinuity:</b> monitors the period during which no data is written to DataHub. If the period exceeds a specified threshold, an alert is generated.</li> </ul> <div>  <b>Note:</b> Before configuring a stream discontinuity rule, you need to activate Alibaba Cloud Realtime Compute and create a project.         </div>
Alerts Threshold	The maximum number of alerts generated for data latency. Data Quality reports an alert when the number of alerts generated for data latency exceeds this threshold. This parameter takes effect only when you select Data Delay for Template Type.
Data Timestamp Field	The data timestamp field of the topic for which the rule is created. This field supports two data types: Timestamp and String (YYYY-MM-DD hh:mm:ss). This parameter takes effect only when you select Data Delay for Template Type.
Alert Frequency	The interval for reporting an alert. You can set the alert interval to 10 minutes, 30 minutes, 1 hour, or 2 hours.
Warning Threshold	The warning threshold, in seconds. The value must be an integer and less than the error threshold.

Parameter	Description
Error Threshold	The error threshold, in seconds. The value must be an integer and greater than the warning threshold.

- If template rules do not meet your requirements for monitoring the data quality of DataHub topics, you can click **Create Custom Rule** to create a rule as required.

**Note:**

- The field in the **SELECT** clause must be a column. Ensure that you can compare the field values with the warning and error thresholds.
- The **FROM** clause must include the current topic and all its columns.

Parameter	Description
Rule Name	The name of the rule. The name must be unique in the topic and can be up to 20 characters in length.
Script	<p>The custom SQL script, which is used to set a rule. The return value of the <b>SELECT</b> clause must be unique.</p> <ul style="list-style-type: none"><li>- <b>Example 1: Use a simple SQL statement</b> <pre>select id as a from zmr_tst02;</pre></li><li>- <b>Example 2: Join the topic and a dimension table named test_dim</b> <pre>select e.id as eid from zmr_test02 as e join test_dim for system_time as of proctime() as w on e.id=w.id</pre></li><li>- <b>Example 3: Join the topic and another topic named dp1test_zmr01</b> <pre>select count(newtab.biz_date) as aa from (select o.* from zmr_test02 as o join dp1test_zmr01 as p on o.id=p.id)newtab group by id.biz_date,biz_date_str, total_price,'timestamp'</pre></li></ul>
Warning Threshold	The warning threshold, in minutes. The value must be an integer and less than the error threshold.



Parameter	Description
Error Threshold	The error threshold, in minutes. The value must be an integer and greater than the warning threshold.
Minimum Alert Interval	The minimum interval for reporting an alert, in minutes.
Description	The description of the rule.

2. After the configuration is completed, click **Save** to apply the rule to the topic.

More actions

- **View Log:** Click it to view the runtime logs of rules.
- **Enable Monitoring:** Click it to enable monitoring rules.
- **Manage Subscriptions:** Click it to view and modify subscribers to the current topic, and change the notification method. The changed notification method takes effect for all subscribers of the topic.

You can configure DingTalk Chatbot and DingTalk Chatbot @ALL notification methods to send notifications to DingTalk groups.

If you select DingTalk Chatbot, you need to add a DingTalk Chatbot to a DingTalk group, and then copy the webhook URL of the DingTalk Chatbot to the Manage Subscriptions dialog box. In this way, you can use the DingTalk Chatbot to send notifications for the topic.

## 2.10 Realtime Analysis

### 2.10.1 Apply for joining a workspace

Developers can send requests for joining a workspace. The requests need to be approved by the workspace owner.

You can use the Real-Time Analysis service only if you are a member of a workspace and is an analyst.

The workspace owner can view the requests on the Project Management page and add the requester to the workspace as a member.

## 2.10.2 Apply for data access permissions

**Before performing data analytics, you must obtain access permissions to the required tables.**

Apply for permissions

1. **Log on to the [DataWorks](#) console as a developer.**
2. **Choose Data Management > Table Management.**
3. **Locate the required table, and click Apply for Permissions.**

Grant permissions

**To grant permissions after an application for permissions is submitted, an administrator chooses Data Management > Table Permissions > For My Approval.**

## 2.10.3 Ad hoc query

**This section describes how to query data.**

**In the top navigation bar, click Real-Time Analysis, and right-click the root directory name to create, rename, or delete folders or files.**

1. **Right-click the root directory and select Create File.**

**In this step, create an ad hoc query file. Currently, only MaxCompute SQL is supported.**

2. **Enter a query SQL statement.**
3. **Click Run to obtain the result.**

**If an error occurs, you can view running logs to locate the error.**



### Note:

**If a syntax error occurs, you can view the number of row where the error locates. However, the row number in the logs may not be the same as that on the page where the code runs. This is because the row in the logs indicates the row when the code is submitted to MaxCompute.**

## 2.10.4 Personal tables

**You can save query results to personal tables, and query data in personal tables.**

**You can create personal tables by using either of the following methods:**

- **Run a CREATE TABLE statement in a node created on the Query tab.**

- Save query results to a personal table.

## 2.11 Data Service

### 2.11.1 Overview

With Data Service, you can manage all your table APIs after you create new APIs or register existing APIs. You can also easily publish your APIs to API Gateway. Together with API Gateway, Data Service provides a secure, stable, low-cost, and easy-to-use data sharing service.

Data Service adopts a serverless architecture and allows you to develop table APIs without thinking about infrastructure such as compute resources. Data Service supports automatic scaling for compute resources, which significantly reduces your OPEX.

#### Create an API

In Data Service, you can quickly create APIs based on tables in relational databases or NoSQL databases using a visual wizard. It takes only a few minutes to configure a data API, and coding is not required. You can also create APIs by specifying SQL scripts. The script mode supports advanced functions such as associative tables, complex criteria, and aggregate functions.

#### Register an API

You can register existing RESTful APIs to Data Service for unified API management. Four request methods and three data formats are supported. The four request methods are GET, POST, PUT, and DELETE. The three data formats are tables, JSON, and XML.

#### API Gateway

API Gateway provides API lifecycle management services, including API publishing, management, maintenance, and monetization. It enables low-risk, simple, cost-effective, and fast microservice integration, front and back end separation, and system integration. You can use API Gateway to share functions and data with your partners and third-party developers.

API Gateway supports authorization, authentication, flow control, and billing for Data Service.

## 2.11.2 Terms

This section introduces terms of Data Service.

Name	Description
Data source	Indicates database links. Data Service accesses data through data sources. Data sources are configured in Data Integration .
Create an API	Creates APIs based on data tables.
Register an API	Registers existing APIs to Data Service for unified management.
Wizard mode	Guides you through the procedure of API creation. This method is suitable for beginners who want to create simple APIs. You do not need to write any code.
Script mode	Allows you to create APIs by writing SQL scripts. This method supports associative tables, complex queries, and aggregate functions. This method is suitable for experienced developers who want to create complex APIs.
API group	Indicates a set of APIs for a specific scenario or for consuming a specific service. An API group is the smallest group unit in Data Service, and the smallest unit for API Gateway management. API groups are published in Alibaba Cloud API Marketplace as API products.
API Gateway	Indicates a hosted service provided by Alibaba Cloud to manage APIs. API Gateway supports API lifecycle management, permission management, access management, and traffic control.

## 2.11.3 Create an API

In Data Service, you can quickly create APIs based on tables in relational databases or NoSQL databases using a visual wizard. It takes only a few minutes to configure a data API, and coding is not required.

You can also create APIs by specifying SQL scripts. The script mode supports advanced functions such as associative tables, complex criteria, and aggregate functions.

The differences between the wizard mode and script mode are described as follows:

Table 2-1: Differences between the wizard mode and script mode

Category	Description	Wizard mode	Script mode
Query object	Queries a single table from one data source	Supported	Supported
	Queries associative tables from one data source	Not supported	Supported
Search condition	Searches for an exact number	Supported	Supported
	Searches for a range of numbers	Not supported	Supported
	Matches an exact string	Supported	Supported
	Performs fuzzy search for strings	Supported	Supported
	Sets required and optional parameters	Supported	Supported
Query result	Returns the field value	Supported	Supported
	Performs a mathematical calculation for field values	Not supported	Supported
	Performs an aggregate operation on field values	Not supported	Supported
	Displays results with pagination	Supported	Supported

### 2.11.3.1 Configure data sources

Before generating an API, make sure that you have configured the relevant data sources. Data Service allows you to obtain table schemas and query data through APIs from data sources.

If you need to configure data sources, navigate as follows: DataWorks > Data Integration > Data Sources. The following table lists whether each data source type supports API generation in wizard or script mode.

Data source type	Generate API in wizard mode	Generate API in script mode
RDS	Supported	Supported
MySQL	Supported	Supported
PostgreSQL	Supported	Supported
Oracle	Supported	Supported
Table Store (OTS)	Supported	Not supported

### 2.11.3.2 Generate APIs in wizard mode

This section describes the procedure and precautions for generating APIs in wizard mode.

You only need to follow simple steps to generate APIs in wizard mode without writing any code. If you do not require advanced API features or have limited experience in code development, we recommend that you generate APIs in wizard mode.



**Note:**

Before configuring APIs, navigate to DataWorks > Data Integration to configure the data source information.

Configure basic API information

Click **Generate API** on the **API List** tab page and select **Wizard Mode**. Enter the basic API information.

You must pay special attention to the configuration of API groups. An API group is a collection of APIs designed for a specific feature or scenario. It is also the smallest management unit in API Gateway.

**Note:**

Assume that you need to design an API product for checking the weather. The product consists of APIs for checking weather by city name, by scenic spot name, and by zip code. Then, you can create an API group named Weather Checking and put the three types of APIs into this group. Then, this API group can be sold as a weather checking API product on the API market. If you generate APIs for your own application, you can use the API group for classification.

Currently, protocol, request method, and response type can only be set to HTTP, GET, and JSON.

After entering the basic API information, click Next to go to the API Parameters page.

Configure API parameters

1. Choose **Connection Type > Connection Name > Table** to select the table to be configured.

**Note:**

Before configuring the API parameters, you must configure the data source in Data Integration. In the Select Table drop-down list, you can search for a table by name.

2. Specify request and response parameters. After you select a table, all fields of the table are automatically listed on the left. Select the fields to be used as request parameters and response parameters, and then add them to the corresponding parameter list.
3. Edit parameter information. Click **Change** in the upper-right corner of the request and response parameter lists to edit the parameters. You can specify the parameter name, example value, default value, and description. You can also specify whether the parameter is required and can be fuzzy matched. Only parameters of string type can be fuzzy matched. The Required and Description fields must be specified.

Pay attention to the page settings for response. If Multiple Pages is disabled, the API returns a maximum of 500 results by default. If the API may return more than 500 results, enable Multiple Pages. The following common parameters are available when the Multiple Pages feature is enabled.

- **Common request parameters**
  - **pageNum:** The current page number.
  - **pageSize:** The number of records per page.
- **Common response parameters**
  - **pageNum:** The current page number.
  - **pageSize:** The number of records per page.
  - **totalNum:** The total number of records.

**Note:**

- **Only exact match is supported for table queries through APIs. Field values are returned in the response as they are in the table.**
- **To enhance the matching efficiency, set an indexed field as a request parameter.**
- **If you do not specify any request parameters for an API, the Multiple Pages feature must be enabled.**
- **To make it easier for API callers to understand the details of an API, we recommend that you specify the sample value, default value, description, and other parameters of the API.**
- **You can click APIs Created to view a list of APIs that have been generated for the current table. Avoid generating an API that already exists in the system.**

When the configuration of the API parameters is complete, click Next to enter the API testing section.

#### Test the API

After completing the configuration of API parameters, you can start the API test.

Set the parameters and click Test to send an API request. The request and response details are displayed on the right. If the test fails, check carefully the error message, make modifications accordingly, and test the API again.

In particular, pay attention to the configuration of the successful response example. After the API is configured, the system automatically generates the exceptional response examples and error codes, but cannot automatically generate a successful response example. After the test succeeds, you need to click Save as Successful Response Example to save the current test result as the successful response



example. If the response contains sensitive data that must be masked, you can manually edit the response.

**Note:**

- The successful response example is an important reference for API callers, and therefore must be configured.
- The Call Latency value is the latency of the current API request, which is used to evaluate the API performance. If the latency is long, consider optimizing the database.

After the API test is completed, click **Complete**. The API is generated.

View API details

Return to the API List tab page. Click **Details** for an API in the **Actions** column to view the details of the API. The API details page provides the detailed information about the API from the perspective of callers.

### 2.11.3.3 Create an API by specifying scripts

This section describes how to create an API by specifying SQL scripts.

You can create APIs by specifying SQL scripts to meet customized needs. The script mode supports advanced functions such as associative tables, complex criteria, and aggregate functions.

Configure the API basic information

Click **Create API** in the list of API services, select **Script Mode**, and enter the API basic information.

You must pay special attention to configuration of the API group. An API group includes a collection of APIs that are used for a specific scenario. It is the minimum management unit in API Gateway.

**Note:**

For example, you want to configure an API product for weather query. This weather API product consists of three APIs: API for weather query by city name, API for weather query by resort, and API for weather query by postal code. In this case, you can create an API group for weather query, and assign these three APIs to this group. After you publishing this group to Marketplace, customers can

**purchase it as a weather query product. However, if you are creating APIs for your own app, you can use create groups to classify groups.**

**Currently, the APIs only support HTTP, the GET method, and the JSON return type.**

**After entering the basic information, click Next to specify API parameters.**

Configure the SQL query statements and parameters

- 1. Choose Data Source Type > Data Source Name > Table, and select a data source and table. Click the table name to view its field information.**



**Note:**

- You must select a data source. You can only query associative tables from one data source.**
- You must configure the data source in the Data Integration service before performing this step.**

- 2. Write a SQL query statement for the API.**

**You can enter SQL code in the area on the right side. The system supports adding a SQL statement with one-click. You only need to select fields from the field list and click Add SQL Statement. The system then creates a `SELECT xxx FROM xxx` statement and inserts the statement at the pointer. This feature improves the efficiency of writing a SQL statement when a large number of fields need to be added. The `SELECT` clause specifies the fields that the API response outputs. The `WHERE` clause specifies the API request parameters. You must use `${}` to interpolate a request parameter.**

- 3. Specify API parameters.**

**Click Parameters in the upper-right corner. On the page that appears, you can specify the parameter type, sample value, default value, and description. You must specify the Type and Description parameters. To allow API callers to learn more about this API, we recommend that you provide complete parameter information if possible.**

**During configuration, pay attention to the response paging settings. If you do not enable the response pagination feature, the API returns up to 500 records by default. To return more records, enable the response pagination feature. When the**

pagination feature is enabled, the following common parameters are automatically added:

- Common request parameters
  - pageNum: the current page number.
  - pageSize: the page size, indicating the number of records per page.
- Common response parameters
  - pageNum: the current page number.
  - pageSize: the page size, indicating the number of records per page.
  - totalNum: the total number of records.



**Note:**

Notes about SQL rules are described as follows:

- Only one SQL statement is supported.
- Only the SELECT statement is supported. Other statements such as INSERT, UPDATE, and DELETE are not supported.
- The SELECT statement specifies the fields that the API response outputs. The variable param in \${param} of the WHERE clause specifies an API request parameter.
- SELECT \* is not supported. You must specify the columns to be queried.
- Single table queries, associative table queries, and nested queries within one data source are supported.
- If the name of the column that the SELECT statement specifies has a table prefix, such as t.name, you must create an alias for the corresponding response parameter, for example, t.name as name.
- If you use an aggregate function, such as min, max, sum, and count, the alias must be used as the response parameter name, such as sum (num) as total \\_ num.
- In SQL, \${param} specifies a request parameter. The \${param} in the string also specifies a request parameter. If an escape character \ is placed before \${param}, the \${param} is processed as an ordinary string.
- You cannot quote a \${param} in the format of '\${id}' 、 'abc\${xyz}123'. You must quote a \${param} in the format of concat('abc', \${xyz}, '123').

## Test an API

After specifying SQL query statements and API parameters, you can test the API.

Specify the parameters and click **Test** to send API requests. You can view request details and responses on the right side of the page. If the API fails the test, read the error message carefully, make appropriate adjustments, and test your API again.

In particular, pay attention to the setting of the normal response sample. When testing an API, the system automatically generates exception examples and error codes. However, normal response examples are not automatically generated.

After the API passes the test, click **Save as Normal Response Sample** to save the current test result as the normal response sample. If sensitive data is included in the response, you can manually edit it.



### Note:

- Normal response examples provide an important reference for the API callers. Specify an example if possible.
- The API calling delay is the delay of the current API request, which is used to evaluate the API performance. If the delay is excessively long, we recommend that you optimize the database.

After completing the API test, click **Finish**. The API is successfully created.

## View API details

You can return to the API list and click **Details** to view API details. This page displays the detailed information about an API from the view of a caller.

### 2.11.4 Register an API

This section shows how to register existing APIs, and manage all your table APIs after you create new APIs or register existing APIs. You can also easily publish your APIs to API Gateway.

Currently, you can only register RESTful APIs. Four request methods and three data formats are supported. The four request methods are GET, POST, PUT, and DELETE. The three data formats are tables, JSON, and XML.

## Configure the API basic information

1. Click **Register API** in the list of API services.



## 2. Configure the basic API information.

Field	Description
API Name	The name must be 4 to 50 characters in length. It must start with a Chinese character or an English letter, and can contain Chinese characters, English letters, digits, and underscores (_).
API Group	An API group includes a collection of APIs that are used for a specific scenario. It is the minimum management unit in API Gateway. In Alibaba Cloud API Marketplace, each API group corresponds to a specific API product. To create an API group, click Create API Group.
Protocol	Currently, only HTTP is supported.
Backend Service Host	Enter the host of the API to be registered. The host must start with http:// or https://, and cannot contain the path.
Backend Service Path	Enter the path of the API. Wrap parameter names in brackets ([ ]), for example, /user/[userid]. In the next step, parameters defined in Backend Service Path are automatically added to the request parameter list.
API Path	API Path is the alias of Backend Service Path. APIs with different API paths can share the same backend service path and backend service host. If a parameter is defined in the background service path, it must also be defined in the API path, which is wrapped in a bracket ([ ]).
Request	You can select GET, POST, PUT, or DELETE. The parameters to be configured vary with the request mode.
Return Type	Currently, JSON and XML return types are supported.
Description	You can briefly describe the API.

## 3. After entering the basic information, click Next to specify API parameters.

Configure API parameters

After entering the basic API information, you can specify the API parameters, including the request parameters, response example, and error code.

Configuration item	Description
Request parameters	<ul style="list-style-type: none"><li>• <b>Parameter Location:</b> The available request parameter locations (Path, Header, Query, or Body) vary with the request mode.</li><li>• <b>Constant Parameters:</b> A constant parameter is fixed and is invisible to the caller. You do not have to specify the constant parameters when calling an API . Instead, the constant parameters are automatically sent to the backend service. This is useful when you want to set a parameter to a fixed value and hide the parameter value from the caller.</li><li>• <b>Request Body:</b> Request Body is required only when the request method is POST or PUT. You can enter the body description in the request body definition . It is equivalent to an example of the request body, and API callers can refer to the format. The content types of the request body support JSON and XML.</li></ul> <div> <b>Note:</b> If a parameter has been defined in both the request body and the request parameter list, the parameter value in the request body takes priority.</div>
Sample responses	You can enter a normal example or an exception example for API callers to refer to when writing the return parse code.
Error code	Enter the common errors and solutions in API calling. This helps API callers to troubleshoot these errors. <div> <b>Note:</b> To ensure that the API is easily used by the callers, provide complete API parameter information if possible, especially the parameter sample values, default values, and response examples.</div>

### Test an API

After specifying API parameters, you can test the API.

Specify the parameters and click Test to send API requests. You can view request details and responses on the right side of the page. If the API fails the test, read the error message carefully, make appropriate adjustments, and test your API again.

In particular, pay attention to the setting of the normal response sample. When testing an API, the system automatically generates exception examples and error codes. However, normal response examples are not automatically generated. After the API passes the test, click **Save as Normal Response Sample** to save the current test result as the normal response sample. If sensitive data is included in the response, you can manually edit it.

**Note:**

- Normal response examples provide an important reference for the API callers. Specify an example if possible.
- The API calling delay is the delay of the current API request, which is used to evaluate the API performance. If the delay is excessively long, we recommend that you optimize the database.

After completing the API test, click **Finish**. The API is successfully created.

## 2.11.5 Test APIs

During API generation and registration processes, you can test the APIs. In addition, Data Service provides a separate API testing feature.

In the Data Service service, choose **More > Test** in the **Actions** column for an API on the API list. Alternatively, you can click **API Test** in the left-side navigation pane to go to the API Test page and select an API for testing.

**Note:**

- The API Test page provides only online API testing. You cannot update or save the successful response example for an API on this page. To update the successful response example for an API, click **Change** in the **Actions** column for this API on the API list to enter the **Generate API** page. Update the successful response example for this API in the API testing step.
- For more information about how to test an API, see [Generate APIs in wizard mode](#).

## 2.11.6 Delete APIs

On the API List page, find the target API and choose **More > Delete** in the **Actions** column to delete this API.

**Note:**

- An API can be deleted only when it is in Offline status. If it has been published, take the API offline before deleting it.
- The delete operation is irreversible. Use caution when deleting an API.

## 2.11.7 Publish APIs

API Gateway provides API hosting service. As a service bus, it enables you to publish, manage, maintain, and sell APIs in their lifecycle. It helps you easily and quickly aggregate microservices, separate frontend from backend system, and integrate systems at low costs and low risks, making features and data available to partners and developers.

API Gateway provides permission management, traffic control, access control, and metering services. The service makes it easy for you to create, monitor, and secure APIs. Therefore, we recommend that you publish the APIs that have been created and registered in Data Service to API Gateway. Data Service and API Gateway are interconnected, which allows you to publish APIs to API Gateway easily.

### Publish APIs to API Gateway

To publish an API, you must first activate the API Gateway service.

After activating the API Gateway, you can click **Publish** in the **Actions** column for an API in the API list to submit the API to the API Gateway. The system automatically registers the API with API Gateway during the publish process. The system also creates a group in API Gateway with the same name as the API group that the API belongs in Data Service and publishes the API in this group. After the API is published, you can access the API Gateway console to view API details or configure bandwidth throttling, access control, and other features.

If you generate an API to be called by your own application, you need to create an application in API Gateway, authorize the application to use the API, and enable the application to call the API by using AppKey and AppSecret. For more information, see *API Gateway Document* . API Gateway also provides SDKs for mainstream programming languages to help you quickly integrate the API with your own application.



## 2.11.8 Call APIs

This section describes how to call an API after this API is published on API Gateway.

API Gateway provides API authorization and the SDK for calling APIs. You can authorize yourself, your associates, or third parties to use APIs. If you want to call an API, perform the following operations.

Three conditions for calling an API

The following three conditions must be met to call an API:

- **API:** The API that you call is clearly defined by API parameters.
- **App:** The app that you use to call the API has a key pair that uniquely identifies you. The key pair consists of the AppKey and AppSecret.
- **Relationship between the API and app:** If you want to call an API by using an app, the app must have the permission to call this API. This permission is granted through authorization.

Procedure

### 1. Get the API documentation.

The method of getting the API documentation varies depending on how you obtain the API. You can obtain the API by purchasing it from the marketplace, or you are authorized to use the API for free.

### 2. Create an app.

The app identifies you when you call the API. Each app has a key pair: AppKey and AppSecret, which are equivalent to an account and password.

### 3. Get the permission to call the API.

Authorization means granting the app the permission to call the API. Your app must be authorized to call the API. The authorization method varies depending on how you obtain the API.

### 4. Call the API.

You can edit an HTTP or HTTPS request to call the API. Before calling the API, you can use examples of calling APIs in multiple languages on the API Gateway console to test the calling.



**Note:**

For more information about API Gateway, see *API Gateway document*.

## 2.11.9 FAQ

- **Q: Do I need to activate the API Gateway service?**

**A:** API Gateway provides you with high-performance and highly available API hosting services. If you need to make your APIs available to others, activate the API Gateway service first.

- **Q: Where can I add and change data sources?**

**A:** Navigate as follows: DataWorks > Data Integration > Data Source. Data Service can automatically read data from the data sources that you have configured.

- **Q: What is the difference between the wizard mode and the script mode?**

**A:** The script mode provides more powerful features. For more information, see [Create an API](#).

- **Q: What are API groups in Data Service? Are they the same as those in API Gateway?**

**A:** An API group is a set of APIs specific to a feature or scenario. An API group is the smallest organization unit in Data Service, which is similar to an API group in API Gateway. After you publish an API from Data Service to API Gateway, API Gateway automatically creates an API group with the same name.

- **Q: How can I configure an API group appropriately?**

**A:** An API group includes APIs that provide a specific feature or solution. For example, you can put the API for querying weather by city name and the API for querying weather by coordinates into the same API group named "Query Weather".

- **Q: How many API groups can I create?**

**A:** You can create a maximum of 100 API groups by using the same Alibaba Cloud tenant account.

- **Q: When do I need to enable the return results to be displayed in multiple pages?**

**A:** By default, an API call returns a maximum of 500 records. If there are more than 500 records, enable the Multiple Pages feature. If you do not specify any request parameters, the API call usually returns a large number of records and the Multiple Pages feature is automatically enabled.

- **Q: Do APIs generated by Data Service support POST requests?**

**A:** Currently, APIs generated by Data Service only support GET requests.

- **Q: Does Data Service support HTTPS?**

**A: Currently, Data Service does not support HTTPS. HTTPS may be supported in later versions.**

## 2.12 Data Protection

### 2.12.1 Overview

Data Protection is a data security management platform. It can be used to detect data assets, detect sensitive data, classify data, de-identify data, monitor data access behavior, report alerts, and audit risks.

Data Protection provides security management services for MaxCompute.

Data Protection provides the following features:

- **Intelligent sensitive data detection**

Data Protection automatically detects an enterprise's sensitive data based on self-training models and algorithms, and clearly displays statistics on data types, volume, and visitors. It also recognizes custom data types.

- **Accurate data classification:** Data Protection allows you to classify data and create custom levels for better data management.
- **Flexible data de-identification**

Data Protection provides diverse and configurable methods for dynamic data de-identification.

- **Risky behavior monitoring and auditing**

Data Protection uses various correlation analysis algorithms to detect risky behavior. It also provides alerts and supports visualized auditing for detected risks.

### 2.12.2 Services

This topic describes the commonly used services provided by Data Protection.

Service	Description
Configure rules for defining sensitive data	You can configure rules to define sensitive data based on the security regulations and requirements of your organization. Data Protection can detect the sensitive data you have defined .

Service	Description
View the distribution of data	With an authorized account, Data Protection automatically detects sensitive data in the tenant's MaxCompute projects based on the data detection rules that are defined. Then, you can view the distribution of detected data on the next day.
View the information about data activities	Data Protection provisions manipulate, query, and export activities related to sensitive data from different perspectives .
View the information about data export	Data Protection monitors activities with sensitive data exported from MaxCompute based on the risk detection rules you have configured.
Manage the data security levels	You can specify data security levels based on security regulations and requirements of your organization.
Manage data that is incorrectly detected	You can manually correct detected data types and remove or recover sensitive data.

### 2.12.3 Access Data Protection

This topic describes how to access Data Protection.

1. Log on to the DataWorks console.
2. Move the pointer over the DataWorks icon in the upper-left corner and then choose **All Products > Data Protection**.

Data Protection provides security management services for MaxCompute.

Toolbar	Description
Top navigation bar	The services that the current user has permissions to access, including DataStudio, Data Management, Operation Center, Organization Management, Project Management, Real-Time Analysis, DataService Studio, Machine Learning Platform For AI, and Data Protection.
User information	The personal information that can be viewed and edited by the current user, including the email address, mobile number, and AccessKey.
Left-side navigation pane	The left-side navigation pane for the services that can be navigated to from the top navigation bar. Items in the navigation pane vary depending on the specific service.
Home page of Data Protection	The brief description of core features.

## 2.12.4 Configure rules for defining sensitive data

This topic describes how to configure rules for defining sensitive data.

### Procedure

1. Log on to the DataWorks console.
2. Move the pointer over the DataWorks icon in the upper-left corner and then choose **All Products > Data Protection**.
3. In the left-side navigation page, choose **Data Definition > Management**. On the Data Definition page that appears, click **Create Rule**.
4. Specify required parameters in the Create Rule dialog box and click **OK**.

Parameter	Description
Rule Name	The name of the rule. It can contain digits, letters, and underscores (_).
Rule Type	The type of the rule. The system provides eight built-in templates for detecting the ID card number, bank card number, email address, mobile number, IP address, MAC address, telephone number, and license plate number respectively. The system also supports custom rules.
Owner	The user who configures the rule.
Description	The additional information about the rule.

5. Click **Change** in the **Actions** column of a rule for advanced configuration.

Parameter	Description
Level	The security level of the data to which the rule applies. If the existing levels do not meet the requirements, modify the rules on the Levels page.
Method	The method for de-identifying data. Do not specify this parameter if data de-identification is not required.
Content Scanning	Specifies whether to enable content scanning. This option is selected by default for all the eight built-in data detection templates. If you have selected a template, you cannot modify the detection rule, but you can verify the accuracy of the rule. If you select regular expression matching, you can customize the detection rule.

Parameter	Description
Field Scanning	Specifies whether to enable field scanning. This approach provides two matching methods: exact matching and fuzzy matching of field names. Multiple-field matching is supported, and the relationship between the fields is OR.

6. After the rule configuration is completed, click Save.

7. The rule does not take effect immediately after you save the configuration. Check the rule and change its status to effective after confirming the rule is correct.



**Note:**

Follow these rules when configuring a data detection rule:

- The rule name must be unique.
- The content scanning and field scanning configuration must be unique.
- You can only view the sensitive data that is detected based on the data detection rule one day after the rule takes effect.

### 2.12.5 View the distribution of data

On the next day after you configure and activate sensitive data detection rules as a data security administrator, you can access Data Recognition Rules to view the overall data distribution, hierarchical data distribution, and field details.

You can filter the data distribution statistics by project, rule name, rule type, and risk level based on your query requirements.

### 2.12.6 View the information about data activities

On the next day after you configure and activate sensitive data detection rules as a data security administrator, you can access Data Activities to view related activity statistics, trend, and details.

You can filter the data activity statistics by project, user, rule name, rule type, and risk level based on your query requirements.

### 2.12.7 View the information about data export




On the next day after you configure and activate sensitive data detection rules as a data security administrator, you can access Data Export to view the information

about data export from MaxCompute. You can view the overall export status, top N users with the largest amount of export data, and export details.

You can filter the data export statistics by rule name, rule type, and greater than condition based on your query requirements.

## 2.12.8 Manage the data security levels

When creating a rule, you need to specify a security level for the data to which the rule applies. You can create and delete security levels on the Levels tab. You can also modify the priority of each security level and manage rules by security level.

Operation	Description
Create a security level	Click Create Level. Specify the security level name and operator.
Manage rules by security level	Click  in the Actions column.
Modify the priority of a security level	Drag and drop  in the Actions column.
Delete a security level	Click  in the Actions column and then click Delete in the dialog box that appears.

## 2.12.9 Manage data that is incorrectly detected

On the Manual Check page, you can manually correct the sensitive data that is incorrectly detected by rules. For example, you can delete incorrectly detected data, change the type of the detected data, and delete or recover data in batches.

Operation	Description
Delete incorrectly detected data	Turn off the switch in the Actions column to change the status to Deleted. The deleted data can be recovered.
Change the type of the detected data	If a license plate number is recognized as an email address, click the Edit icon next to Email and change the data type. You can only select a rule that has been configured.
Delete or recover data in batches	You can delete or recover data in batches by selecting the check boxes of the target records and clicking the corresponding button.

## 2.12.10 Customize de-identification rules

This topic describes how to customize de-identification rules in Data Protection so that DataWorks can dynamically de-identify the results of queries.

Customize de-identification rules in Data Protection



**Note:**

You must first activate Data Protection to use this feature.

1. Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and then choose **All Products > Data Protection**. The Data Protection page appears.
2. In the left-side navigation pane, choose **Management > Data Masking**.
3. Set Masking Scene to **Default (\_default\_scene\_code)** and then click **Create Rule** in the upper-right corner.
4. In the **Create Rule** dialog box that appears, set **Rule**, **Owner**, and **Method**.



**Note:**

You must first create sensitive data detection rules and activate them on the Data Recognition page. Then, you can use the rules when configuring data de-identification rules.

Currently, Data Protection provides two methods for data de-identification, including Hashing and Masking Out.

- **Hashing**

If you select this method, you need to specify a security domain. Different hash values are generated for the same data record based on the same rule configured with different security domains.

- **Masking Out**

This method uses asterisks (\*) to mask specified parts of a data record. It is commonly used.

Parameter	Description
Recommended	You can select recommended policies to mask data of common types such as ID card numbers and bank card numbers.



Parameter	Description
Custom	You can flexibly specify whether to mask the specified number of characters at the first, middle, or last part of a data record.

5. After the configuration is completed, click OK. The Data Masking page appears.
6. On the Data Masking tab, change the status of a rule to Active or Inactive as needed.

After the configuration is completed, click  in the Actions column of the rule to test whether it works.

7. Click the Whitelist tab. On the tab that appears, click Add Account in the upper-right corner.
8. In the Add Account dialog box that appears, set Rule, Account, and Effective From and click Save.

**Note:**

If you query data beyond the time range specified for the whitelist, the query results will still be de-identified.

Verify the de-identification effect in DataWorks

After you successfully create and configure de-identification rules, DataWorks dynamically de-identifies the results of queries in your workspace based on the rules.

**Note:**

You must first turn on the Mask Data in Page Query Results switch for your workspace in the DataWorks console. For more information, see [Enable data desensitization for DataWorks workspaces](#).

## 2.13 Data Asset Management

### 2.13.1 Overview

Data Management provides you with an overview of your data assets. Data Management requires that data be synchronized by using Data Integration and

processed by using DataStudio before you manage your tables and APIs stored in your business system and DataWorks.

### 2.13.2 Asset administrator (View data asset information)

This section shows how asset administrators view data asset information.

Procedure

1. Log on to the DataWorks console and click Data Management.



**Note:**

You can also log on to the DataWorks console, and change the web address of another service of DataWorks, such as DataStudio, to go to the Data Management page. For example, change "ide" in the web address of DataStudio to "asset" to go to the Data Management page.

2. Configure keywords on the homepage.
3. In the top navigation bar, click Asset Category to configure the required information.

### 2.13.3 Asset user

Asset users can access the Data Assets service to perform operations such as searching for assets, applying for asset permissions, and using assets.

1. Navigate to the Data Assets service.



**Note:**

To jump to the Data Assets page, replace ide in the URL of DataStudio with asset.

2. Enter an asset keyword in the search box on the homepage of the Data Assets service.
3. Click Asset Category in the top navigation bar. On the page that appears, you can search assets.
4. Click a file name to view details.
5. Click Apply for Permission under the file name. Complete the settings in the dialog box that appears to apply for the file permission.
6. Choose Approvals > Submitted by Me to view the applications you have submitted.
7. After the permission application is approved, you can click Download under the file name to download this file.

## 2.13.4 Asset manager

Asset managers can manage assets and authorizations in the Data Assets service.

### Manage assets

On the Assets page, you can view file details, upload files, and edit files.

1. Log on to the DataWorks console, and jump to the Data Assets page.



**Note:**

To jump to the Data Assets page, replace `ide` in the URL of DataStudio with `asset`.

2. Click Assets in the top navigation bar. In the left-side navigation pane, click Files. On the Files tab, you can view the name, owner, size, status, and other information of each file.
3. Click Upload File in the upper-right corner to upload a file.
4. Click Edit in the Actions column to edit the corresponding file.

### Manage authorizations

1. Log on to the DataWorks console, and jump to the Data Assets service.
2. Click Permissions in the top navigation bar. In the left-side navigation pane, click Pending Approval.
3. View the applications that have not been handled. Click Agree or Reject to handle these applications.

## 2.13.5 Create a data asset category

This section shows how to create a data asset category.

1. Log on to the DataWorks console and click Data Management.



**Note:**

You can also log on to the DataWorks console, and change the web address of another service of DataWorks, such as DataStudio, to go to the Data Management page. For example, change `"ide"` in the web address of DataStudio to `"asset"` to go to the Data Management page.

2. In the top navigation bar, click Asset Management, and go to the Category Management page.
3. Click Create Category, and enter the required information.
4. Click the Table tab for the category.

5. Select the tables to be added to the category, and click OK.

## 2.13.6 Manage tables

Tables in all DataWorks services are synchronized to Data Assets.

View tables

1. Log on to the DataWorks console and jump to the Data Assets page.



**Note:**

To jump to the Data Assets page, replace `ide` in the URL of DataStudio with `asset`.

2. Click **Assets** in the top navigation bar and select **Tables** from the left-side navigation pane.

Edit basic table information

Click **Change** in the **Actions** column for a table. In the dialog box that appears, enter the basic information of the table.

The parameters are described as follows:

Parameter	Description
Display Name	(Required) The alias of the table.
Description	(Required) The description of the table.
Table Properties	<ul style="list-style-type: none"><li>• <b>Core Table:</b> The core data assets in the business system. If <b>Core Table</b> is selected, this table is published to the government directory. The configuration of the table properties requires special attention from asset administrators.</li><li>• <b>Temporary Table:</b> The system table or table irrelevant to business. This table does not require additional business information or special attention from asset administrators.</li></ul>
Label	The remarks made by the table administrator.

Edit fields

Click **Change** in the **Actions** column for a table. In the dialog box that appears, select the **Field** tab.

Specify the **Display Name** and **Description** for the fields. Select an option for **Share**, **Open**, and **Publish to Government Directory**. If you select **Publish** for **Publish to Government Directory**, this field is published to the government directory.

### Change table category

**Click Change Category in the Actions column for a table to change the category of the table.**

## 2.13.7 Departments

**You can manage departments in with Data Assets service. The association between departments and tables is maintained in MaxCompute.**

### View departments

- 1. Log on to the DataWorks console and jump to the Data Assets page.**



**Note:**

**To jump to the Data Assets page, replace ide in the URL of DataStudio with asset.**

- 2. In the top navigation bar, click Assets. In the left-side navigation pane, choose Business Management > Departments.**

### Manage departments

**Find the target department and click Edit to modify its information.**

## 2.14 Security Center

### 2.14.1 Overview

**Security Center provides flexible permission management features. It allows you to request permissions and handle permission requests on the GUI, and view and manage permissions. Security Center not only improves data security but also facilitates data permission management.**

**To go to the Security Center page, you can move the pointer over the DataWorks icon in the upper-left corner, and click Security Center.**

**Security Center consists of the following modules: My Permissions, Authorizations, and Approval Center.**

**Currently, Security Center provides the following features:**

- Self-service permission request: Users can select the required tables to quickly initiate a permission request online. This online request mode is more efficient than the original mode in which users need to contact administrators offline.**

- **Permission management:** Administrators can view the users who have permissions on database tables and revoke permissions as required. Users can also revoke unnecessary permissions.
- **Permission request approval:** Before granting permissions to users, administrators approve permission requests initiated by users. This implements a visual and process-based permission management mechanism, and supports a review of the approval process.

In Security Center, you can view permissions on all the tables in an organization, request and revoke table permissions, and approve or reject permission requests.

Each operation in Security Center applies to all the workspaces of a tenant in standard mode and basic mode.

## 2.14.2 My Permissions

On the My Permissions page, you can view your table and field permissions in a workspace, and request or revoke table and field permissions.

View table and field permissions

1. Move the pointer over the DataWorks icon in the upper-left corner, and click Security Center. In the left-side navigation pane, click My Permissions. The Table tab appears.
2. On this tab, you can select a workspace and specify the environment (for a workspace in standard mode) to view all the tables of the workspace in the specified environment. You can also enter a table name in the search box to search for required tables in fuzzy match mode.

You can view the names and owners of tables in the workspace, view your permissions on the tables, and request or revoke table and field permissions.

## Request table and field permissions

**1. Select the tables and fields on which you want to request permissions.**

- Request permissions on a table or some fields in the table

Select the required fields on which you have no permissions in a table and choose **More > Request Permission** in the **Actions** column.

Alternatively, choose **More > Request Permission** in the **Actions** column for a table without selecting any fields to request permissions on all the fields in the table.

**Note:**

You can request permissions on fields only in a workspace with LabelSecurity enabled. If LabelSecurity is disabled for a workspace, you can request permissions only on tables in this workspace.

- Request permissions on multiple tables and fields

Select all the required tables and fields and click **Request Permission**.

**Note:**

You can also click **Request Permission** without selecting any tables or fields, and then select the required tables and fields in the **Table Permission Request** dialog box.

**2. Set the parameters in the Table Permission Request dialog box.**

Parameter	Description
Workspace	The name of the workspace, which is automatically entered based on the information you specified on the <b>My Permissions</b> page. You can change the workspace as required.
Environment	The environment of the workspace.
MaxCompute Project	The name of the MaxCompute project.
Grant To	The account for which you request permissions. You can request permissions for the current account or a production account of another workspace you joined.

Parameter	Description
Reason for Request	The reason why you request permissions.
Objects Requested	The tables on which you request permissions. The tables that you select on the previous page are displayed. You can add tables or delete existing tables as required.

3. After the configuration is completed, click **Submit**. If you do not want to request the permissions, click **Cancel**.

#### Revoke permissions

You can revoke table and field permissions.

- **Revoke field permissions**



#### Note:

- You can revoke permissions on fields only in a workspace with LabelSecurity enabled.
- To revoke permissions on all the fields in a table, you can directly revoke the permissions on the table.

1. Choose **More > Revoke Field Permission** in the **Actions** column for the table on which you want to revoke permissions.
2. In the **Revoke Field Permission** dialog box, select the fields on which you want to revoke permissions.
3. Click **OK**.

- **Revoke table permissions**

1. Choose **More > Revoke Permission** in the **Actions** column for the table on which you want to revoke permissions.
2. In the **Revoke Permission** dialog box, select the permissions you want to revoke.
3. Click **OK**.



## 2.14.3 Authorizations

**On the Authorizations page, a workspace administrator can view the accounts that have permissions on tables and fields in each workspace, and revoke unnecessary table and field permissions.**

**You can move the pointer over the DataWorks icon in the upper-left corner, and click Security Center. In the left-side navigation pane, click Authorizations. On the Table tab that appears, you can view and search for tables in workspaces of the current organization.**

**On the Table tab, you can select a workspace and specify the environment (for a workspace in standard mode) to view all the tables of the workspace in the specified environment. You can also enter a table name in the search box to search for required tables in fuzzy match mode.**

View accounts that have permissions on a table

**On the Table tab of the Authorizations page, click the plus sign (+) in front of a table to view all the accounts that have permissions on the table.**

Revoke table permissions

**Click Revoke Permission in the Actions column for an account to revoke the permissions of the account on the current table.**

View field permissions

**Click View Field Permissions in the Actions column for an account to view the permissions of the account on the fields in the current table.**

Revoke field permissions

**If LabelSecurity is enabled for the workspace, select fields on the Field Permissions page and click Revoke Field Permissions to revoke the permissions on the fields.**

## 2.14.4 Approval Center

On the Approval Center page, you can view your requests and their status, view and handle the requests pending your approval, and view the requests that you have handled.

### My Requests

1. Move the pointer over the DataWorks icon in the upper-left corner, and click Security Center. In the left-side navigation pane, click Approval Center. On the Approval Center page, click the My Requests tab.

On this tab, you can view the information about each of your requests, including Object Type, Workspace, Status, MaxCompute Project, Request Time, and Table.



#### Note:

If a request contains permission requests for tables that belong to different owners, Security Center automatically splits the request into multiple requests by table owner.

2. Click View in the Actions column to view the details about a request.

### Pending My Approval

1. On the Approval Center page, click the Pending My Approval tab.

On this tab, you can view the requests pending your approval. If a request is pending your approval, a red dot appears next to Approval Center and Pending My Approval to remind you.

You can view the information about each of requests pending your approval, including Object Type, Grant To, Request Time, Workspace, MaxCompute Project, and Table.

2. Click Handle in the Actions column to view the details about a request and handle it on the Request Details page. The request details include the progress and objects requested.
3. Enter your comments and click Approve or Reject as required.

Handled by Me

1. On the Approval Center page, click the Handled by Me tab.

On this tab, you can view the information about each request that you have handled, including Object Type, Grant To, Result, Workspace, MaxCompute Project, Table, and Request Time.

2. Click View in the Actions column to view the details about a request. The request details include the progress and objects requested.

## 2.14.5 FAQ

This topic describes the frequently asked questions (FAQs) about the Security Center service of DataWorks.

- Q: What permissions can I request in Security Center?

A: In Security Center, you can request permissions on tables in DataWorks workspaces in the development environment and production environment.

- Q: What is the relationship between Data Management and Security Center?

A: Security Center is a product that upgrades and replaces the permission and security features in Data Management. You can choose Security Center > My Permissions to view the permissions requested or granted by using the `odpscmd grant` command in Data Management.

If you want to request other permissions and handle permission requests on the GUI, go to Security Center and perform operations as required. The Data Management service does not support permission request and approval any more.

- Q: Why cannot I select fields when I request permissions?

A: If LabelSecurity is enabled for a workspace, you can request permissions on fields in this workspace. If LabelSecurity is disabled for a workspace, you can request permissions only on tables in this workspace.

- Q: Who will handle my request?

A: Your request is handled by a workspace administrator or a table owner. After either of them approves or rejects your request, the request is closed.

- **Q: Why do I find two requests on the My Requests page after I submit only one request?**

**A: The tables in your request belong to two owners. In this case, Security Center automatically splits your request into two by table owner.**

- **Q: I request permissions on a field for one month only. Why does the validity period of the permissions become permanent after my request is approved?**

**A: The security level of this field is zero or not higher than the security level of your account.**

- **Q: Why do I obtain permissions on some tables and fields on which I have not requested any permissions?**

**A: The possible causes are as follows:**

- **An administrator has granted the permissions to you by running commands in the DataWorks console.**
- **After your request is approved in Security Center, Security Center also grants you the permissions on fields whose security level is zero or not higher than the security level of your account, even though you have not requested the permissions.**

- **Q: Why does a request disappear from the Pending My Approval tab before I handle it?**

**A: Another workspace administrator or the table owner has approved or rejected the request. The request is closed and no longer needs to be handled.**

- **Q: What can I do if the message "An error occurred in the MaxCompute project" appears when I specify the workspace and environment?**

**A: Send the error message and error code to a workspace administrator for troubleshooting.**

- **Q: Why do I fail to revoke permissions on a field?**

**A: You can revoke permissions only on the fields whose security level is higher than the security level of your account.**

- **Q: Why do I fail to request permissions by using my tenant account?**

**A: By default, a tenant account has all permissions. Therefore, you do not need to request permissions for your tenant account. The tenant account hides**

unnecessary operations such as permission request. This does not affect the use of the tenant account.

- **Q: In Security Center, can I view the permission request and approval records of Data Management?**

**A: Security Center and Data Management have not synchronized permission request and approval records yet. You need to go to Data Management to view the permission request and approval records of Data Management.**

- **Q: Can I revoke permissions based on the request records in Security Center?**

**A: Currently, Security Center is not the only service that provides authorization . To facilitate permission revocation, the Authorizations page in Security Center provides an access control list (ACL) of all users, regardless of the authorization channel. You can revoke any granted permissions without using the request records.**

- **Q: A permission request submitted in Data Management has not been approved yet. Do I need to submit it again in Security Center?**

**A: Security Center and Data Management have not synchronized permission request and approval records yet. You need to submit the permission request again in Security Center.**

- **Q: How do I specify the LabelSecurity parameter for fields?**

**A: You need to go to Data Map to set the LabelSecurity parameter for fields.**

## 2.15 DataOS API

**The DataOS API supports core API operations of DataWorks. The following DataOS API operations are available for use in DataWorks:**

```
GetInstanceSummary
AddResGroupGateWay
Control
CreateConnection
CreateDag
CreateDQCEntity
CreateDQCfollower
CreateDQCRule
CreateManualDag
CreateMetaSpiderJob
CreateResGroup
DeleteConnection
DeleteConnections
DeleteDQCEntity
DeleteDQCfollower
DeleteDQCRule
```

```
DeleteProjectResGroup
DeleteResGroupGateWay
GetComsumeDmu
GetDagDetail
GetDataServiceApiDetail
GetDataServiceAppDetail
GetDefaultTenant
GetDQCEntity
GetDQCfollower
GetDQCRule
GetMetaSpiderJobList
GetNodeDetail
GetNodeUpdateStatus
GetResGroupAk
GetResGroupGatewayList
GetResGroupInitCommand
GetResGroupList
GetResourceId
GetTableColumn
GetTableList
GetTaskLog
GetUserInfo
ListConnection
ListDataServiceApps
ListDataServiceAuthedApi
ListPermission
ListProjectModule
ListProjectModules
ListProject
ListTenantModule
ListUserPermission
ModifyBusiness
ModifyNode
ModifySolution
QueryConnection
RerunTask
ResumeTask
RunMetaSpiderJob
SearchBusiness
SearchManualDagNodeInstance
SearchNodeInstanceList
SearchSolution
SearchTasks
TestConnectivity
UpdateConnection
UpdateDQCfollower
UpdateDQCRule
UpdateResGroupGateWayPolicy
```

**The preceding API operations are standard POP API operations. You can use the methods for calling normal POP API operations to call DataOS API operations.**

**A POP API whitelist is used to control the access to DataOS API operations. To authorize an Apsara Stack tenant account to call these API operations, you need to**

add the account to the whitelist. If the account that you use to make API requests is not in the whitelist, the system returns the following error message:

```
com.aliyuncs.exceptions.ClientException: InvalidApi.NotFound :
Specified api is not found, please check your url and method.
```

This error message indicates that the account is not authorized to call the specified API operation. To resolve this issue, perform the following operations:

Contact the DataWorks technical support staff, and provide them with the Apsara Stack tenant account to be authorized and the API operations to be called. After receiving an encrypted license generated by the DataWorks technical support staff based on the configuration in the runtime environment, use the license to authorize the account to make API requests. For more information, see the following procedure.

#### Procedure

1. Log on to the Apsara Infrastructure Management Framework console.

For more information, see [Operations of basic platforms > Apsara Infrastructure Management Framework > Log on to Apsara Infrastructure Management Framework](#) in *Operations Guide*.

2. In the top navigation bar, move the pointer over Tasks and select Deployment Summary from the drop-down menu.
3. On the Deployment Summary page that appears, click Deployment Details. The Deployment Details page appears.
4. Move the pointer over the base project, click Details, and then choose BasicCluster-A-20190708-376d Machine List > base-baseBizApp > BaseBizMetaservice.
5. On the page that appears, click Terminal in the Actions column for the target host to go to the TerminalService page.
6. On the TerminalService page, click the add icon, and run the `docker ps | grep meta` command on the tab that appears to obtain the ID of the base-biz-metaservice container.

7. Run the docker exec -it d6eb89507a17 bash command on the container and use the license for authorization.

Run the following command to authorize the account to make API requests:

```
curl -X POST -d "license=4AA481CE03EAFA6529ECA131ED6E29AAD511B66A760C33786AD241BE1D90D7007C4056FA26FD13318B92BCFB4B254E42" "http://127.0.0.1/dataos/v1/conf_dataos_api";
```

In this command, the license is provided by the DataWorks technical support staff. If the command is run successfully, it returns {"data": "ok", "errCode": 0, "errMsg": "success", "requestId": "0a04142615555747936747847e031b"}. You can then use the authorized account to make API requests.

#### Call methods

DataWorks provides two domain names for calling DataOS API operations on the premises. For example, if the field domain name is \*.env4b.shuguang.com, the domain name for calling API operations on a non-VPC network is dataworks.env4b.shuguang.com, and the domain name for calling API operations in a VPC is dataworks-vpc.env4b.shuguang.com. You can use the following three methods to call DataOS API operations:

- Use the SDK. Run the following sample code:

```
IClientProfile profile = DefaultProfile.getProfile("default", "AccessKey ID", "AccessKey secret");

DefaultProfile.addEndpoint("default", " default", " dataworks-private-cloud", "dataworks.env4b.shuguang.com"); // The field domain name.
IAcsClient client = new DefaultAcsClient(profile);
GetMetaTableRequest getMetaTableRequest = new GetMetaTableRequest();

getMetaTableRequest.setTableGuid(tableGuid);

GetMetaTableResponse getMetaTableResponse = client.getAcsResponse(getMetaTableRequest);

log.info("GetMetaTable: " + gson.toJson(getMetaTableResponse));
```

- Use CommonRequest. Run the following sample code:

```
DefaultProfile
profile = DefaultProfile.getProfile("default", "AccessKey ID", "AccessKey secret");
IAcsClient client = new DefaultAcsClient(profile);
CommonRequest request = new CommonRequest();

request.setMethod(MethodType.GET);
```



```
request.setDomain("dataworks.env8d.international.com");
request.setVersion("2019-01-17");

request.setAction("GetDefaultTenant");

CommonResponse response = client.getCommonResponse(request);

System.out.println(response.getData());
```

- **Use an HTTP request URL. For example, use the following URL to make an API request:**

```
http://dataworks.env8d.international.com/?SignatureVersion=1.0&Action
=GetDefaultTenant&Format=JSON&SignatureNonce=c8660b3a-33a5-4cc7-9429-
2d9c31fff2ea&Version=2019-01-17&AccessKeyId=mXHVXTiw6AEVtCrX&Signature
=SzHYQ%2BlQsKylbrDkT842gM4UQoY%3D&SignatureMethod=HMAC-SHA1&RegionId=
default&Timestamp=2019-03-19T10%3A39%3A58Z.
```

**For more information about the SignatureNonce parameter in the URL, see the standard POP API documentation.**

## 2.16 App Studio

### 2.16.1 Overview

**App Studio is a tool designed to help you develop data products. It comes with a rich set of front-end components that you can drag and drop to simply and quickly build front-end apps.**

**With App Studio, you do not need to download and install a local integrated development environment (IDE) or configure and maintain environment variables. Instead, you can use a browser to write, run, and debug apps and enjoy the same programming experience as that in a local IDE. App Studio also allows you to publish apps online.**

#### Advantages

**App Studio has the following core advantages:**

- **Data development anytime, anywhere**

**You do not need to download and install a local IDE or configure and maintain environment variables. Instead, you can use a browser to develop data in your office, at home, or anywhere that you can connect to the network.**

- **Editor with complete features**

App Studio provides a browser-based editor that allows you to easily write, run, and debug projects. When you enter the code, App Studio provides code hinting, code completion, and repair suggestions. You can also find all references and the definition of a method to automatically generate code.

- **Online debugging**

App Studio comes with all breakpoint types and operations of a local IDE. It supports thread switching and filtering, variable checking and watching, remote debugging, and hot code replacement.

- **Multi-feature terminal**

You can directly access the runtime environment, which is currently built based on CentOS as the base image. The multi-feature terminal supports all bash commands, including vim and other interactive commands.

- **Collaborative coding**

You and your team members can use App Studio to share the development environment for collaborative coding. Currently, App Studio allows a maximum of eight users to edit the same file of a project online concurrently, improving work efficiency. In the future, the collaborative coding component will support chatting, bullet screen messages, code annotations, videos, and other features to make teamwork efficient and pleasant.

- **Plug-in system**

App Studio supports business plug-ins, tool plug-ins, and language plug-ins.

- App Studio allows you to customize any required menu or add any service portal based on your business needs.
- You can customize project management processes, project types, and templates dedicated to your business.
- You can develop common tools, such as enhanced Git features, code rule scanning, keyboard shortcuts, enhanced editing features, and code snippets, and integrate them into App Studio.
- You can use language plug-ins to enrich the languages supported by App Studio, enabling App Studio to serve users with more languages while addressing your own business needs.

- **Visual building**

**App Studio provides a WYSIWYG designer that has rich components and deeply integrates DataService Studio and DataStudio. Among all components of DataWorks, you can call DataWorks API operations only in App Studio. In addition to calling the API operations, you can quickly build front-end apps by dragging and dropping components and configuring them in the WYSIWYG designer based on the santa file system, developing web apps without code.**

- **Rich templates and flexible project management**

**App Studio provides rich project templates, allowing you to develop your project accordingly with fewer steps and higher efficiency. You can also save your project as a template for future development and use, or share it with other users**

.

## 2.16.2 Get started with App Studio

**To build a data portal, engineers need to develop data, build back-end services, and develop front-end pages. This topic describes the basic features of App Studio and how to use App Studio.**

**Originally, DataWorks is mainly used by data engineers to implement offline or streaming data development. As DataWorks becomes increasingly easy to use, many roles such as algorithm engineers, BI analysts, operators, and product managers who are familiar with SQL can use DataWorks to develop data.**

**App Studio helps different types of users quickly build webpages for data viewing and apps for data query.**

### Create a front-end project

**App Studio provides complete front-end development capabilities that allow you to develop front-end projects in the same way as in a local integrated development environment (IDE). Without the need to master or understand any new concepts, you can create front-end projects in App Studio and develop HTML, CSS, JavaScript , and React files in a way that you are familiar with.**

**1. Create a project based on the sample project.**

- a. Go to the App Studio page and click Projects in the left-side navigation pane.  
On the Projects page, click Create Project from Code.
- b. On the Create Project page, specify Name and Description, and set Select the runtime environment to react-demo.
- c. After the configuration is completed, click Submit.

**2. Set running parameters.**

In the upper-right corner, choose Edit Config > Edit Configurations. In the Run/Debug Configurations dialog box that appears, specify the required parameter. Select the instance type and specify the port number as required. You can use the default configuration unless otherwise required. Then, click OK.

**3. Run the project.**

Click the Run icon in the upper-right corner to run the project. Currently, you can run the `tnpm start` command to start front-end projects. You can seamlessly run projects with the `webpack-dev-server` set up.

During project running, you can view the dependency installation and app startup logs. After the project running is completed, the Preview tab appears on the right sidebar. You can edit and save the code in real time. The edited code takes effect immediately.

**4. Access the project.**

Click the arrow next to the access link to open the project. In App Studio, you can edit and develop front-end projects in the same way as in a local IDE. App Studio supports code completion, method signature, refactoring, and redirection for HTML, CSS, LESS, SCSS, JavaScript, TypeScript, JSX, and TSX files. In addition, you can develop front-end projects based on templates without the need to build any environment or download any dependency.

## Create a back-end project

### 1. Create a project based on the sample project.

- a. Go to the App Studio page and click Projects in the left-side navigation pane.  
On the Projects page, click Create Project from Code.
- b. On the Create Project page, specify Name and Description, and set Select the runtime environment to springboot.
- c. After the configuration is completed, click Submit.

### 2. Set running parameters.

In the upper-right corner, choose Edit Config > Edit Configurations. In the Run/Debug Configurations dialog box that appears, specify the required parameter and then click OK.

You can click Add on the left of the Run/Debug Configurations dialog box to add multiple configurations for running.

### 3. Run the project.

Click the Run icon in the upper-right corner to run the project.

The first time that the project is run takes a longer time because App Studio needs to allocate the ECS instances and initialize the language service. After the running is completed, the Runtime tab appears, showing the access link.

#### 4. Access the project.

Click Open Link to access the project.



Append /testapi to the link and refresh the page.



#### Understand App Studio

The following operations are supported for created projects:

- Top navigation bar
  - Project

From the Project menu, you can configure the project or view detailed information by selecting Character Set or Project Information. Provided information about the current project includes the ID specified by Project ID,

name specified by Project Name, type specified by Project Type, creation time specified by Created At, and UUID.

- **File**

From the File menu, you can create a file or open a recently created file by selecting Create File or Re-Open Most Recent Files.

- **Edit**

From the Edit menu, you can perform common editing operations. To search all the code in the project and open the related file, select Find in Path.

- **Versions**

From the Version menu, you can select Check Out Branch, View Changes, Commit, Log, Connect to Remote Repo, and Merge Abort.

- **Check Out Branch**

In the Check Out Branch dialog box, you can click +Create Branch to create a local branch and push it to the remote repo. You can click a local branch and select checkout from the shortcut menu on the right to switch to the branch. You can also select merge to merge the selected branch to the current branch.

You can click a remote branch and select check out as a new local branch from the shortcut menu on the right to check out the remote branch locally.

Then rename the branch. You can also select merge to merge the selected branch to the current branch.

■ **View Changes**

Click View Changes to view the list of edited files on a local branch in the right-side navigation pane.

■ **Commit**

Click Commit to commit edits on a local branch for staging. You must enter the commit information.

■ **Log**

On the Log page, you can view all commit records of branches and filter them.

■ **Connect to Remote Repo**

You can associate a new project with a remote repo for version control.

- **View**

You can click Toggle Full Screen or press Esc on the keyboard to enter or exit the full screen mode of the page. You can also click Hide Sidebar or Hide Status Bar to hide the right-side sidebar or the status bar. If these bars are hidden, you can click Show Sidebar or Hide Status Bar to show them respectively.

- **Debug**

■ If you create a front-end project, you can set running parameters and create a custom image for it.

■ App Studio supports Java-based debugging. In addition to setting running parameters and adding custom images, you can perform many other



operations for debugging back-end projects. You can also perform full or incremental builds and compile the Main.java file.

- **Settings**

From the Settings menu, you can set the Git configuration to import the Git code to create a project. You can also configure your preference and shortcut keys.

- **Publish**

You can choose Publish > Download Source Code to download the source code.

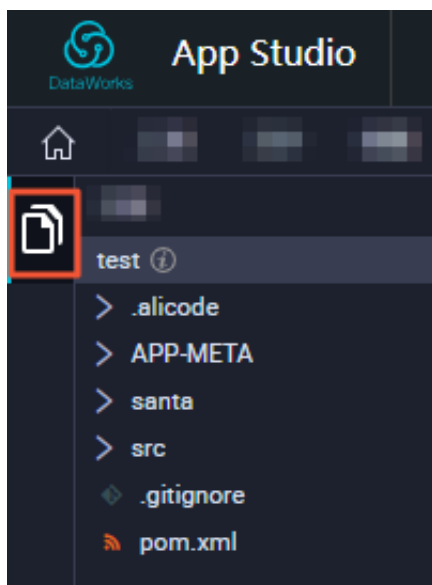
- **Template**

You can choose Template > Manage Templates to go to the My Templates page to manage templates.

- **Left sidebar**

- **Entry**

Click the icon framed in red in the following figure. The project section appears.



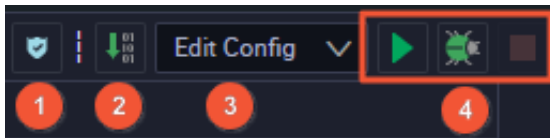
- **Edit section**

Double-click a file that you want to edit. In the Edit section that appears, right-click the code section to perform the following operations.

Operation	Description
Go to Definition	Navigates to the definition page.
Peek Definition	Previews the definition.

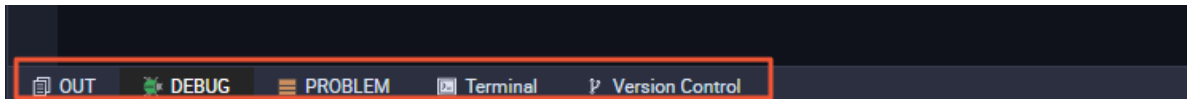
Operation	Description
Find All References	Searches for all references.
Workspace Symbol	Searches for a symbol in the project.
Go to Symbol...	Navigates to the symbol in the project.
Generate...	Generates the code.
Rename Symbol	Renames the symbol.
Change All Occurrences	Changes the name of all occurrences of a symbol throughout the file.
Format Document	Formats the file.
Cut	Cuts the file.
Copy	Copies the file.
Command Palette	Goes to the command palette.

• Icons in the upper-right corner



No.	Feature
1	Alibaba Coding Guidelines
2	Build Program. The build program can only be run when the project is running or being debugged.
3	Run/Debug Configurations. You can set parameters for running or debugging the project.
4	Operations on the project, including running, debugging, or stopping the project.

- **Bottom bar**

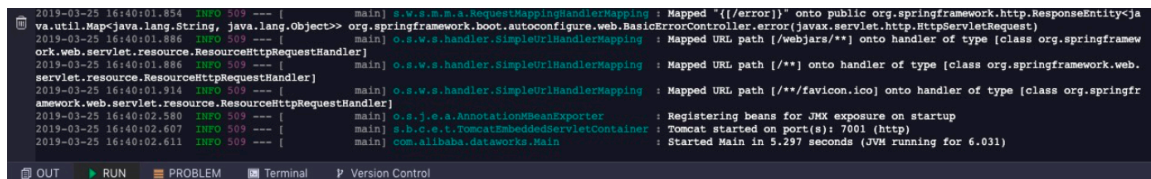


- **OUT tab**

You can click the OUT tab to view the output.

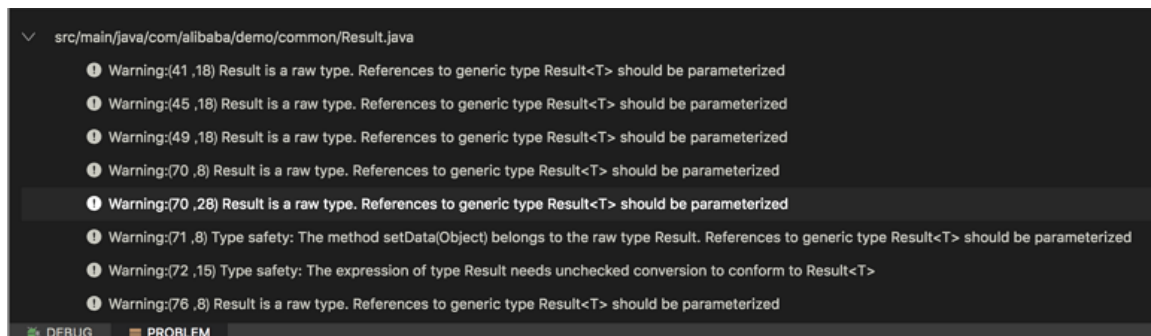
- **RUN or DEBUG tab**

If you click the Run or Debug icon for a project, this tab appears, showing the progress and information of the project.



- **PROBLEM tab**

If you click the Run or Debug icon for a project that has a problem, this tab appears.



- **Terminal tab**

When running or debugging a project, you can click the Terminal tab and run bash or vim commands on the ECS instance.



- **Version Control tab**

You can click the Version Control tab to view the logs and history of the project

.

## 2.16.3 Navigation pane

### 2.16.3.1 View and manage projects

You can create and manage projects on the Projects page.

Go to the App Studio page and click Projects in the left-side navigation pane. On the page that appears, you can view projects that you have created. For more information about how to create template-based and code-based projects, see [Project management](#).

Click a project to go to the project editing page. You can also click Create Template of a project to create a template based on the project.

Create a template

1. Click Create Template of a project.
2. In the Create Template dialog box that appears, set each parameter.

Parameter	Description
Name	The name of the template.
Description	The description of the template.
Class	The class of the template.

3. After the configuration is completed, click OK.

### 2.16.3.2 View and manage templates

You can view all templates created based on projects on the Templates page.

Click a template to go to the template details page. Then, click Code Editor to view the project code that this template is based on.

You can also click Create Project of a template to create a project based on this template.

## 2.16.4 Project management

This topic describes how to create and manage projects.

You can create a template-based or code-based project.

### Create a template-based project

1. Go to the App Studio page and click Projects in the left-side navigation pane. On the Projects page, click Create Project from Template.
2. On the Create Project page, specify Name and Description, and select a template.



**Note:**

- You can select a custom template or a template provided by the system.
- All projects created by using templates support WYSIWYG development.

3. After the configuration is completed, click Submit.

### Create a code-based project

You can create a project by running code. App Studio provides code templates for three types of runtime environments. Select a code template as required.

1. Go to the App Studio page and click Projects in the left-side navigation pane. On the Projects page, click Create Project from Code.
2. On the Create Project page, specify Name and Description, and select a template.
3. After the configuration is completed, click Submit.

### View and manage projects

You can view the created projects on the Projects page.

You can click a project name to go to the project editing page. You can also click Create Template of a project to create a template based on the project.



**Note:**

You can view projects shared by others but cannot create templates based on those projects.

## 2.16.5 Code editing

### 2.16.5.1 Overview

Code editing supports common IDE features, such as automatic completion, code hinting, syntax diagnosis, and global content search.

The following tables list the basic and advanced features that App Studio supports in different languages.

Basic feature	Java	Python	JavaScript and TypeScript
Completion	Supported	Supported	Supported
Hover	Supported	Supported	Supported
Diagnostics	Supported	Supported	Supported
SignatureHelp	Supported	Supported	Supported
Definition	Supported	Supported	Supported
References	Supported	Supported	Supported
Implementation	Supported (coming soon)	Not supported	Not supported
DocumentHighlight	Supported	Supported	Supported
DocumentSymbol	Supported	Supported	Supported
WorkspaceSymbol	Supported	Supported	Supported
CodeAction	Supported (Alibaba Java Guidelines coming soon)	Supported	Supported
CodeLens	References implementation	Not supported	Not supported
Formatting	Supported	Supported	Not supported
RangeFormatting	Supported	Not supported	Not supported
FindInPath	Supported	Supported	Supported

Advanced feature	Java	Python	JavaScript and TypeScript
Rename	Supported	Supported	Supported
WorkspaceEdit	Supported	Not supported	Not supported
UnitTest (quick start)	Supported	Not supported	Not supported
MainClass	Supported	Not supported	Not supported
MainClassQuickStart	Not supported	Not supported	Not supported
ListModules	Supported	Not supported	Not supported

Advanced feature	Java	Python	JavaScript and TypeScript
Generate	Constructor Override Getter and Setter Implement	Not supported	Not supported

### 2.16.5.2 Generate code snippets

Currently, App Studio supports the Java class constructor, getter and setter methods, override methods of the parent class that a child class inherits, and API methods to be implemented.

#### Entry

Perform either of the following operations to generate the Java code:

- Right-click the code section and select Generate.
- Press Command+M on the keyboard. The Java code is automatically generated.

#### Constructor

On the Generate menu, click Constructor.

Select the fields to be included in the constructor and click OK.

The constructor that contains the initialization statement of the fields is generated.

#### Getter and setter methods

Generate the getter and setter methods in a way similar to the constructor.

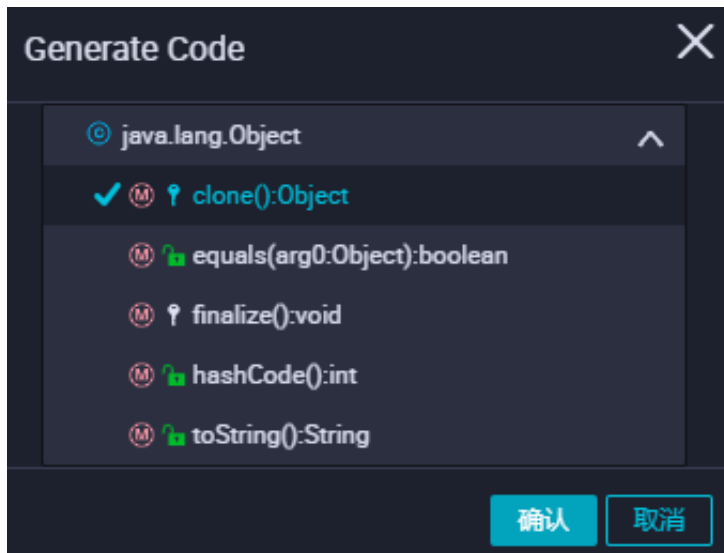


#### Note:

If a Java class does not have any field or the Java class is overwritten by the @data annotation of lombok, the getter or setter method is not required for the Java class. In this case, the Getter, Setter, and Getter And Setter options do not appear on the Generate menu.

#### Override methods

Click Override Methods on the Generate menu. All methods that can be overridden are listed in the Generate Code dialog box.



Select a method. The corresponding method is generated.

### 2.16.5.3 Run UT

App Studio currently supports unit testing (UT), including automatically generating UT code, detecting the entry for UT, running UT code, and displaying the UT result.

Automatically generate UT code

Open the target file, right-click the code editing section, select **Generate** and then click **Create Test**. The UT class file and UT code are automatically generated in the test directory.

Detect the entry for UT



#### Note:

- UT class files must be stored in the `src/test/java` directory. A Java UT class file that is not stored in this directory cannot be identified as the Java UT class.
- For a method annotated with `@Test` annotation, Run Test appears, indicating the entry for UT.

After the Java UT class file is created, add the `@Test` annotation of `org.junit.Test` to the corresponding sample UT method.

Run UT code

Click the Run icon in the upper-right corner. The sample UT starts.



## 2.16.5.4 Search all content of files

App Studio provides the Find in Path feature to support global content search.

Move the pointer over Edit in the top navigation bar and click Find in Path.

You can select Match Case, Words, Regex, or File Mask to set the filter criteria.

You can also click Module or Directory to search files by module or directory.

After selecting a file, you can locate the searched content in the file and open the file in the editor.

## 2.16.6 Debugging

### 2.16.6.1 Configuration and startup

You can configure the entry method, start debugging, and set breakpoints to debug an app.

Configure the entry method

Parameter	Description
Main class	The entry method (which is the main method) you want to start. You can select a value from the drop-down list.
VM options	The parameters for starting a Java Virtual Machine (JVM ), for example, -D, -Xms, and -Xmx.
Program arguments	The startup parameter, which is obtained by the args parameter in the main method.
Environment Variables	The environment variables.
JRE	The Java runtime environment. Default value: 1.8 - SDK.
PORT	The port you want to expose in the app, for example, classic port 7001 or port 8080 for Spring Boot-based projects.
ECS Instance	The type of the ECS instance used for debugging.
Enable Hot Code	This configuration takes effect only in Run mode. By default, the HotCode2 plug-in that Alibaba Cloud provides is used.

Start debugging

Move the pointer over Debug in the top navigation bar and click Start Debugging.

The first startup is slower, because the system needs to prepare the runtime environment and download Maven dependencies for you. When you restart debugging, App Studio skips this process and provides user experience similar to that in a local IDE.

## 2.16.6.2 Online debugging

App Studio supports the online debugging of Java apps and Spring Boot-based web projects.

Before online debugging, you must configure the entry method and start debugging. For more information, see [Configuration and startup](#).

### Exposed services

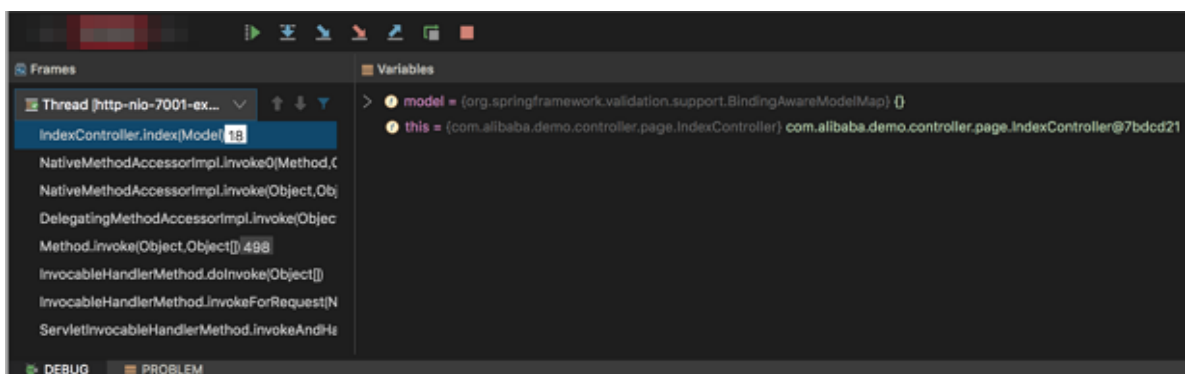
After your app is started, two basic services are provided. You can click the link next to Backend to debug the back-end Java code.

### Panel introduction

- **Output**

The Output panel displays the standard output, excluding System.in, of all apps. It supports the ANSI color and guarantees consistent experience as a local terminal.

- **Call Stack**

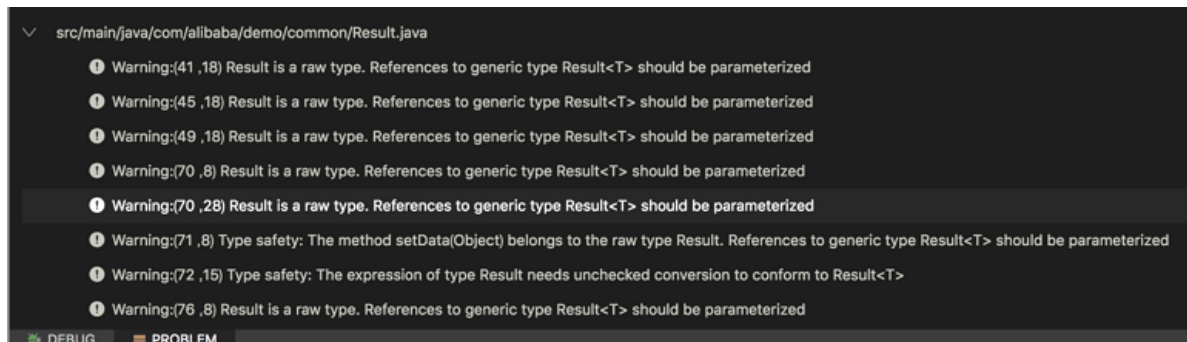


- **Breakpoint**

The Breakpoint panel displays the breakpoints that are currently set. For more information about the breakpoint types and usage, see [Breakpoint types](#).

## • PROBLEM

The **PROBLEM** panel displays compilation problems of apps. You can click a record to go to the corresponding line in the file.



### 2.16.6.3 Breakpoint types

App Studio supports normal line breakpoints, method breakpoints, and exception breakpoints.

Normal line breakpoint

You can click the blank section next to a line in the current file to generate a breakpoint for that line. The breakpoint also appears on the Breakpoint panel.

Method breakpoint

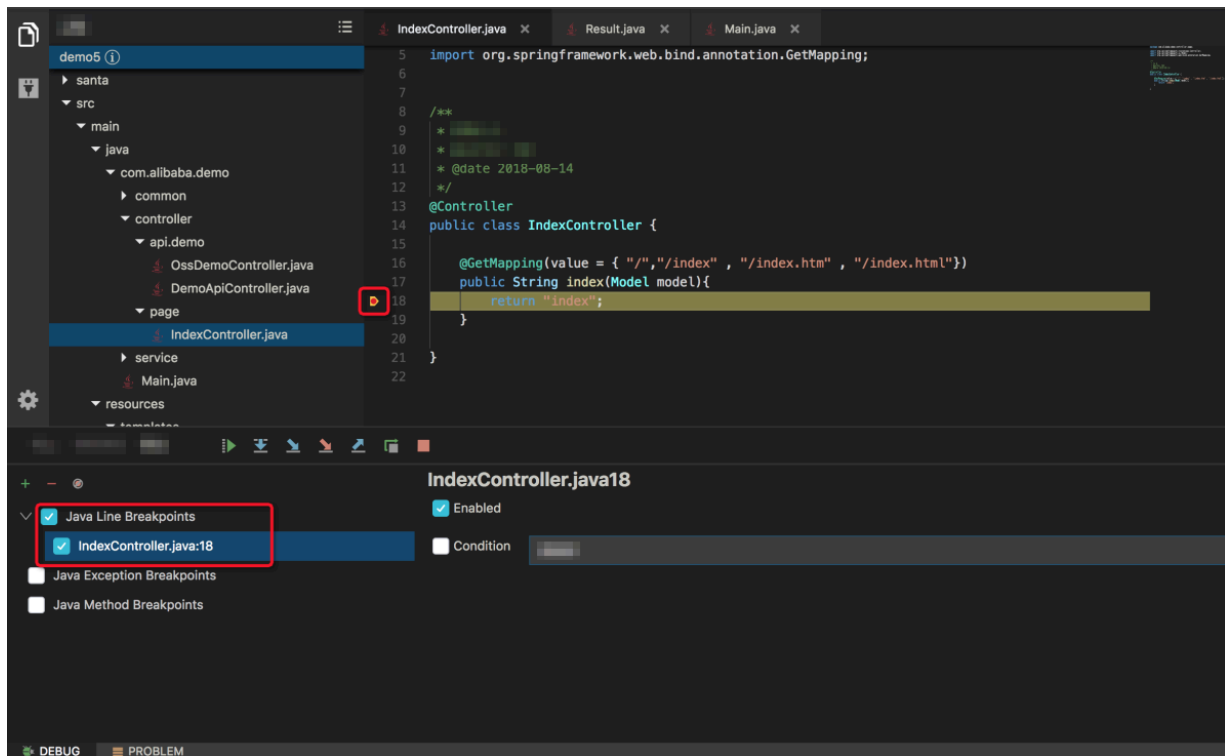
Different from a line breakpoint or an exception breakpoint, a method breakpoint triggers two events, namely, entry and exit. You can manually add a method breakpoint, or set a breakpoint at the place where the method is defined.

If the method breakpoint is triggered, the program stops when stepping into or out of the method.

Exception breakpoint

If an exception breakpoint is set, the program stops when encountering the exception.

As shown in the following figure, after index is triggered, the program stops in line 23 because NullPointerException appears.



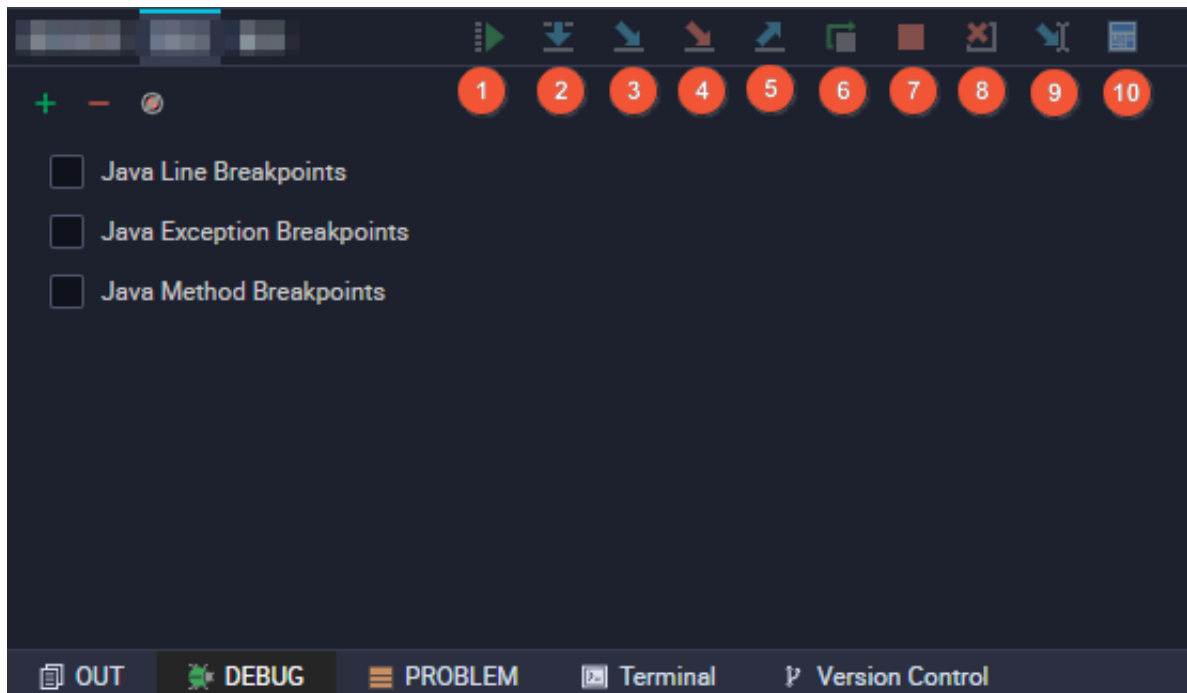
#### 2.16.6.4 Breakpoint operations

The Breakpoint panel displays the breakpoints that are currently set. This topic describes how to operate breakpoints.

Breakpoints can be classified into normal line breakpoints, method breakpoints, and exception breakpoints. For more information, see [Breakpoint types](#).

#### Debugging buttons

You can perform the debugging operations by clicking the following buttons listed in the table:



No.	Feature	Description
1	Continue	Resumes the current breakpoint to continue the current thread.
2	Step Over	Runs to the next line.
3	Step Into	Steps into a method.
4	Force Step Into	Forcibly steps into a method of a class not to be stepped into. Different from Step Into, Force Step Into enables you to step into a method from a built-in Java library.
5	Step Out	Steps out of the current method.
6	Restart	Currently, the Restart button is not perfect enough and may not be able to clean up the program. This button is being optimized.
7	Stop	Stops debugging.
8	Drop Frame	Deletes the current stack and returns to the previous method.
9	Run to Cursor	Runs to the current line of code. You can set a temporary breakpoint in a line.

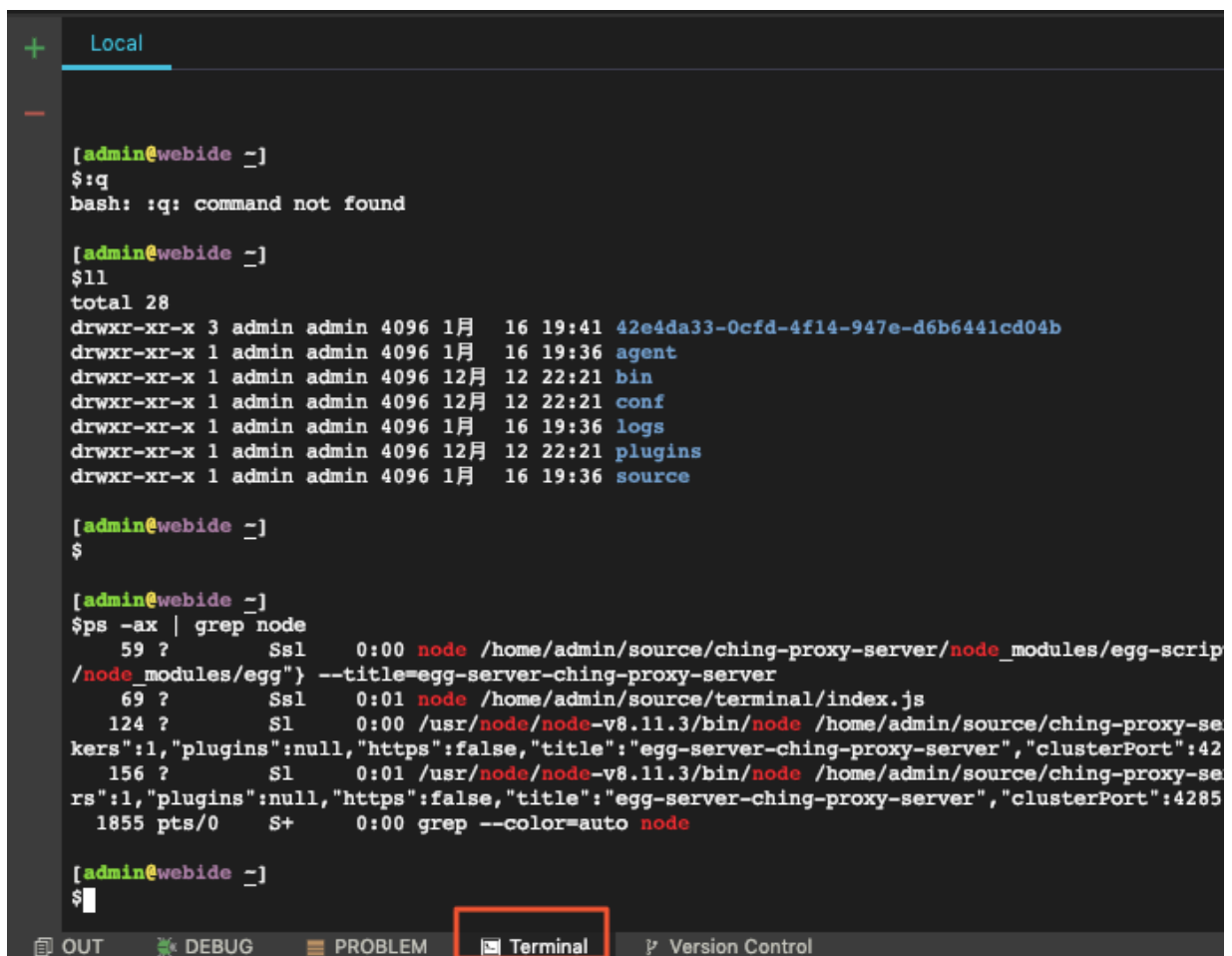
No.	Feature	Description
10	Evaluate Expression	Calculates an expression.

### 2.16.6.5 Terminal

The Terminal tab appears on the bottom of the panel.

App Studio supports common shell commands such as `ls` and `cat` and interactive commands such as `vi` and `top`.

You can also start multiple terminals.



```
+ Local

[admin@webide ~]
$:q
bash: :q: command not found

[admin@webide ~]
$ll
total 28
drwxr-xr-x 3 admin admin 4096 1月 16 19:41 42e4da33-0cfd-4f14-947e-d6b6441cd04b
drwxr-xr-x 1 admin admin 4096 1月 16 19:36 agent
drwxr-xr-x 1 admin admin 4096 12月 12 22:21 bin
drwxr-xr-x 1 admin admin 4096 12月 12 22:21 conf
drwxr-xr-x 1 admin admin 4096 1月 16 19:36 logs
drwxr-xr-x 1 admin admin 4096 12月 12 22:21 plugins
drwxr-xr-x 1 admin admin 4096 1月 16 19:36 source

[admin@webide ~]
$

[admin@webide ~]
$ps -ax | grep node
59 ? Ssl 0:00 node /home/admin/source/ching-proxy-server/node_modules/egg-scrip
/node_modules/egg") --title=egg-server-ching-proxy-server
69 ? Ssl 0:01 node /home/admin/source/terminal/index.js
124 ? Sl 0:00 /usr/node/node-v8.11.3/bin/node /home/admin/source/ching-proxy-se
kers":1,"plugins":null,"https":false,"title":"egg-server-ching-proxy-server","clusterPort":42
156 ? Sl 0:01 /usr/node/node-v8.11.3/bin/node /home/admin/source/ching-proxy-se
rs":1,"plugins":null,"https":false,"title":"egg-server-ching-proxy-server","clusterPort":4285
1855 pts/0 S+ 0:00 grep --color=auto node

[admin@webide ~]
$
```

The screenshot shows the App Studio interface with the Terminal tab selected. The terminal window displays the results of several shell commands: `:q` (command not found), `ll` (listing directory contents), and `ps -ax | grep node` (showing running Node.js processes). The terminal output is color-coded, and the terminal tab is highlighted in the bottom panel.

### 2.16.6.6 Hot code replacement

Using the hot code replacement feature, you can edit the running code of an app and make the edits effective without restarting the app.

For example, after you edit the code while debugging a Spring Boot-based app, you do not need to restart the app. The edited code takes effect once it is saved. App Studio supports this feature by default.

**App Studio also supports hot code replacement while an app is running. To trigger hot code replacement, you only need to save the file without installing any plug-in or manually compiling the file.**

**If you are editing the code in Debug mode, App Studio automatically deletes the current running stack and returns to the method entry.**

Configure hot code replacement in Run mode

**Enable hot code replacement on the Run/Debug Configurations page.**

**After you click Run or Debug, the output information of the HotCode2 plug-in appears on the OUT tab.**

**Save the file after editing it.**

Configure hot code replacement in Debug mode

**You can use the native Java Debug Interface (JDI) to enable hot code replacement in Debug mode. However, due to Java Virtual Machine (JVM) restrictions, hot code replacement is unavailable when a method is added to or deleted from a class. You can save the file to trigger hot code replacement.**



**Note:**

**The native JVM supports hot code replacement for operations such as adding or deleting a class. However, hot code replacement is unavailable when you change the class structure.**

## 2.16.7 WYSIWYG designer

### 2.16.7.1 Basic usage

**This topic describes basic operations in the WYSIWYG designer, including creating a project and building a visual page.**

Create a project

- 1. Log on to the DataWorks console.**
- 2. Move the pointer over the DataWorks icon in the upper-left corner and then choose All Products > App Studio to go to the App Studio page.**
- 3. Click Projects in the left-side navigation pane. On the page that appears, click Create Project from Code.**

4. On the Create Project page, specify Name and Description, and set Select the runtime environment to appstudio.
5. After the configuration is completed, click Submit.
6. Go to the *santa/pages* directory.
7. Click any *santa* file to go to the WYSIWYG designer.

You can also right-click pages and choose Create > Template to develop the page based on a template.

#### Build a visual page

The WYSIWYG designer consists of the component menu and operation panel.

- Component menu

The component menu lists all components that the WYSIWYG designer presets, including layout components, basic components, form components, chart components, and advanced components.

Select a component from the component menu and drag and drop it to the visual operation section. Click the component. The Component Settings panel appears on the right.

On the Component Settings panel, you can configure the component on the Properties, Style, and Advance tabs.

- Operation panel

You can click the corresponding icon on this panel to undo an operation, redo an operation, preview the rendering result, enable the code mode, use the global style, configure the navigation, configure a global data flow, deploy as a template, and save edits.

Click the Configure Navigation icon in the upper-right corner to go to the navigation configuration page. For more information, see [Navigation configuration](#).

#### Configure a global data flow

For more information about how to configure a global data flow, see [Global data flow](#).



- **Configure component properties**

**On the Properties tab, you can visually configure component properties.**

**Based on the rules for configuring component properties, a visual form is generated on the Properties tab. After you configure component properties in this form, the WYSIWYG designer re-renders the component in the visual operation section based on the new properties. You can view the rendering results of the component with different properties in real time.**

- **Configure component styles**

**On the Style tab, you can configure the styles of a component.**

**A visual panel for configuring common styles is provided on the Style tab. On this panel, you can customize the basic styles of a component, including the layout, text, background, border, and effect.**

**After you add or modify the component styles on this tab, the WYSIWYG designer collects all the style settings and re-renders the component in the visual operation section based on the new component style. You can view the component configuration effect in real time.**

- **Configure association between components**

**On the Advanced Settings tab, you can configure association between components.**

**Select a component in the visual operation section and click the Advance tab.**

**The properties of the selected component are listed on the left of the tab. Click the Magnifier icon on the right and select the component to be associated to your selected component.**

**The properties of the associated component appear on the right of the tab.**

**Select a property, for example, searchParams, in the left property list and connect it to a property, for example, requestParams, in the right property list.**

**In this way, any change of the searchParams parameter of the left component is transferred to the requestParams parameter of the right component in real time. This achieves property-based association between the two components.**

Configure the code mode

**By using the code mode, you can implement complex interactions in a more advanced way. For more information, see [Code mode](#).**

Save, preview, run, and hot code replacement

**For more information, see [Save, preview, run, and hot code replacement](#).**

## 2.16.7.2 Code mode

**By using the code mode, you can implement complex interactions in a more advanced way.**

**Click the Code Mode icon in the upper-right corner of the operation panel to enable the code mode.**

**The WYSIWYG designer uses domain-specific language (DSL) at the intermediate layer to switch between the visualization mode and code mode. DSL can be considered as a simplified version of React. The DSL syntax is basically the same as the React syntax.**

**As shown in the code section in the preceding figure, DSL uses a tag to describe a component. The tag properties are the component properties. The property value can be of a simple data type such as a string or a number. The property value can also be an expression. You can enter `state.xxx` to obtain data from the global data flow.**

**The code mode has the following features:**

- **If you drag and drop a component or configure the component properties in the visualization section, the edits are updated in the code in real time.**
- **If you edit the code in the code section, the edits are updated in the visualization section in real time.**
- **The drag-and-drop operation and component property configuration in the visualization section and code edits in the code section can be converted between each other.**

### 2.16.7.3 DSL syntax

Domain-specific language (DSL) is a component-based language developed based on the features of React JSX and Vue templates and is more suitable for UI layout design.

#### JSX

The DSL syntax is similar to the JSX syntax in the React.render method. The following section provides a brief description of JSX:

- You can use `{ }` to switch an HTML scope to a JavaScript scope. In a JavaScript scope, you can write any valid JavaScript expression. The return value appears on the page, for example, `<div>{'Hello' + ' Relim'}</div>`.



#### Note:

You can write any JavaScript expressions such as computing statements or literals in `{ }`.

- An HTML tag is used to switch a JavaScript scope to an HTML scope, for example, `<div>Hello Relim</div>`.
- The HTML scope and JavaScript scope can be nested, for example, `{<div>{'Hello' + ' Relim'}</div>}`.

#### Valid JavaScript expressions

```
// Computing statements
{aaa} // ✓ Variable aaa must be defined.
{aaa * 111} // ✓
{1 == 1 ? 1 : 0} // ✓
{/^123/.test(aa)} // ✓
{[1,2,3].join('')} // ✓
{(()=>{return 1})()} // The self-executing function. ✓

// Literals
{1}
{true}
{[11,22,33]} // ✓
{{aa:"11",bb:"22"}} // ✓
{()=>1} // Describe a function, which is valid but meaningless. ✓
```



#### Note:

If certain complex logic must be implemented by multiple computing statements rather than only one statement, you can wrap the logic in a self-executing function, which must be a valid expression. The following statements provide an example:

```
{(function(){
```

```
// Sum the even digits of a number array.
var input = [1,2,3,4,5,6,7,8,9,10];
var temp = input.filter(i => i % 2 == 0)
return temp.reduce((buf, cur) => buf + cur, 0)
})();
```

Invalid JavaScript expressions

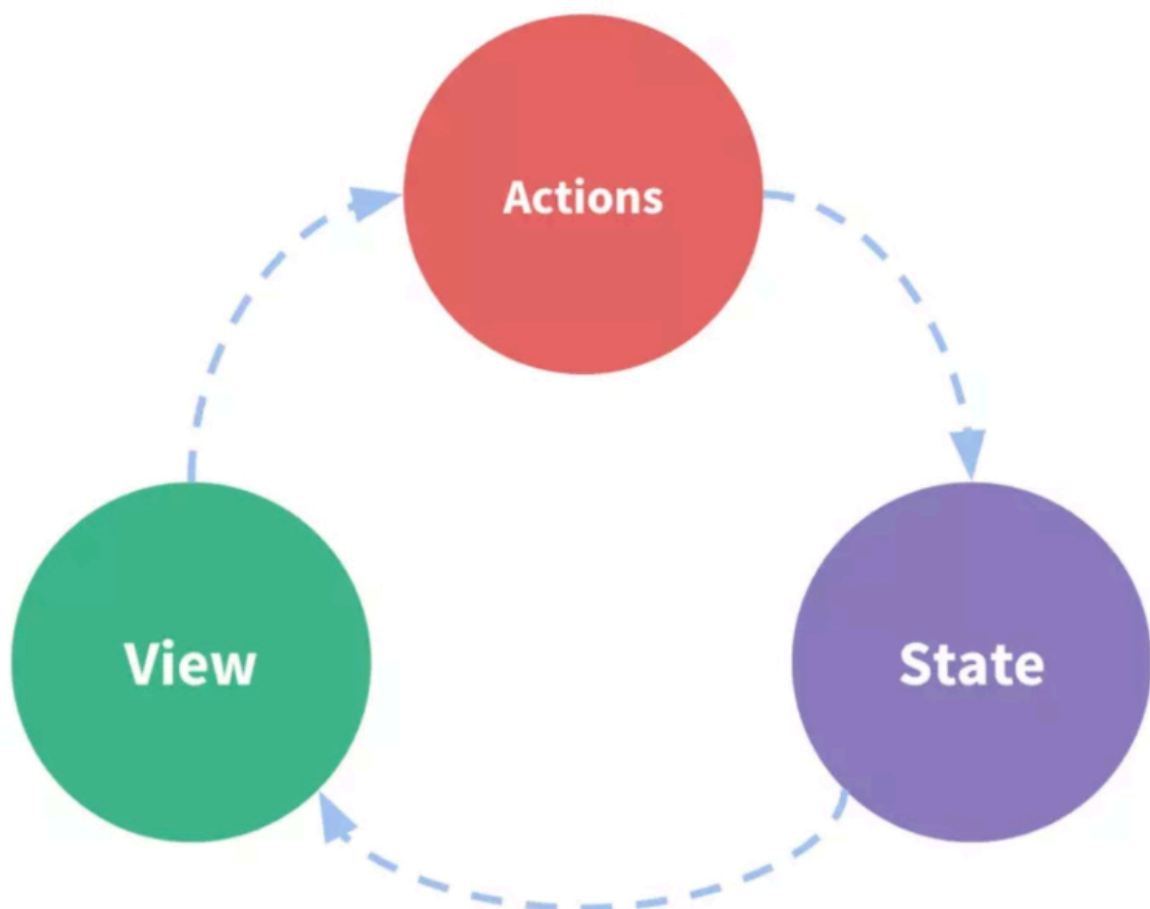
```
{ var a = 1 } // The value assignment statement.
{ aaa * 111; 2 } // Multiple statements separated with semicolons (;).
```

#### 2.16.7.4 Global data flow

A global data flow is used for front-end data management. For multiple components that need to share a state, it is difficult to transfer the state among them. To resolve this issue, you can extract the shared state and use a global data flow to transfer it to all related components.

Principles

In a global data flow, global data is transferred in a globally unique way. Once the data declared in global data changes, the data flow shown in the following figure is executed.



1. A component triggers an action when, for example, a user clicks the component.
2. The action triggers global data changes.
3. Upon the global data changes, components that reference the global state are automatically re-rendered.

#### Scenarios

A global data flow is applicable to the association of two or more components on a page. You can refine public data into global data for unified management, and then use a global data flow to associate two or more components.

#### Configure a global data flow

1. Click the Global Data Flow Settings icon in the upper-right corner of the operation panel.
2. In the Global Data Flow Settings dialog box that appears, set Variable Name and Value.
  - The variable value can be a number, character string, or JSON string.
  - If the variable value is declared as an API endpoint, data obtained from the API is automatically used as the value of the variable name.
3. Click Save.

#### Use a global data flow

- Obtain global data

Use `state.name` in the component to obtain global data.

```
<Input value={state.name} />
```

- Modify global data

Use the `$setState()` method in the component to modify global data.

```
<Input onChange={value => $setState({ name: value })} />
```



#### Note:

You must use the `$setState()` method to modify global data. If you use `state.name = 'new value'`, re-rendering cannot be triggered.

### 2.16.7.5 Save, preview, run, and hot code replacement

**In the WYSIWYG designer, you can perform operations such as saving edits, previewing the rendering result, running an app, or making edits in hot code replacement mode.**

#### Save edits

**The WYSIWYG designer periodically saves your edits. You can also click the Save icon in the upper-right corner of the operation panel to save edits.**

#### Preview the rendering results

**In the WYSIWYG designer, code in the operation section is in the editable status. However, special processing is added for the editable status of some components . For these components, you can run the rendering logic only when the app is running. To preview the rendering result, click the Preview icon in the upper-right corner of the operation panel.**

#### Run an app

**In the WYSIWYG designer, you can open and edit only one santa file at a time. To view the effect of the entire app, click the Run Program icon on the Debug panel of App Studio to run the app.**

#### Make edits in hot code replacement mode

**If you are not satisfied with any page after running the app, you can edit the code in the WYSIWYG designer and save the edits.**

**The edited code takes effect on the running page in hot code replacement mode.**

### 2.16.7.6 Navigation configuration

**This topic describes how to configure the site navigation in the WYSIWYG designer.**

**The WYSIWYG designer provides each app with a public page header, a public bottom bar, and public sidebars, where you can configure various menus and themes. You can also specify whether to display the public header, bottom bar, and sidebars as required.**

**Click the Navigation Settings icon in the upper-right corner of the operation panel to go to the page for configuring the navigation of an app.**

## Configure the public header

You can configure the public header based on your business requirements.

Parameter	Description
Enabled	Specifies whether to display the public header.
Theme	The theme of the public header. You can select a dark or light theme.
Logo Image	The logo image of the site. You can enter an image URL or upload a local image.
Title	The title of the site.
Fix to Page Top	Specifies whether to fix the public header to the top of the page. If you turn on this switch, the public header stays at the top of the page when the page scrolls.
Menu Items	The menu items such as the link name and link URL that are displayed in the public header.

## Configure the sidebars

You can configure the sidebars based on your business requirements.

Parameter	Description
Enabled	Specifies whether to display the sidebars.
Theme	The theme of the sidebars. You can select a dark or light theme.
Enable Folding	Specifies whether the sidebar menus can be hidden.

## 2.17 Data Map

### 2.17.1 Overview

Data Map allows you to search for data globally, use a personal account to manage data, or manage configurations as an administrator.

Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.

- If you prefer a powerful search engine, go to the homepage to search for data. The homepage is displayed by default when you access Data Map. To return to the homepage from other pages, click Data Map in the upper-left corner.

Currently, if you enter keywords to search for data, the search results are more accurate. Data Map also supports other search approaches. For example, if you use Data Map frequently, you can directly select tables in the Recently Viewed Tables and Recently Read Tables sections. You can also select tables in the Most Read Tables and Most Viewed Tables sections. These tables are recommended based on your access records.

- If you would like to search for tables by category, click All Categories. Tables are displayed by category. The number of tables in each category is also displayed.
- If you need to handle personal data, such as modifying your tables or using tools, click My Tables.
- If you are a category administrator or workspace administrator and need to modify the workspace configuration or global categories, click Settings.

## 2.17.2 View the overall information


This topic describes how to view the overall information about a workspace on the Overview page.

Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.

In the top navigation bar, click Overview to go to the Overview page. Data Map collects data of the previous day for the entire organization and generates data on the Overview page offline.

Item	Description
Projects	The total number of projects in the organization.
Tables	The total number of tables in the organization.
Table Storage	The total storage occupied by all tables in the organization.
CPU Usage	The number of compute units (CUs) consumed in one day in the organization. One CU is equivalent to the computing resources consumed by one fully loaded CPU core in one day.



Item	Description
Project Lineage	The chart that shows the relationships between projects in the organization. An arc in the chart represents a project. Two projects are connected if they have the lineage relationship.
Details	The lineage relationship between projects in the organization. The first column indicates a project in which the ancestor table is located, the second column indicates a project in which the descendant table is located, and the third column indicates the number of lineage relationships between the two projects.
Top Projects by Table Storage	The top 10 projects that occupy the most storage space in the organization.
Top Tables by Occupied Storage	<p>The top 10 data tables that occupy the most storage space in the organization. You can click a table name to go to the details page of the table.</p> <div> <b>Note:</b> The logical storage space occupied by projects and tables is collected in a T+1 manner. The numbers next to the project and table names indicate the sizes of the occupied logical storage space. Besides the table storage volume, the project storage volume includes the storage volumes of resources, data in the recycle bin, and other system files. Therefore, the project storage volume is larger than the table storage volume. The table storage volume is charged based on the logical storage rather than the physical storage.</div>
Most Frequently Used Tables	The top 10 most frequently referenced tables in the organization. You can click a table name to go to the details page of the table.

### 2.17.3 Manage data


This topic describes how to manage data on the My Tables page of Data Map.

Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.

In the top navigation bar, click **My Tables**. Then, you can view data on the **Owned by Me**, **Managed by Me as Workspace Administrator**, **Managed by Tenant Account**, and **My Favorites** pages. You can also manage the permissions on the relevant data.

#### Owned by Me

In the left-side navigation pane, click **Owned by Me**. On the page that appears, you can search for data based on the table name, environment, project or data store, and visible range. You can also view the details about a table or perform relevant operations on the table.

Parameter	Description
Table Name	The name of the table. You can click a table name to go to the table details page.
Display Name	The display name of the table. You can click the icon next to a display name to modify it.
Project/Data Store	The project or data store of the table. Tables have different suffixes when they are deployed in different environments. For example, <code>_dev</code> indicates the development environment.
Environment	The environment type. Two environment types are available: development environment and production environment.
Storage	The amount of data that is stored.
TTL (Days)	The time to live (TTL) of the table, which is the same as that you set when creating the table.
Actions	<p>The operations that you can perform. In the Actions column for a table, click the buttons to perform the corresponding operations, such as deleting the table, changing the category, hiding the table, and automatically detecting the table.</p> <div> <b>Note:</b> If you hide a table, the Request Permission button does not appear on the details page of the table.</div>

#### Managed by Me as Workspace Administrator

In the left-side navigation pane, click **Managed by Me as Workspace Administrator**. On the page that appears, you can search for data based on the table name,

environment, and project or data store. You can also view the details about a table or perform relevant operations on the table.

Parameter	Description
Table Name	The name of the table. You can click a table name to go to the table details page.
Display Name	The display name of the table. You can click the icon next to a display name to modify it.
Project/Data Store	The project or data store of the table. Tables have different suffixes when they are deployed in different environments. For example, _dev indicates the development environment.
Environment	The environment type. Two environment types are available: development environment and production environment.
Storage	The amount of data that is stored.
TTL (Days)	The TTL of the table, which is the same as that you set when creating the table.
Actions	The operations that you can perform. In the Actions column for a table, click the buttons to perform the corresponding operations, such as deleting the table and changing the category.

Managed by Tenant Account

In the left-side navigation pane, click Managed by Tenant Account. On the page that appears, you can search for data based on the table name, environment, and project or data store. You can also view the details about a table.

Parameter	Description
Table Name	The name of the table. You can click a table name to go to the table details page.
Display Name	The display name of the table. You can click the icon next to a display name to modify it.
Project/Data Store	The project or data store of the table. Tables have different suffixes when they are deployed in different environments. For example, _dev indicates the development environment.

Parameter	Description
Environment	The environment type. Two environment types are available: development environment and production environment.
Storage	The amount of data that is stored.
TTL (Days)	The TTL of the table, which is the same as that you set when creating the table.
Favorites	The number of times that users add the table to favorites.
Views in Last 30 Days	The number of times that users browse the table in the last 30 days.
Created At	The time when the table was created.

### My Favorites

After adding a table to favorites, you can view the table information on the My Favorites page.

You can click Remove from Favorites to remove a table from your favorites.

### Permissions

In the left-side navigation pane, click Permissions. The Permissions page appears.

- For more information about permission management, see [Manage permissions](#).
- You can click Request Permission in the upper-right corner to apply for permissions on functions or resources. For more information, see [Apply for data permissions](#).

## 2.17.4 View table details

This topic describes how to view the details about a table.

Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.

You can click the name of a data table in any list on the homepage to go to the details page of the table.

On the details page, you can view the basic information, business information, permission information, technical information, detailed information (including the

fields, partitions, and change history), output information, lineage information, reference record, and usage notes of the table. You can also preview the table.

Apply for table permissions

**For more information, see [Apply for table permissions](#).**

Add a table to favorites


**To add a table to favorites, click Add to Favorites under the table name. To view a table after it is added to favorites, follow these steps: On the Data Map page, click My Tables in the top navigation bar. On the page that appears, click My Favorites in the left-side navigation pane.**

Access DataService Studio

**Click Create API in DataService Studio under the table name to go to the DataService page. For more information, see [Overview](#).**

Basic information

**In the Basic Information section, you can view the number of reads, favorites, and views. You can also check the output task, MaxCompute project name, owner name, creation time, time to live (TTL), storage capacity, description, and tags of the table.**

- **You can click the name of the MaxCompute project to go to the project details page.**
- **You can click  next to Tags to add tags for the table.**

Business information

**In the Business Information section, you can view the DataWorks workspace name, environment type, and category.**

Permission information

**In the Permissions section, you can view your permissions on tables. To apply for more permissions, click More on the right side. The Request Permission dialog box appears.**

Technical information

**In the Technical Information section, you can view the technical type, last DDL change time, last data change time, last data view time, and compute engine information.**

- The default time format is yyyy-mm-dd hh:ss:mm.
- Click View next to Compute Engine. A dialog box appears, displaying information about the compute engine.

#### Detailed information

On the Content tab, you can view the metadata of the table, including the definition, popularity, and security level. You can also check the schema changes and whether a field is a primary key or foreign key.

- Field information

On the Fields tab, you can view the name, type, description, and popularity of fields. You can also check whether a field is a primary key or foreign key. In addition, you can edit field information, such as the display name and description, and determine whether to set a field as the primary key.

Operation	Description
Download Field Information	Click this button to download the corresponding field information.
View DDL Statement	Click this button to view the corresponding table creation statements in the dialog box that appears.
Generate SELECT Statement	Click this button to view the corresponding SELECT statements in the dialog box that appears.

- Partition information

On the Partitions tab, you can view the name, number of records, storage volume, creation time, and last update time of each partition of the table.

- Change history

On the Change History tab, you can view the partition name, change type, granularity, time, and operator involved in each change.

You can also select a change type from the drop-down list to filter change records

.

## Output information

**If the table data changes periodically with the corresponding task, you can view the change status and data that is continuously updated.**

## Lineage information

**On the Lineage tab, you can view the source and destination of data and manage the lineage information with ease.**



### Note:

**Only the users who have purchased the DataWorks standard edition or a more advanced edition can view the table lineage information.**

- **Table Lineage:** You can search for the ancestor and descendant tables of a table based on the GUID.
- **Field Lineage:** You can filter data by field.

## References

- **Foreign Key References:** On this tab, you can check the number of users who reference the data.
- **References in Clause:** On this tab, you can view the reference record in a line chart.

## Data preview

**On the Data Preview tab, you can preview the data information of the current table.**



### Note:

**Only authorized users can preview tables in the production environment. If you do not have the corresponding permissions, click Apply to go to the application page.**

## Usage notes

**On the Usage Notes tab, you can edit the usage notes and view historical versions. You can also learn the relevant information based on the description of the data.**

## 2.17.5 Manage permissions

On the Permissions page, you can manage the applications for permissions on tables, resources, and functions. The page consists of the following tabs: To Be Approved, Submitted by Me, and Handled by Me.

Log on to the DataWorks console. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.

In the top navigation bar, click My Tables. On the page that appear, click Permissions in the left-side navigation pane. Then, you can view the To Be Approved, Submitted by Me, and Handled by Me tabs.

### To Be Approved

Only administrators can access the To Be Approved tab to view and approve the applications for permissions on tables, resources, and functions in all workspaces.

### Submitted by Me

On the Submitted by Me tab, you can view historical permission applications that you submitted.

### Handled by Me

If you log on to the system as an administrator, you can click the Handled by Me tab to view the processed applications for permissions on tables, resources, and functions in all workspaces.

## 2.17.6 Apply for data permissions

This topic describes how to apply for data permissions.

DataWorks supports the following three data types:

- **Table:** data tables.
- **Function:** user-defined functions (UDFs) that can be used in SQL statements.
- **Resource:** such as text files or MapReduce JAR files.

DataWorks strictly controls permissions on these three types of data. You must apply for the required permissions before using the data.

### Apply for the permission to preview table data

1. **Log on to the DataWorks console.**



2. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.
3. On the homepage, search for the target data table and click its name to go to the details page of the table.
4. On the table details page, click Request Permission.
5. Set all parameters in the Request Permission dialog box.

Parameter	Description
Table	The table on which you want to apply for permissions. Use the default value.
Object Type	The type of object on which you want to apply for permissions. Valid values: Table and Field.
Grant To	The user to which the permission is granted. Valid values: Current Account and Specified Account.
Validity Period	The validity period of the requested data permission. If this parameter is not specified, the permission is permanently valid.
Reason	The reason for applying for the permission. Enter a brief reason for faster approval.

6. After the configuration is completed, click Submit. After the application is approved, you can preview the table data.

**Note:**

To view the application status, follow these steps: On the Data Map page, click My Tables in the top navigation bar. On the page that appears, click Permissions in the left-side navigation pane. Then, click the Submitted by Me tab.

Apply for function and resource permissions

1. On the Data Map page, click My Tables in the top navigation bar. On the page that appears, click Permissions in the left-side navigation pane.
2. Click Request Permission in the upper-right corner.
3. Set all parameters in the Request Permission dialog box.

Parameter	Description
Object Type	The type of object on which you want to apply for permissions. Valid values: Functions and Resources.

Parameter	Description
Grant To	<p>The user to which the permission is granted. Valid values: Current Account and Specified Account.</p> <ul style="list-style-type: none"><li>• If you select Current Account, the permission will be granted to you after the application is approved.</li><li>• If you select Specified Account, you must set Account. After the application is approved, the permission is granted to the specified user.</li></ul>
Project Name	The name of the MaxCompute project that contains the requested function or resource. The project must belong to the current organization. Fuzzy match is supported.
Function Name or Resource Name	The name of the function or resource in the project. Enter the full name of the resource, including the file suffix, such as my_mr.jar.
Validity Period	The validity period of the applied permission, in days . If this parameter is not specified, the permission is permanently valid. When the validity period expires, the system automatically revokes the permission.
Reason	The reason for applying for the permission. Enter a brief reason for faster approval.

4. After the configuration is completed, click Submit and wait for approval. To view the application status, follow these steps: On the Data Map page, click My Tables in the top navigation bar. On the page that appears, click Permissions in the left-side navigation pane. Then, click the Submitted by Me tab.

## 2.17.7 Manage configurations

This topic describes how to manage configurations on the Settings page of Data Map.



1. Log on to the DataWorks console.
2. Move the pointer over the DataWorks icon in the upper-left corner and select Data Map. The Data Map page appears.
3. On the Data Map page, click Settings in the top navigation bar.

The Settings page consists of the Manage Categories and Manage Workspaces tabs.

## Manage categories

**On the Manage Categories page, you can create a category and add tables to the category. By adding tables to categories, you can manage tables more efficiently.**

- 1. Move the pointer over Categories and click + next to Categories to add a level-1 category.**
- 2. Click + next to a level-1 category to add a level-2 category.**

**A maximum of four category levels are supported. You can click  to rename a category or click  to delete a category.**

- 3. After a category is configured, you can perform the following operations:**

- Add Tables:** You can only add tables that are not in the category. If a table has been removed from a category, you can add it to the category again.
- Search:** You can search for tables by table name or by project or data store.
- Remove from Category:** You can remove one or more tables from a category.

## Manage workspaces

**On the Manage Workspaces page, you can view the workspaces for which you serve as the owner or administrator. In addition, you can turn on or off the Preview Table Data in Development Environment or Preview Table Data in Production Environment switch in the Manage MaxCompute Tables section.**



### **Note:**

**For a workspace in basic mode, only Preview Table Data in Production Environment is available.**

## 3 Realtime Compute

---

### 3.1 What is Realtime Compute?

Realtime Compute is a big data processing platform that provides real-time analysis tools for streaming data based on Apsara Stack. By using Flink SQL statements, you can create streaming data analysis and computing jobs without the need to develop the underlying logic for streaming data processing.

Currently, every industry is facing the challenge of an increased demand for up-to-date data. This requires software applications to improve the efficiency of data processing. In traditional models for big data processing, online transaction processing (OLTP) and offline data analysis are separately performed at different times. These models cannot satisfy the growing demand for real-time big data processing.

Realtime Compute comes from the strict demand for the timeliness of data processing. The business value of data decreases as time passes by. Therefore, data must be computed and processed as soon as possible after it is generated. In traditional models, accumulative data up to the current time is processed in the computing cycle of an hour or a day. Apparently, such data processing methods cannot meet requirements of real-time computing. Batch processing cannot meet the business needs in the scenarios where an extremely low processing delay is required. These scenarios include real-time big data analysis, early warning and risk control management, real-time forecasting, and financial transactions. Realtime Compute enables real-time processing over data streams. With Realtime Compute, you can achieve a short data processing delay, easily implement real-time computational logic, and greatly reduce computing costs. This helps you meet the business needs for real-time processing of big data.

#### Streaming data

Big data can be viewed as a series of discrete events. These discrete events form event streams or data streams along a timeline. Unlike offline data, streaming data is continuously generated by thousands of data sources. It is typically sent in data records simultaneously and in small sizes. Any kind of data is produced as a stream of events. Streaming data includes a wide variety of data, such as log files

generated by customers using your mobile or web applications, online purchases, in-game player activities, information from social networks, financial trade centers, geospatial services, and telemetry from connected devices in data centers.

Realtime Compute has the following features:

- **Real-time and unbounded data streams.** Realtime Compute processes data streams in real time, which are continuously generated from data sources. Streaming data is subscribed to and consumed in the order of the time when it is generated. Data streams are continuously and permanently integrated into the Realtime Compute system. For example, in scenarios where Realtime Compute processes data streams from website visit logs, the log data streams continuously enter Realtime Compute before the website is shut down. In Realtime Compute, unbounded data streams are processed in real time.
- **Continuous and efficient computing.** Realtime Compute is an event-driven system where unbounded event streams or data streams continuously trigger real-time computations. Each streaming data record triggers a computing task and is continuously processed in real time.
- **Real-time integration of streaming data.** Realtime Compute enables you to write the result data of stream processing to data storage systems. For example, the result data can be directly written to the RDS result table, which allows you to easily view the result data in the corresponding reports. Realtime Compute enables the result data to be continuously written to data storage systems in real time.

## 3.2 Quick start

### 3.2.1 Log on to the Realtime Compute console

This topic describes how to log on to the Realtime Compute console.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

#### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/`manage, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username super. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. In the left-side navigation pane, choose Big Data > Realtime Compute.
6. Select the region and department from the Region and Department drop-down lists.
7. Click Console. The administrator page appears.
8. Click Development Platform.

## 3.2.2 Real-time security monitoring

### 3.2.2.1 Overview

With the wide application of digital technologies, every industry is facing the ever-increasing demand for data security, especially for real-time monitoring and alerting. To monitor streaming data and report alerts in real time, you need to ensure that the data is accurate and is processed instantly after it has been generated. To address these challenges, Realtime Compute allows you to perform JOIN operations on source tables of streaming data and dimension tables that include blacklists. The following sections describe a use case of real-time monitoring and alerting.

### 3.2.2.2 Preparations

Before proceeding with the development process in the Realtime Compute console, you need to create a source table and a result table in the upstream and downstream data storage systems, and upload data to the source table.

#### Context

To simplify operations, the incoming streaming data is organized based on a two-dimensional table `datahub_IpPlace`.

Table 3-1: `datahub_IpPlace`

Field	Type	Description
<code>name</code>	VARCHAR	The name.
<code>Place</code>	VARCHAR	The place.

The dimension table `rds_dim` is described as follows.

Table 3-2: `rds_dim`

Field	Type	Description
<code>name</code>	VARCHAR	The name.
<code>Place</code>	VARCHAR	The place.

A result table named `rds_IpPlace` is obtained after a JOIN operation is performed on the `datahub_IpPlace` and `rds_dim` tables. This result table is described as follows.

Table 3-3: `rds_IpPlace`

Field	Type	Description
<code>name</code>	VARCHAR	The name.
<code>Place</code>	VARCHAR	The place.

Create source and result tables

#### Procedure

1. Log on to the DataHub console. For more information, see the relevant documentation.
2. Create a DataHub project.
3. On the page that displays the project details, create a topic.

4. Create an RDS instance. For more information, see the relevant documentation.
5. Log on to the RDS console, select an instance where the result table is stored, click the icon in the Actions column, and then click View Details.
6. On the Basic Information page, click Log On to DMS.
7. Log on to the RDS database, and create a result table in the database.
8. Upload data to DataHub.

Log on to the DataHub console, click Data Collection in the left-side navigation pane, and then click Upload File. For more information, see *Upload local files in DataHub User Guide*.

9. Upload data to the dimension table `rds_dim`.

For more information, see *Import data in DataHub User Guide*.

### 3.2.2.3 Development

After you have created source and result tables in the external data storage systems, you need to [create references to them](#) in the Realtime Compute console before proceeding with the development process.

#### Context

After the data has been collected, you can continue to edit Flink SQL statements. For beginners, Realtime Compute provides a sample job named `bj_dim_join` in the Sample folder on the Development page of the Realtime Compute console. This sample job is created to display product ranking. You can click the sample job to view the Flink SQL statements.

#### Procedure

1. [Log on to the Realtime Compute console](#).
2. Click Development in the top navigation bar.
3. Click Create File. The Create File page appears.



Note:



Table 3-4: Field description

Field	Description
File Name	The name of the file. The specified name must be 3 to 64 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.
File Type	The type of the file. Valid values: BLINK_DATASTREAM, BLINK_TABLEAPI, and BLINK_SQL.
Engine Version	The version of the engine. Typically, you can keep the default setting.
Storage Path	The folder of the file. You can click the icon on the right side of an existing folder and create a subfolder.

4. Reference a DataHub table as the source table. You can use the DataHub parameter settings and schema information that are automatically generated on the Data Storage tab. For more information, see [Register a DataHub project](#).
5. Double-click a DataHub project in the DataHub Data Storage folder.
6. In the dialog box that appears, click Reference as Source Table on the top.
7. Create a reference to the RDS dimension table. You need to specify RDS parameters and enter schema information. The reason is that currently you cannot register dimension tables on the Data Storage tab. The sample code is as follows:

```
create table rds_input (
 place varchar,
 `name` varchar,
 primary key (place),
 period for system_time
) with (
 type = 'rds',
 url = 'yourURL',
 tableName = 'yourTableName',
 password = 'yourPassword'
);
```

8. Reference an RDS table as the result table. You can use the RDS parameter settings and schema information that are automatically generated on the Data Storage tab. For more information, see [Register an RDS instance](#).
9. In the dialog box that appears, click Reference as Result Table on the top.

**10. Edit SQL statements for the Realtime Compute logic on the Development page.**

```
INSERT INTO rds_output
SELECT
 t.`name`,
 w.place
FROM datahub_input1 as t
JOIN rds_intput FOR SYSTEM_TIME AS OF PROCTIME() as w
ON t.place = w.place
```

**11. Debug the Flink SQL code.**

**12. Publish the SQL file for the job.** After the computational logic has been verified in the debugging phase, click **Publish** on the Development page to publish the SQL file for the job. Then, you can view the job on the Administration page of the Realtime Compute console, and manage the job in the production environment, such as starting the job.

### 3.2.2.4 Administration

After you have created and published the SQL file for the job, you can manage the job on the Administration page. For example, you can start, suspend, terminate, publish, or take the job offline.

**Procedure**

1. On the Administration page, click **Start** for the `bj_dim_join` job.
2. Specify **Start Time for Reading Data**. Data entered Realtime Compute, so we recommend that you set the starting time earlier than the current time.

The **Start Time of Reading Data** parameter specifies the start time of reading data from the source data store.

3. Click **OK**.
4. View the result table in the RDS database. The output data in the result table is consistent with the result data of the Realtime Compute job. In this way, you can perform an end-to-end verification of the SQL code.

## 3.2.3 Frequently used words

### 3.2.3.1 Overview

Statistical analysis of frequently used words is widely applied in diverse fields, including the analysis of frequently used words in search engines, forums, and tags. For example, you can easily view the latest and most frequently searched words in microblogging websites through real-time statistics. Statistical analysis

of frequently used words is, at its core, a simple word count job. In word count jobs for streaming data, real-time processing logic is used to analyze and display frequently used words in real time.

If you are new to working with big data computing, a word count job is for you to easily get started. The word count job in big data computing is similar to a `Hello , World!` program that is often the first program that a developer learns to write. The following topics take a word count job in Realtime Compute as an example to describe how to create a word count job based on real-time processing logic. This example helps you quickly get familiar with basic Flink SQL syntax and basic operations of Realtime Compute jobs, such as creating an SQL file for a job and publishing the job.

### 3.2.3.2 Code development

This topic uses a word count job as an example to describe how to create a Realtime Compute job.

#### Prerequisites

Before creating a word count job, you have created a source table named `stream_source` and a result table named `stream_result` in external data storage systems. The `stream_source` table includes only one column. The column name is `word` and the type of data in the column is `string`. The `stream_result` table includes two columns. One column is named `word` and its data type is `string`. The other column is named `cnt` and its data type is `bigint`. Then, register the data storage systems with the two tables in Realtime Compute. To create a word count job, follow these steps:

1. Log on to the [Realtime Compute console](#).
2. Click Development in the top navigation bar.
3. Right-click the Sample folder.
4. Select Create File. The Create File page appears.
5. Set related fields.
  - File Name: Set the value to `wordcount`.
  - File Type: Set the value to `FLINK_STREAM/SQL`.
  - Storage Path: Keep the default setting.

## 6. Enter the following code in the development section.



### Note:

**In the Flink SQL statements for the word count job, declare that the string type for the referenced table is varchar.**

```
create table stream_source (word varchar);
create table stream_result (word varchar, cnt bigint);
insert into
 stream_result
select
 t.word,
 count (1)
from
 stream_source t
group by
 t.word;
```

The following section explains the SQL code.

Line 1 creates a reference to the source table `stream_source`.



### Note:

**Streaming data continuously enters Realtime Compute and triggers stream processing jobs. Each data record or batch of data from the `stream_source` table triggers a stream processing procedure.**

Line 2 creates a reference to the result table `stream_result`. The `stream_result` table is used to store the computing results of the word count job.



### Note:

**Realtime Compute does not have built-in components for data storage, and result data is stored in external data storage systems, such as RDS and Table Store. This line of code creates a reference to a result table that is used to store result data.**

Lines 3 through 11 implement the computational logic: Realtime Compute reads data from the `stream_source` table and counts how often words occur based on incoming data records.



### Note:

**Flink SQL supports standard SQL, which allows you to easily and cost-effectively create Realtime Compute jobs.**

A word count job is implemented in the same way in Realtime Compute as that for batch processing jobs. The only difference lies in that a word count job continuously processes unbounded data streams before you terminate the job.

### 3.2.3.3 Code debugging

Realtime Compute provides the high-performance debugging feature to verify SQL statements. You can simulate and debug data stores that store data streams or static data and export the outputs to a data sink.



**Note:**

- To avoid negative impacts on online storage systems, Realtime Compute is not allowed to read data from these systems during the debugging process. You must prepare test data in input tables before debugging.
- The outputs of all INSERT operations are only exported to the screen that is not connected to the online system.

#### Debugging method

1. Click **Debug** on the top of the **Development** page.
2. On the **Debug File** page, click **Download Template** and edit the template according to your debugging rules.



**Note:**

The file uploaded for debugging must meet the following requirements:

- The file must satisfy the following conditions: 1. The file size cannot exceed 1 MB. 2. The file includes a maximum of 1,000 records.
- The file must use the UTF-8 encoding.
- You cannot use commas (,) in test data, because the file uses the comma-separated values (CSV) format.
- Numeric values can be displayed only in the general format, and cannot be displayed in scientific notations.

3. Click **Upload** to upload the file.
4. Click **OK**.
5. View the debugging result in the output window.

## Debugging file for the word count job

**Note:**

The file for debugging uses the CSV format. We recommend that you use the following software to open and modify the template:

- Excel for Windows users
- Vim or Sublime Text for MacOS users (To avoid adding irrelevant fields during the modification of CSV files, Number is not recommended.)

## Test data

You can download [test data](#), and upload the data on the Debug File page.

**Note:**

The *test data for statistical analysis of frequently used words* is not available for download in the PDF file. You can contact system administrators to download the test data.

## View the debugging result

Real-time computing is triggered by data streams. Each data record from the `stream_source` table triggers a stream processing procedure, and a computing result is exported. The test file has three data records. After each data record reaches Realtime Compute, a stream processing procedure is triggered. Therefore, a total of three data records are displayed on the screen. The computational logic is described as follows:

- The first data record (aliyun) reaches Realtime Compute. This is the first time that the system has detected the word "aliyun." Therefore, the computing result is `<aliyun, 1>`, which is displayed on the screen.
- The second data record (aliyun) reaches Realtime Compute. The system detects an existing record of `<aliyun, 1>`, and adds 1 to the value. Therefore, the computing result is `<aliyun, 2>`, which is displayed on the screen.
- The third data record (aliyun) reaches Realtime Compute. The system detects an existing record of `<aliyun, 2>`, and adds 1 to the value. Therefore, the computing result is `<aliyun, 3>`, which is displayed on the screen.

The third computing result `<aliyun, 3>` is considered as the final output of the debugging. Another sample of [test data](#) that includes different keywords is provided

for you to test the debugging feature. You can use different samples of test data and view the outputs of debugging.

### 3.2.3.4 Administration

After the SQL file has been verified, you can publish the SQL file for the job on the Administration page of Realtime Compute. Then, you can start the job. The job runs on a Realtime Compute cluster.

#### Procedure

1. On the Development page, click Publish. The Publish New Version dialog box appears.
2. In the Resource Configuration step, click Next.
3. In the Check step, click Next.
4. In the Publish File step, click Publish.
5. On the Administration page, view the published word count job.
6. Click Start in the Actions column of the word count job. The Start dialog box appears.
7. Specify Start Time of Reading Data and click OK. Then, the job runs on a Realtime Compute cluster.

#### Result

After the job is started, click the job name. The Overview page appears.

#### FAQ

**Q:** Why does the word count job have no input or output while it is running on the distributed compute clusters of Realtime Compute?

**A:** When you created the `my_source` and `my_result` tables, you did not specify the data storage type of the referenced data source. In this scenario, the source table is considered to be a random table of strings or digits, and the result table is considered to be discarded data.

## 3.2.4 Big screen service for the Tmall Double Eleven Global Shopping Festival

### 3.2.4.1 Overview

During the Double 11 Shopping Festival, a big screen shows the total sales volume of Alibaba Group in real time. The big screen service is a highlight for the shopping

festival. Stream processing for the big screen service was previously based on Apache Storm that is an open-source distributed real-time computation system. The Storm-based development process took around one month. The application of Flink SQL shortened the development process of the big screen service to three days. The underlying layer of Realtime Compute removes the Apache Storm modules that are designed for execution optimization and troubleshooting. This enables a higher processing efficiency for Realtime Compute jobs.

### 3.2.4.2 Scenario description

The streaming data input for the Tmall big screen service is the transaction data from the Tmall platform. The incoming transaction data is organized based on a two-dimensional table: `tmall_trade_detail`.

Field	Type	Description
tid	BIGINT	The order ID.
buyer_uid	BIGINT	The buyer ID.
seller_uid	BIGINT	The seller ID.
gmtdate	TIMESTAMP	The time when the order is completed.
payment	DOUBLE	The order amount.

Realtime Compute calculates two metrics based on the preceding table: the total number of orders and the total order amount up to the current time. The two metrics are written to an online RDS system and displayed on a big screen in real time. The online RDS system is used to store the result table: `tmall_trade_state`.

Field	Type	Description
gmtdate	VARCHAR(16)	The date when the order is completed.
trade_count	BIGINT	The total number of orders.
trade_sum	DOUBLE	The total order amount.

The following topics describe how to build an end-to-end solution for the Tmall big screen service in around 10 minutes.



### 3.2.4.3 Preparations

Before editing Flink SQL statements for a Realtime Compute job, you must register data storage resources for data input and output. This topic uses DataHub as an example to describe how to register data storage resources.

Create a DataHub topic

1. Log on to the DataHub console. For more information, see the relevant documentation.
2. Click View to view the project information.
3. Click Create Topic.
4. Create the topic based on the structure of the `small_trade_state` table as described in the "Scenario description" section.

After performing these steps, you can move on to edit Flink SQL statements for the Realtime Compute job.

Upload data to DataHub

Log on to the DataHub console. Click Data Collection in the left-side navigation pane and then click Upload File. Then, click the DataHub topic that you have created and click Select File to upload data. To test the feature more easily, you can use the [data during the Double 11 Shopping Festival](#) for testing. You can download the data and then upload it to the DataHub topic for data collection.



**Note:**

This method is rarely used to collect data in actual use. For more information about how to use the data collection tool, see the topics about data collection in the user guide.

### 3.2.4.4 Register a data store

The data storage feature of Realtime Compute allows you to easily add DataHub topics, create tables, and reference data sources. To register a data store, follow these steps:

#### Procedure

1. [Log on to the Realtime Compute console](#).
2. Click Development in the top navigation bar.
3. On the Development page, click Data Storage in the left-side navigation pane.

4. Select DataHub Data Storage.
5. Click Registration and Connection on the top.
6. Register a DataHub project in Realtime Compute. For more information about parameter settings, see [Register a DataHub project](#).

If you use a MySQL RDS table to store the result data, you need to register RDS data storage resources in Realtime Compute. For more information, see [Register an RDS instance](#).

### 3.2.4.5 Development

After the data has been collected to Realtime Compute, you can continue to edit Flink SQL statements.

1. Create a reference to the source.

To create references to the DataHub source table and RDS result table, click **Data Storage** in the left-side navigation pane of the Development page in the Realtime Compute console, and perform the following operations:

- Find the target DataHub topic, and click **Reference as Source Table**. Realtime Compute automatically parses the schema of the topic and adds the corresponding SQL statements to the Development page.
- Find the target RDS table, and click **Reference as Result Table**. Realtime Compute automatically parses the schema of the table and adds the corresponding SQL statements to the Development page.

2. Edit Flink SQL statements.

If you have created the DataHub topic and RDS table as described in the previous topics, the Flink SQL code for the `tmall_d11` job can be executed directly.

Otherwise, change the names of the DataHub topic and RDS table based on the topic and table that you have created. The sample code is as follows:

```
replace into tmall_trade_state
select
 from_unixtime(FLOOR(tmall_trade_detail.gmtime/1000), 'yyyy-MM-dd') as gmt_date,
 count(tid) as trade_count,
 sum(payment) as trade_sum
from
 tmall_trade_detail
group by
 from_unixtime(FLOOR(tmall_trade_detail.gmtime/1000), 'yyyy-MM-dd');
```



**Note:**

You can modify the information about tables and fields as required.

**3. Debug the Flink SQL code.**

The *data during the Double 11 Shopping Festival* is available for testing. To debug the code, download the test data and upload the data on the Development page for debugging.

**4. Publish the SQL file for the tmall\_d11 job.**

After the computational logic has been verified in the debugging phase, click **Publish** on the Development page to publish the SQL file for the tmall\_d11 job. Then, you can view the tmall\_d11 job on the Administration page of the Realtime Compute console, and manage the job in the production environment, such as starting the job.

### 3.2.4.6 Administration

On the Administration page of the Realtime Compute console, select `tmall_d11` and click **Start**.



**Note:**

After you click **Start**, a dialog box appears. You must specify the start time for reading data from the source data storage system in the dialog box.

The specified time must be earlier than the time when you uploaded the data to DataHub during the data collection phase. In this example, the current time is 13:50 and the data was uploaded 10 minutes ago. Therefore, the start time is set to 13:00.

**Start**

**Start Settings**

Start Time for Reading Data: 09/08/2016, 13:00:00

The time specified in the WITH clause has a higher priority than the time specified in this dialog box.

**Auto Upgrade**

Enable Auto Upgrade: ☒

Upgrade Time: From 00:27 to 00:30 Every Day

Offset: Start at 0 00:00 Days before Upgrade

OK Cancel

The job is then scheduled by the clusters of Realtime Compute. After the job is started, its status changes to Running, and a green dot is displayed next to it.

You can check the result data in the RDS database after the job runs properly. In the result table, five transactions and a turnover of CNY 500 are displayed, which is consistent with the source data specified for testing. In this way, you can perform an end-to-end verification of the SQL code.

### 3.3 Project management

This topic describes how to create and search for a project.

Create a project

You can use an administrator account to log on to the Realtime Compute console and create and manage projects.

1. Enter the logon URL, such as `https://xxxx/#/admin`, in the address bar of a browser and then press Enter.
2. In the left-side navigation pane, click Project Management. The Projects page appears.
3. Click Create Project in the upper-right corner.

#### 4. Configure the project.

Table 3-5: Field description

Field	Description
Project Name	The name of the project.
Project Type	By default, Blink Project is selected.
Cluster	The cluster on which the jobs in the project run.
Administrators	The project administrators. Only project administrators can manage jobs in the project.
Description	The description of the project.
GPUs	The number of GPUs that are used by the project.
Slots	The number of compute units (CUs) that are used by the project. Currently, one CU is assigned with one CPU core and 4 GB memory.
Alert Methods	The methods by which the alerts are sent when an exception occurs on running jobs. You can receive alerts through short message services (SMSs) or TradeManager messages.
File Types	The supported file types.
Storage Types	The supported storage types.
Max Data Stores	The maximum number of data stores that can be registered in Realtime Compute. Typically, you can keep the default value.
Max File Versions	The maximum number of code versions for the SQL file. Typically, you can keep the default value.
Max Folders	The maximum number of folders that can be created in the project. Typically, you can keep the default value.
Max Folder Levels	The maximum number of folder levels in the project. Typically, you can keep the default value.
Max Files	The maximum number of SQL files that can be created for jobs in the project. Typically, you can keep the default value.
Max Resources	The maximum number of JAR files or DICTIONARY resources that can be uploaded. Typically, you can keep the default value.

Field	Description
Max Referenced Resources	The maximum number of JAR files and DICTIONARY resources that can be referenced. Typically, you can keep the default value.
Monitoring and Alerting	Specifies whether to enable the monitoring and alerting feature. Typically, you can keep the default setting.
Data Collection	Specifies whether to collect data while a job is running. Typically, you can keep the default setting.
Data Display	Specifies whether to display data. Typically, you can keep the default setting.
Data Storage	Specifies whether to enable the feature of data store registration. This feature is enabled by default. Typically, you can keep the default setting.
Engine	Specifies whether to display the engine. Typically, you can keep the default setting.
Online Logs	Specifies whether to record the running logs of jobs. This feature is enabled by default. Typically, you can keep the default setting.
Resource Management	Specifies whether resources such as JAR files can be uploaded. This feature is enabled by default. Typically, you can keep the default setting.

5. Click OK.

Search for a project

To search for a project, enter the keywords or full name of a project in the search bar on the Projects page.

## 3.4 Data storage

### 3.4.1 Overview

This chapter describes data storage systems supported by Realtime Compute.

### 3.4.2 Overview

#### 3.4.2.1 Overview

To facilitate data storage management, you can register data storage resources on the Realtime Compute development platform. This enables you to leverage the advantages of the one-stop Realtime Compute service. In Realtime Compute, you

can manage multiple data storage systems, such as ApsaraDB for RDS, and Table Store. With this one-stop management service, you no longer have to navigate across multiple management consoles of different storage systems.

### 3.4.2.2 Types

Realtime Compute supports both streaming data storage and static data storage.

#### Streaming data storage

Streaming data storage systems provide inputs and outputs for downstream Realtime Compute jobs.

Table 3-6: Streaming data storage

Storage system	Input	Output
DataHub	Supported	Supported
Log Service	Supported	Supported
MQ	Supported	Supported

#### Static data storage

Static data storage systems provide outputs for downstream Realtime Compute jobs and allow you to perform association queries.

Table 3-7: Static data storage

Storage system	Dimension table	Output
ApsaraDB for RDS	Supported	Supported
Table Store	Supported	Supported

### 3.4.2.3 Scenarios

The topic describes the scenarios where external data storage systems are used.



**Note:**

If you must use resources managed by another account, you can write DDL statements where the AccessKey ID and AccessKey Secret of the account are specified.

- Register data resources

To register data resources, follow these steps:

1. [Log on to the Realtime Compute console.](#)
2. Click Development in the top navigation bar. On the page that appears, click Data Storage in the left-side navigation pane and then click +.



**Note:**

With the data storage feature enabled, you can only register data resources owned by your own level-1 department.

- Preview data

Realtime Compute provides the data preview feature for each registered data store. To preview the data, click Data Storage and double-click one of the folders on the left side of the page. The following uses DataHub as an example to describe the data preview feature.

1. Log on to the Realtime Compute console. Choose Data Storage > DataHub Data Storage.
2. Select a project and double-click a topic to view the details.

- Automatically generate DDL statements

You must declare tables from external data storage systems before you can reference these tables in Realtime Compute. The following is an example of how to reference a source table that includes streaming data inputs.

```
CREATE TABLE in_stream(a varchar, b varchar, c timeStamp) with (type
='datahub', endPoint='http://dh-cn-hangzhou.aliyuncs.com', project='
blink_test', topic='ip_count02', accessId='LTAIYtafPsXXXX', accessKey
='gUqyVwfkK2vfJI7jF90QXXXXX');
```

The field names in the referenced table must be the same as those in the DataHub source table. You can change the field data types in the code if necessary to ensure that Realtime Compute can recognize the data. Realtime Compute offers the feature of automatic DDL generation. The following is an example of how to use this feature.

To reference a source table, log on to the Realtime Compute console and open the target SQL file on the Development page. Click Data Storage, select a table for reference, and then click Reference as Source Table. Then, the required DDL statements are displayed on the current page.



### Reference data storage resources owned by another level-1 department

Currently, you can only register and use data storage resources that are owned by your level-1 department. To use data storage resources that are owned by another level-1 department, use DDL statements to create a reference to these data storage resources. For example, if a user from department A needs to use data storage resources that are owned by department B, the user can enter the following DDL statements:

```
CREATE TABLE in_stream(a varchar, b varchar, c timeStamp) with (type
='datahub', endPoint='http://dh-cn-hangzhou.aliyuncs.com', project='
blink_test', topic='ip_count02', accessId='AccessKey ID authorized by
department B ', accessKey='AccessKey Secret authorized by department B
');
```

### 3.4.3 Register a DataHub project



DataHub is a real-time data distribution platform that is designed to process streaming data. It provides a channel for the Apsara Stack DTplus platform to process big data. DataHub can work with multiple Apsara Stack services to build an end-to-end data processing platform. Realtime Compute typically uses DataHub to store source and result tables for streaming data.

Register a DataHub project

1. [Log on to the Realtime Compute console.](#)
2. Click Development in the top navigation bar.
3. In the left-side navigation pane, click Data Storage.
4. Right-click DataHub Data Storage and then select Register Data Store to register a DataHub project in Realtime Compute.

Table 3-8: Field description

Field	Description
Test Connection	A network connectivity test is automatically performed on storage systems that support the feature of registering data stores. To test the connection between Realtime Compute and data storage systems that do not support the feature of registering data stores, turn on Test Connection.
Storage Type	By default, DataHub is selected.

Field	Description
Endpoint	<p>Specifies the endpoint of DataHub. The endpoint of DataHub varies by region. For more information about the endpoint, contact your administrator.</p> <div> <b>Note:</b> To specify this field for Apsara Stack, contact your Apsara Stack administrator about the DataHub endpoint.</div>
Project	<p>The name of the DataHub project.</p> <div> <b>Note:</b> You can only register DataHub projects that are owned by your level-1 department. For example, if DataHub project A is owned by department A, a user from department B cannot register DataHub project A in Realtime Compute.</div>
AccessId	The AccessKey ID of the current account.
AccessKey	The AccessKey Secret of the current account. It enables Realtime Compute to access the DataHub project.

#### Scenarios

DataHub is a streaming data storage system. Therefore, it can be used to store source and result tables, but not dimension tables for Realtime Compute.

#### FAQs

**Q: Why does registration fail?**

**A:** Realtime Compute uses a storage software development kit (SDK) to access different data storage systems. The Data Storage tab in the Realtime Compute console only helps you manage the data from different data storage systems. You can perform the following operations to troubleshoot registration errors:

- Check whether you have created the DataHub project and have permission to access the project. To perform the check, log on to the DataHub console, and check whether you can access the project.
- Check whether you are the project owner. You can only register DataHub projects that are owned by your level-1 department. For example, if DataHub project A is owned by department A, a user from department B cannot register DataHub project A in Realtime Compute.

- Check whether you have entered the correct DataHub endpoint and project name.
- Check whether you have specified a classic network endpoint for the Endpoint field. Currently, VPC endpoints are not supported by Realtime Compute.
- Check whether you have already registered the DataHub project. Realtime Compute provides a registration check mechanism that prevents duplicate registration.

**Q:** Why is only time-based sampling supported?

**A:** DataHub stores streaming data, and you can only specify time parameters for DataHub in the APIs. Therefore, Realtime Compute only supports time-based sampling.

### 3.4.4 Register a Log Service project



Log Service provides an end-to-end solution for log management, which allows you to easily collect, subscribe to, ship, and query large amounts of log data. If you use Log Service to manage Elastic Compute Service (ECS) logs, Realtime Compute can integrate with Log Service to process ECS logs. This eliminates the need to transfer data between these systems.

Register a Log Service project

1. [Log on to the Realtime Compute console](#).
2. Click Development in the top navigation bar.
3. In the left-side navigation pane, click Data Storage.
4. Right-click Log Service Data Storage and then select Register Data Store to register a Log Service project in Realtime Compute.

Table 3-9: Field description

Field	Description
Test Connection	A network connectivity test is automatically performed on storage systems that support the feature of registering data stores. To test the connection between Realtime Compute and data storage systems that do not support the feature of registering data stores, turn on Test Connection.
Storage Type	By default, Log Service is selected.

Field	Description
Endpoint	<p>The endpoint of Log Service. The endpoint of Log Service varies by region.</p> <div> <b>Note:</b> For more information about the endpoint for Log Service, contact the Apsara Stack system administrator.</div>
Project	<p>The name of the Log Service project.</p> <div> <b>Note:</b> You can only register Log Service projects that are owned by your level-1 department. For example, if Log Service project A is owned by department A, a user from department B cannot register Log Service project A in Realtime Compute.</div>
AccessId	The AccessKey ID of the current account.
AccessKey	The AccessKey Secret of the current account. It enables Realtime Compute to access the Log Service project.

#### Scenarios

Log Service is a streaming data storage system. Therefore, it can be used to store source and result tables, but not dimension tables for Realtime Compute.

#### FAQs

• **Q: Why does registration fail?**

**A:** Realtime Compute uses a storage software development kit (SDK) to access different data storage systems. The Data Storage tab in the Realtime Compute console only helps you manage the data from different data storage systems. You can perform the following operations to troubleshoot registration errors:

- Check whether you have created the Log Service project and have permission to access the project. To perform the check, log on to the Log Service console, and check whether you can access the project.
- Check whether you are the project owner. You can only register Log Service projects that are owned by your level-1 department. For example, if Log

Service project A is owned by department A, a user from department B cannot register Log Service project A in Realtime Compute.

- Check whether you have entered the correct Log Service endpoint and project name.



**Note:**

The endpoint must start with **http** and cannot end with a forward slash (/).

For example, `http://cn-hangzhou.log.aliyuncs.com` is correct, but `http://cn-hangzhou.log.aliyuncs.com/` is incorrect.

- Check whether you have already registered the Log Service project. Realtime Compute provides a registration check mechanism that prevents duplicate registration.
- Q: Why is only time-based sampling supported?

A: Log Service stores streaming data, and you can only specify time parameters for Log Service in the APIs. Therefore, Realtime Compute only supports time-based sampling.



**Note:**

To use the search feature of Log Service, log on to the Log Service console.

### 3.4.5 Register a Table Store instance

Table Store is a NoSQL database service built based on the Apsara system. Table Store allows you to store and access large amounts of structured data in real time. Realtime Compute needs to access data with an extremely short delay, but is not highly demanding for relational algebra. This makes it suitable to use Table Store to store dimension tables and result tables.

Register a Table Store instance

1. [Log on to the Realtime Compute console.](#)
2. Click **Development** in the top navigation bar.
3. In the left-side navigation pane, click **Data Storage**.

4. Right-click Table Store Data Storage and then select Register Data Store to register a Table Store instance in Realtime Compute.

Table 3-10: Field description

Field	Description
Test Connection	A network connectivity test is automatically performed on storage systems that support the feature of registering data stores. To test the connection between Realtime Compute and data storage systems that do not support the feature of registering data stores, turn on Test Connection.
Storage Type	By default, Table Store is selected.
Endpoint	The endpoint of Table Store. Log on to the Table Store console to view the Table Store endpoint. You must enter the endpoint for the internal network. To enable Realtime Compute to access Table Store instances that are connected to VPC networks, change the network type of Table Store instances to allow access from any network.
Instance Name	The name of the Table Store instance.
AccessId	The AccessKey ID of the current account.
AccessKey	The AccessKey Secret of the current account. It enables Realtime Compute to access the Table Store instance.

#### Scenarios

Table Store features massive data storage and low access delays, which makes it suitable to store dimension tables and result tables for Realtime Compute.

Enable Realtime Compute to access a Table Store instance that is connected to a VPC network

You can access Table Store in one of the following modes:

- Allow access from any network
- Allow VPC access only
- Allow console or VPC access only

To enable Realtime Compute to access a Table Store instance that is connected to a VPC network, follow these steps:

1. Log on to the Table Store console. For more information, see the relevant documentation.
2. Click the name of the target instance. The Instance Details page appears.

3. Click the Edit icon to change the network type of the instance.
4. In the dialog box that appears, change the network type to allow access from any network.

### 3.4.6 Register an RDS instance

This topic describes how to register and use an RDS instance in Realtime Compute.

#### RDS overview

ApsaraDB for RDS (RDS for short) is a stable, reliable, and scalable online database service. Based on the Apsara system and full solid-state drive (SSD) storage, RDS supports a wide range of engines, such as MySQL, PostgreSQL, and Postgres Plus Advanced Server (PPAS, which is highly compatible with Oracle). Currently, Realtime Compute supports the following RDS engines: MySQL, and PostgreSQL.

Due to the limits of relational models, the performance of RDS is not as good as that of Table Store in high concurrency scenarios where large amounts of data needs to be processed. Therefore, RDS is usually used to store result tables for Realtime Compute. In low concurrency scenarios where only small batches of data need to be processed, RDS can be used to store dimension tables for Realtime Compute.



#### Note:

Realtime Compute uses relational databases, such as MySQL, to store result data. The relational databases use Distributed Relational Database Service (DRDS) and RDS connectors. When Realtime Compute frequently writes data to a table or resource file, a deadlock may occur. In scenarios that require high queries per second (QPS), high transactions per second (TPS), or highly concurrent write operations, we do not recommend that you use DRDS or RDS to store result tables. To prevent deadlocks, we recommend that you use Table Store to store result tables.

#### Register an RDS instance

1. [Log on to the Realtime Compute console](#).
2. Click Development in the top navigation bar.
3. In the left-side navigation pane, click Data Storage.

4. Right-click RDS Data Storage and then select Register Data Store to register an RDS instance in Realtime Compute.

Table 3-11: Field description

Field	Description
Test Connection	A network connectivity test is automatically performed on storage systems that support the feature of registering data stores. To test the connection between Realtime Compute and data storage systems that do not support the feature of registering data stores, turn on Test Connection.
Storage Type	By default, RDS is selected.
URL	The connection URL of the database.
DBName	<p>The name of the RDS database to be accessed by Realtime Compute. Note that this field specifies the database name instead of the name of the RDS instance.</p> <p>RDS uses a whitelist mechanism to ensure system security. The IP addresses of the Realtime Compute console and TaskManagers must be added to an RDS whitelist group. Otherwise, Realtime Compute may fail to connect to RDS. For more information, see <a href="#">Specify whitelist settings</a>.</p>
Username	The username that you use to log on to the database.
Password	The password that you use to log on to the database.
Engine	The type of the RDS database.

Reference an RDS table as the result table

After you register an RDS database, double-click the RDS Data Storage folder, click the RDS table that you want to reference as a result table, and then click Reference as Result Table.

After you click the Reference as Result Table button, Realtime Compute automatically generates corresponding DDL statements on the current page.

If the VPC authorization error message appears, analyzes, and rectify the fault as follows.



The cause of this fault is that a VPC rather than a classic network was selected when you created the RDS instance. To rectify this fault, follow these steps:

1. Move the pointer over the Administrator icon.
2. Click System Settings.
3. In the left-side navigation pane, click VPC Access Authorization.
4. Click Add Authorization. The Authorize Realtime Compute to Access VPC page appears.

Table 3-12: Field description

Field	Description
Name	The name of the VPC.
Region	The region where RDS is located.
VPC ID	The ID of the VPC.
Instance ID	The ID of the RDS database instance. You can log on to the RDS console and view the instance ID.
Instance Port	The access port for the instance. To view the internal network port number, log on to the RDS console, click the icon in the Actions column, and then click View Details. On the page that appears, view the internal network port number in the Internal Network Connection Information section.

5. Register an RDS instance. You must specify all required fields during the registration.

You can only register data storage resources that are owned by your level-1 department. For example, if RDS instance A is owned by department A, a user from department B cannot register RDS instance A in the Realtime Compute. To use instance A in a stream processing job, the user from department B must use Flink SQL code to create a reference to the RDS instance.



**Note:**

If you need to use RDS storage resources of a level-1 department account, we do not recommend that you use Flink SQL code to create a reference to these resources.

The user from department B must also specify the following parameters in the **WITH** clause based on the information of instance A: url, userName, password, and tableName.

To use RDS storage resources through Flink SQL code, the user from department B must specify whitelist settings.

#### Specify whitelist settings

Some data storage systems use a whitelist mechanism to ensure high-level security . Only whitelisted IP addresses are allowed to access data stores. This mechanism prevents other Apsara Stack services from writing data to data stores. For example , in the RDS data storage system, a newly created database denies all access requests. You must specify whitelist settings to allow the whitelisted IP addresses to access the database. When Realtime Compute uses RDS to store a dimension table or result table, Realtime Compute needs to frequently read and write the RDS database. Therefore, the IP addresses of the Realtime Compute console and TaskManagers must be added to an RDS whitelist group to establish a connection with RDS.

You can access RDS from both external and internal networks. To enable Realtime Compute to access RDS, you must add the network segments of Realtime Compute to an RDS whitelist group.

To add the network segments of Realtime Compute to an RDS whitelist group, follow these steps:

1. Log on to the RDS console, and click the target instance name to view the details.
2. In the left-side navigation pane, choose **Access Control > Whitelist Settings**.
3. On the page that appears, click the **Edit** icon for the default whitelist group.
4. In the dialog box for modifying the whitelist group, remove 0.0.0.0/0 from the group, and add network segments or IP addresses. Separate network segments or IP addresses with commas (,).

You can also add a custom whitelist group. To add a custom whitelist group, click the **Delete** icon for the default whitelist group, click **Create Whitelist Group**, and then create a custom whitelist group.

1. Click **Create Whitelist Group**. A page appears for you to create a custom whitelist group.



**Note:**

If the whitelist group contains only 0.0.0.0/0, you can access the instance from any IP address.

2. Enter a group name, add IP addresses or network segments, and then click **OK**.
  - **Group Name:** A group name must be 2 to 32 characters in length and can contain lowercase letters, digits, and underscores (\_). It must start with a lowercase letter and end with a letter or digit. The default group cannot be modified or deleted.
  - **IP Addresses:** Enter the whitelisted network segments or IP addresses. Separate the network segments or IP addresses with commas (,).

## FAQ

**Q:** what can I do if I fail to register an RDS instance with error log `The driver has not received any packets from the server?`

**A:** Please add the IP address of your region into the RDS whitelist, for more information please refer [Specify whitelist settings](#).

## 3.5 Data development

### 3.5.1 Create a job

This topic describes how to create a Realtime Compute job.

#### Procedure

1. [Log on to the Realtime Compute console](#).
2. Click **Development Platform**.
3. Click **Development** in the top navigation bar.
4. Click **Create File** in the toolbar.

5. In the Create File dialog box, specify the required fields.

Field	Description
File Name	The name of the file. The specified name must be 3 to 64 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.
File Type	The type of the file. Valid values: FLINK_STREAM/SQL and FLINK_STREAM/DATASTREAM.
Storage Path	The folder of the file. You can click the icon on the right side of an existing folder and create a subfolder.

6. Click OK.

## 3.5.2 Development

### 3.5.2.1 SQL code assistance

The development platform of Realtime Compute offers a complete set of SQL tools in the integrated development environment (IDE). These tools provide the following features to help you with Flink SQL-based development:

- Syntax check

On the Development page of Realtime Compute, the revised script is automatically saved. When the script is saved, an SQL syntax check is automatically performed. If a syntax error is detected, the Development page shows the row and column where the error is located, and the cause of the error.

- Intelligent code completion

When you enter SQL statements on the Development page of Realtime Compute, auto completion popups about keywords, built-in functions, tables, or fields are automatically displayed.

- Syntax highlighting

Flink SQL keywords are highlighted in different colors to differentiate data structures.

### 3.5.2.2 SQL code version management

Realtime Compute provides key features that help you complete development tasks, such as coding assistance and code version management. A code version is generated each time you publish an SQL file for a job. The code management

feature allows you to track code changes and roll back to an earlier version if necessary.

- **Manage code versions**

A snapshot of a code version is created after you submit an SQL file for publishing a job every time. This allows you to track code changes. To view the versions of a job, click **Development** in the top navigation bar of the Realtime Compute console and then click **Properties** on the right side of the **Development** page.

- **Delete code versions**

A snapshot of a code version is created after you submit an SQL file for publishing a job every time. This allows you to track code changes. The maximum number of code versions has been specified. If you use Apsara Stack, a maximum of 20 code versions can be published. To find out the maximum number of code versions in other running environments, contact the system administrator. If the number of code versions exceeds the specified upper limit, a prompt is displayed to alert you to delete some earlier versions.

To delete expired and unnecessary code versions, click **Development** in the top navigation bar of the Realtime Compute console, click **Properties** on the right side of the **Development** page, and then click **Delete** in the **Actions** column.

### 3.5.2.3 Data storage management

On the **Development** page, you can easily and effectively manage data storage. For example, you can register data sources from multiple data storage systems on the **Development** page.

- **Data preview**

The **Development** page of Realtime Compute allows you to preview the data of multiple storage types. Data preview helps you efficiently analyze upstream and downstream data, identify key business logic, and complete development tasks.

- **Auto DDL generation**

In most cases, the DDL statements for data storage systems are manually translated to the DDL statements for stream processing. Therefore, the DDL generation process includes a large number of repetitive tasks. Realtime Compute provides an auto DDL generation feature. This feature simplifies

the way that you edit SQL statements for stream processing jobs, reduces the possibility of encountering errors when manually entering SQL statements, and also improves efficiency.

### 3.5.3 Debug the code

The Realtime Compute development platform provides a simulated running environment where you can customize uploaded data, simulate operations, and check outputs.

After you have developed the Flink SQL code that implements the computational logic, follow these steps to debug the code:

1. [Log on to the Realtime Compute console](#).
2. Click **Development** in the top navigation bar.
3. Open the target file by double-clicking the file or right-clicking the file and selecting **View File**, and then click **Syntax Check** in the toolbar.



**Note:**

You can use the syntax check feature to check whether there are syntax errors in the SQL file. Error messages are prompted for any syntax errors.

4. Click **Debug** in the toolbar. You can debug your file on the **Debug File** page.

The test data for code debugging can be acquired using either of the following two methods:

- Upload local data.
  - a. Click **Download Template**.
  - b. Prepare test data in a file based on the template.
  - c. Click **Upload File**. After the file is uploaded, you can view the uploaded data on the **Data Preview** page.



**Note:**

**By default, a comma (,) is used as the separator in files for debugging.  
For more information about custom separators, see [Separator in files for debugging](#).**

- **Sample online data.**
  - a. **Click Random Online Data Sampling or Sequential Online Data Sampling.**
  - b. **View the sample data on the Data Preview page.**
- 5. **Click OK to start debugging.**
- 6. **In the dialog box that appears, view the debugging result.**

**The debugging feature of Realtime Compute provides the following functions:**

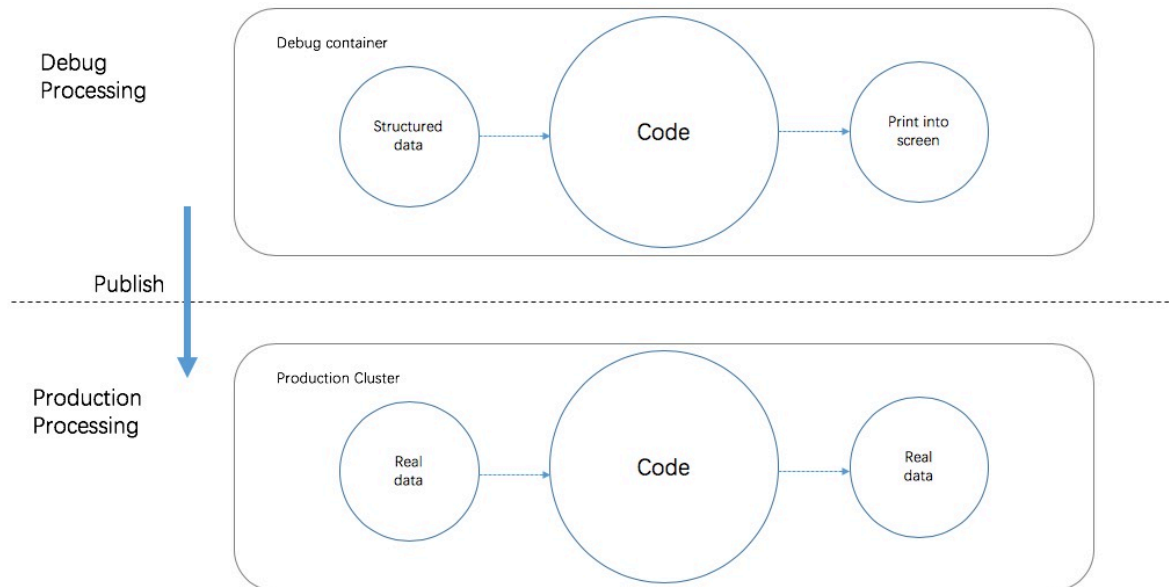
- **Enables isolation between debugging and production environments.**

**In the debugging environment, the Flink SQL code runs in a separate container, and the result data of computations is only displayed on the Development page. In this way, the debugging does not affect online running jobs and data storage systems in the production environment.**

**In the debugging phase, result data is not written to external data storage systems. In the production environment, a running failure may occur due to format errors while writing result data to the target data storage systems. Such failures cannot be identified or prevented in the debugging phase, and can be detected only while jobs are running. For example, if your result data is written to a result table stored in RDS and the length of some character strings exceeds the maximum length allowed for an RDS table, a running failure will occur in the production environment. We are working on support for writing result data to external data storage systems in the production environment. This helps you**

effectively simulate the production environment and resolve more issues in the debugging phase.

Figure 3-1: Debug



- Supports the customization of test data.

In the debugging environment, Realtime Compute does not read data from the source data storage systems, such as source tables in DataHub and dimension tables in RDS. You must create a set of test data and upload the test data on the Development page.

To make the debugging feature easy to use, Realtime Compute provides a template of test data for each type of job. You can download the template and enter your test data.



**Note:**

We recommend that you use the templates to prevent errors.

- Specifies a separator.

By default, a comma (,) is used as the separator in files for debugging. An example of a file for debugging is described as follows:

```
id,name,age
1,alicloud,13
```



```
2,stream,1
```

If the separator is not specified, a comma (,) is used to separate fields. If you need to use a JSON string as the field data and it contains commas (,), you must specify another character as the separator.

**Note:**

Realtime Compute allows you to specify a letter as the separator, but not a multi-character string, such as aaa.

```
id|name|age
1|alicloud|13
2|stream|1
```

In this example, specify the `debug.input.delimiter` parameter as follows: `debug.input.delimiter=|`.

### 3.5.4 Publish the SQL file for a job

After you have created and debugged an SQL file for a job, you can publish the SQL file, and manage the job in the production environment.

#### Procedure

1. [Log on to the Realtime Compute console](#).
2. Click **Development** in the top navigation bar.
3. Open the target file by double-clicking the file or right-clicking the file and selecting **View File**, and then click **Publish** in the toolbar.
4. In the dialog box that appears, select **Automatic CU Configuration**. If you are performing automatic configuration for the first time, use the default number of CUs. Click **Next**.
5. Verify the data and then click **Next**.
6. Click **Publish**.
7. On the **Administration** page, click **Start** to start the job.

## 4 Quick BI

### 4.1 What is Quick BI?

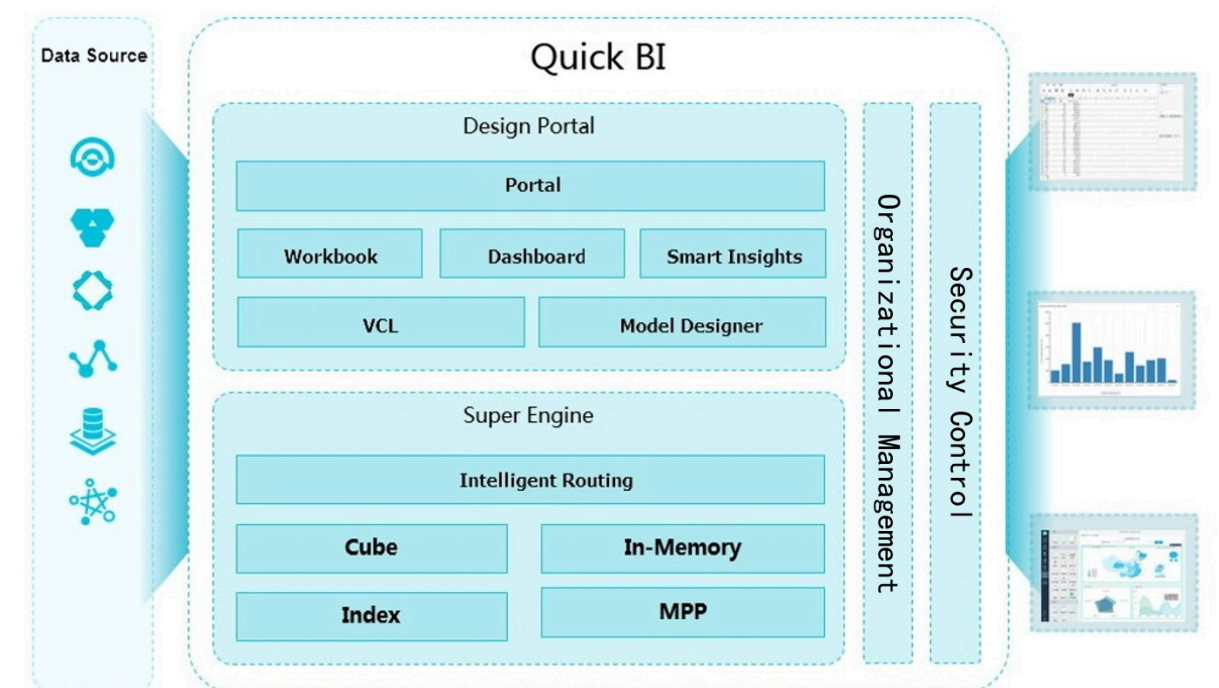
This topic describes features of Quick BI.

Quick BI is a flexible and lightweight self-service BI tool based on cloud computing . It supports a wide range of data sources, including MaxCompute (formerly known as ODPS), ApsaraDB RDS for MySQL, and other data sources. You can also connect VPC-connected Quick BI to user-created MySQL and SQL Server databases deployed on Elastic Compute Service (ECS) instances.

Quick BI analyzes large amounts of data in real time and returns results within seconds. You do not need to preprocess the data. Quick BI analyzes terabytes of incremental data on a daily basis.

With an intelligent data modeling tool and a variety of visual chart tools, Quick BI significantly reduces data acquisition costs and makes it easier for you to use Quick BI features. This allows you to easily complete data analysis, self-service data acquisition, business data query, and report making.

Figure 4-1: Architecture



## 4.2 Log on to the Quick BI console

This topic provides an example of how to log on to the Quick BI console.

### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

### Context

If you are using a RAM account, log on to the Quick BI console through the domain name of the Quick BI cluster. If you are using an Alibaba Cloud account, log on to the Quick BI console by following these steps:

### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username `super`. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. Choose Big Data > Quick BI to go to the Quick BI logon page.

## 6. Select a department and click Quick BI or Quick BI Console.

- You can click Quick BI to go to the product page.
- You can click Quick BI Console to go to the Settings page. You can manage organization members and workspaces on this page.



### Note:

If you choose this method, the Alibaba Cloud account `quickbi_admin@aliyun.com` is used to log on to the Quick BI console no matter which department you select.

## 4.3 Data modeling

### 4.3.1 Overview

Steps of data modeling:

1. Create a data source.
2. Select a table from the data source to create a dataset.
3. Use custom SQL statements to create a dataset (optional).

### 4.3.2 Data sources

#### 4.3.2.1 Overview

Quick BI supports the following types of data sources:

- Cloud data sources, including MaxCompute, MySQL, AnalyticDB for PostgreSQL, PostgreSQL, and PPAS.
- User-created data sources, including MySQL, SQL Server, PostgreSQL, Oracle, Hive, Vertica, IBM DB2 LUW, SAP IQ (Sybase IQ), and SAP HANA.
- VPC-connected data sources



### Note:

Currently, you cannot view SQL Server data sources through views.

#### 4.3.2.2 Cloud data sources

##### 4.3.2.2.1 View whitelisted IP addresses

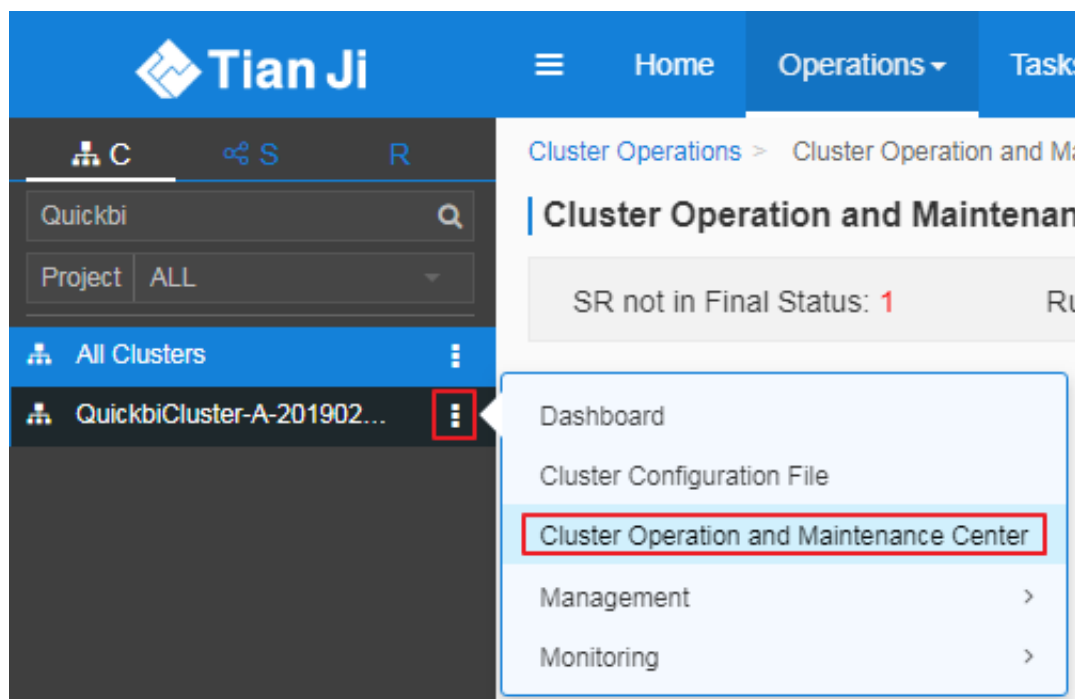
When you add a data source, you may have to add the IP addresses of machines in the Quick BI cluster to the ApsaraDB for RDS whitelist as required. This topic

describes how to view whitelisted IP addresses. For more information, contact Quick BI administrators.

### Procedure

1. On the homepage of Apsara Infrastructure Management Framework, enter the name of the Quick BI cluster in the search box to locate the Quick BI cluster.
2. Hover over the More icon of the Quick BI cluster and select Cluster Operation and Maintenance Center, as shown in [Cluster Operation and Maintenance Center](#).

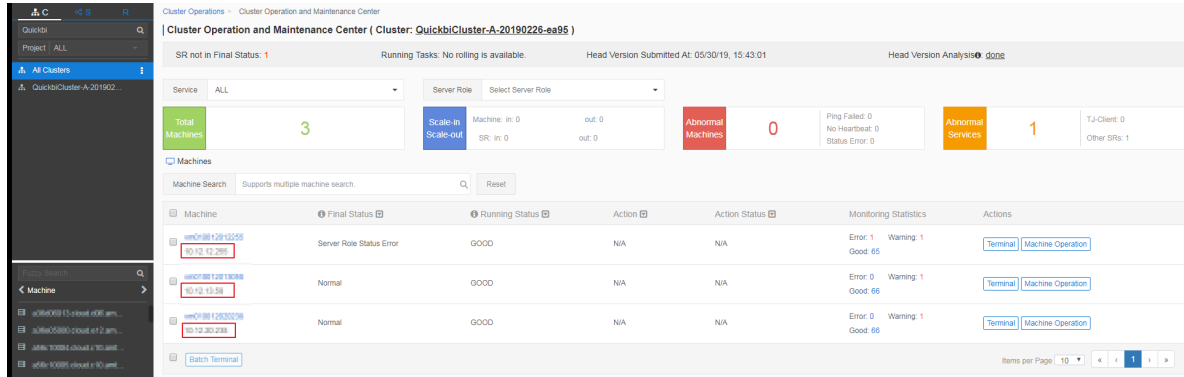
Figure 4-2: Cluster Operation and Maintenance Center



3. Before you add the IP addresses of machines to the ApsaraDB for RDS whitelist, change the last octet of all the machine IP addresses in the cluster to 0/24. For

example, if the IP address of a machine is 10.10.10.10, change it to 10.10.10.0/24, as shown in *Figure 4-3: Cluster Operation and Maintenance Center*.

Figure 4-3: Cluster Operation and Maintenance Center



4. Add the IP address to the ApsaraDB for RDS whitelist. For more information, see *Configure a whitelist in ApsaraDB for RDS User Guide*.

#### 4.3.2.2.2 MaxCompute

This topic describes how to add a cloud MaxCompute data source.

#### Procedure

1. *Add a data source.*

2. Click **MaxCompute** and a dialog box appears, as shown in [Figure 4-4: Add a MaxCompute data source](#).

Figure 4-4: Add a MaxCompute data source

**Add MaxCompute Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: http://service.odps.aliyun.com/api

\* Project Name: Enter a project name.

\* AccessKey ID: Enter the AccessKey ID.

\* AccessKey Secret: Enter the AccessKey Secret.

ⓘ Note: Latency may occur while synchronizing the data source.

[Close] [Test Connection] [Add]

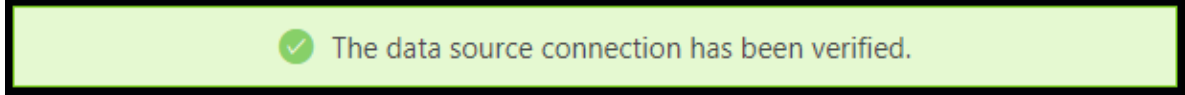
3. Enter the data source connection information in the dialog box.

Parameter	Description
Name	Specify a name for the data source.
Database Address	Use the default database address. Do not modify the address.
Project Name	Enter the name of the target MaxCompute project.
AccessKey ID	Enter the AccessKey ID of the account that purchased the data source instance. The AccessKey ID identifies a user.
AccessKey Secret	Enter the AccessKey Secret of the account that purchased the data source instance. The AccessKey Secret is used to encrypt the signature string on the client and decrypt the signature string on the server for authentication. Keep the AccessKey Secret confidential.

4. Click **Test Connection** to perform a data source connectivity test, as shown in

*Figure 4-5: Connectivity test.*

Figure 4-5: Connectivity test



5. After the connection is established, click **Add**.

After the data source is added, you are redirected to the **Data Sources** page.

Tables under the data source are listed on the right side of the page.

#### 4.3.2.2.3 MySQL

This topic describes how to add a cloud MySQL data source.

##### Context

When you add a MySQL data source, you must add the IP addresses of machines in the Quick BI cluster to the ApsaraDB for RDS whitelist. For more information about adding IP addresses to a whitelist, see *Configure a whitelist in ApsaraDB for RDS User Guide*.

##### Procedure

1. *Add a data source.*
2. Click **MySQL** and enter the data source connection information, as shown in

*Figure 4-6: Add a MySQL data source.*



**Note:**



**If you connect Quick BI to a VPC-connected SQL Server data source, select the VPC Data Source check box and set the parameters as required.**

Figure 4-6: Add a MySQL data source

**Add MySQL Database** [X]

\* Name:  Enter a database name to be displayed.

\* Database Address:  Enter a hostname or an IP address.

\* Port Number:  3306

\* Database:  Enter a database name.

\* Username:  Enter a username.

\* Password:  Enter the password.

VPC Data Source: ☒ ⓘ

\* AccessKey ID:  Enter the AccessKey ID.


\* AccessKey Secret:  Enter the AccessKey Secret.

\* Instance ID:  Enter the instance ID.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

[Close] [Test Connection] [Add]

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 3306.
Database	Enter the name of the database.
Username	Enter the username of the database.

Parameter	Description
Password	Enter the password of the database.  <b>Note:</b> If you do not know the username and password, contact your database administrator.
AccessKey ID	Enter the AccessKey ID of the account that purchased the data source instance.
AccessKey Secret	Enter the AccessKey Secret of the account that purchased the data source instance.
Instance ID	Enter the data source instance ID.

3. Click Test Connection to perform a data source connectivity test.

4. After the connection is established, click Add.

After a data source is added, you cannot add it again. If you attempt to add the same data source, an error message appears.

#### 4.3.2.2.4 SQL Server

This topic describes how to add a cloud SQL Server data source.

##### Context

When you add an SQL Server data source, you must add the IP addresses of machines in the Quick BI cluster to the ApsaraDB for RDS whitelist. For more information about adding IP addresses to a whitelist, see *Configure a whitelist in ApsaraDB for RDS User Guide*.

The procedure to configure an SQL Server data source is similar to configuring a MySQL data source. However, SQL Server data sources require an additional parameter: Schema.

##### Procedure

1. [Add a data source](#).
2. Click SQL Server and enter the data source connection information, as shown in [Figure 4-7: Add an SQL Server data source](#).



**Note:**

If you connect Quick BI to a VPC-connected SQL Server data source, select the VPC Data Source check box and set the parameters as required.

Figure 4-7: Add an SQL Server data source

**Add SQL Server Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 1433

\* Database: Enter a database name.

Schema: dbo

\* Username: Enter a username.

\* Password: Enter the password.

VPC Data Source: ☒ ⓘ

\* AccessKey ID: Enter the AccessKey ID.

\* AccessKey Secret: Enter the AccessKey Secret.

\* Instance ID: Enter the instance ID.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

[Close] [Test Connection] [Add]

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 1433.
Database	Enter the name of the database.

Parameter	Description
Schema	Enter the schema. The default schema is dbo.
Username	Enter the username of the database.
Password	Enter the password of the database.
AccessKey ID	Enter the AccessKey ID of the account that purchased the data source instance.
AccessKey Secret	Enter the AccessKey Secret of the account that purchased the data source instance.
Instance ID	Enter the data source instance ID.

3. Click Test Connection to perform a data source connectivity test.
4. After the connection is established, click Add.

#### 4.3.2.2.5 AnalyticDB

This topic describes how to add a cloud AnalyticDB data source.

##### Context

AnalyticDB is also named ADS.

##### Procedure

1. [Add a data source.](#)

**2. Click AnalyticDB and enter the data source connection information, as shown in**

*Figure 4-8: Add an AnalyticDB data source.*

Figure 4-8: Add an AnalyticDB data source

**Add AnalyticDB Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 3306

\* Database: Enter a database name.

\* AccessKey ID: Enter the AccessKey ID.

\* AccessKey Secret: Enter the AccessKey Secret.

[Close] [Test Connection] [Add]

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number.
Database	Enter the name of the database.
AccessKey ID	Enter the AccessKey ID of the account that purchased the data source instance. The AccessKey ID identifies a user.
AccessKey Secret	Enter the AccessKey Secret of the account that purchased the data source instance. The AccessKey Secret is used to encrypt the signature string on the client and decrypt the signature string on the server for authentication. Keep the AccessKey Secret confidential.

**3. Click Test Connection to perform a data source connectivity test.****4. After the connection is established, click Add.**

#### 4.3.2.2.6 HybridDB For MySQL

This topic describes how to add a cloud HybridDB for MySQL data source.

##### Context

When you add a HybridDB for MySQL data source, you must add the IP addresses of the machines to the HybridDB for MySQL whitelist. For more information about adding IP addresses to a whitelist, see *Configure a whitelist in HybridDB for MySQL User Guide*.

The procedure to add a HybridDB for MySQL data source is similar to adding an SQL Server data source. The default port is the port specific to HybridDB for MySQL.

##### Procedure

1. [Add a data source](#).

2. Click HybridDB for MySQL and enter the data source connection information, as shown in *Figure 4-9: Add a HybridDB for MySQL data source*.

Figure 4-9: Add a HybridDB for MySQL data source

**Add HybridDB for MySQL Database**

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 3306

\* Database: Enter a database name.

\* Username: Enter a username.

\* Password: Enter the password.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 3306.
Database	Enter the name of the database.
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.
4. After the connection is established, click Add.

#### 4.3.2.2.7 AnalyticDB for PostgreSQL

This topic describes how to add a cloud AnalyticDB for PostgreSQL data source.

##### Context

The procedure to add an AnalyticDB for PostgreSQL data source is similar to adding an SQL Server data source. The default port is the port specific to AnalyticDB for PostgreSQL. For more information about configuring a whitelist, *see* [Configure a whitelist in ApsaraDB for RDS User Guide](#).

##### Procedure

1. Click [Create Data Source](#).



2. Click **AnalyticDB for PostgreSQL** and enter the data source connection information, as shown in [Figure 4-10: Add an AnalyticDB for PostgreSQL data source](#).

Figure 4-10: Add an AnalyticDB for PostgreSQL data source

**Add AnalyticDB for PostgreSQL Database**

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 5432

\* Database: Enter a database name.

Schema: public

\* Username: Enter a username.

\* Password: Enter the password.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see [Quick BI User Guide > Create a data source > Data sources from cloud databases](#).

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 5432.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click **Test Connection** to perform a data source connectivity test.
4. After the connection is established, click **Add**.

### 4.3.2.2.8 PostgreSQL

This topic describes how to add a cloud PostgreSQL data source.

#### Context

The procedure to add a PostgreSQL data source is similar to adding a HybridDB for PostgreSQL data source. For more information about configuring a whitelist, see [Configure a whitelist in ApsaraDB for RDS User Guide](#).

#### Procedure

1. Click [Create Data Source](#).
2. Click PostgreSQL and enter the data source connection information, as shown in

[Figure 4-11: Add a PostgreSQL data source](#)

Figure 4-11: Add a PostgreSQL data source

**Add PostgreSQL Database**

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 5432

\* Database: Enter a database name.

Schema: public

\* Username: Enter a username.

\* Password: Enter the password.

SSL: ☐

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add



**Note:**

If you select the SSH check box, the data source supports MaxCompute Lightning, an interactive query service provided by MaxCompute.

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 5432.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.
4. After the connection is established, click Add.

#### 4.3.2.2.9 PPAS

This topic describes how to add a cloud PPAS data source.

##### Context

The procedure to add a PPAS data source is similar to adding a PostgreSQL data source. For more information about configuring a whitelist, see [Configure a whitelist in ApsaraDB for RDS User Guide](#) .

##### Procedure

1. Click [Create Data Source](#).

2. Click PPAS and enter the data source connection information, as shown in [Figure 4-12: Add a PPAS data source](#).

Figure 4-12: Add a PPAS data source

**Add PPAS Database**

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 5432

\* Database: Enter a database name.

Schema: public

\* Username: Enter a username.

\* Password: Enter the password.

**Note:** To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 5432.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.

4. After the connection is established, click Add.

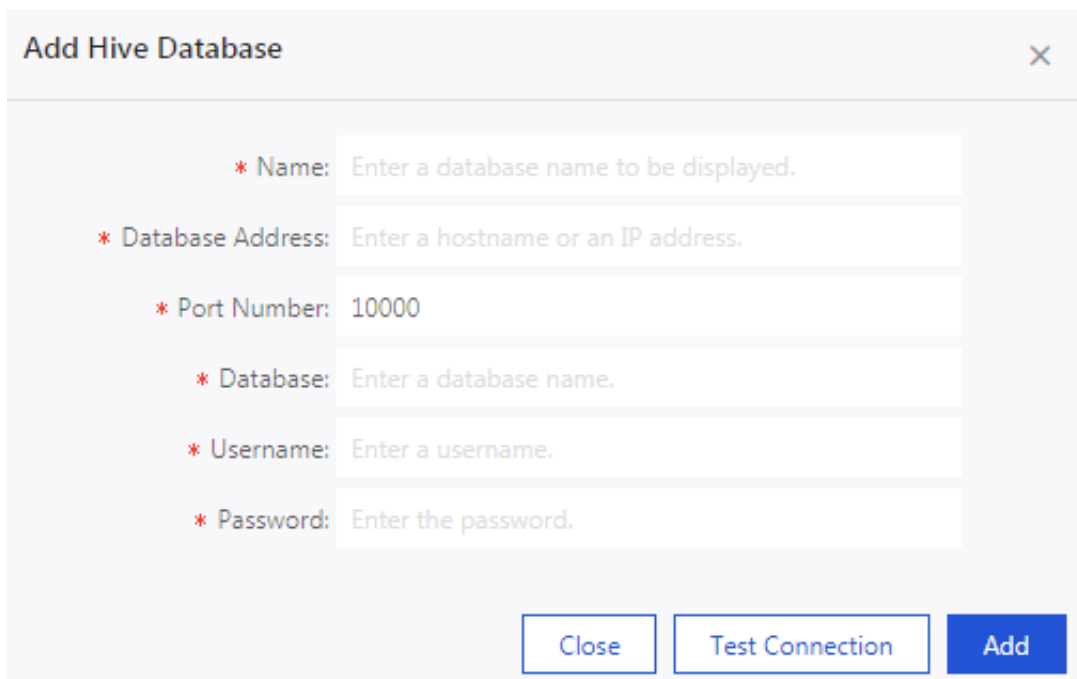
### 4.3.2.2.10 Hive

This topic describes how to add a cloud Hive data source.

#### Procedure

1. Click *Create Data Source*.
2. Click Hive and enter the data source connection information, as shown in *Figure 4-13: Add a Hive data source*.

Figure 4-13: Add a Hive data source



Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 10000.
Database	Enter the name of the database.
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.
4. After the connection is established, click Add.

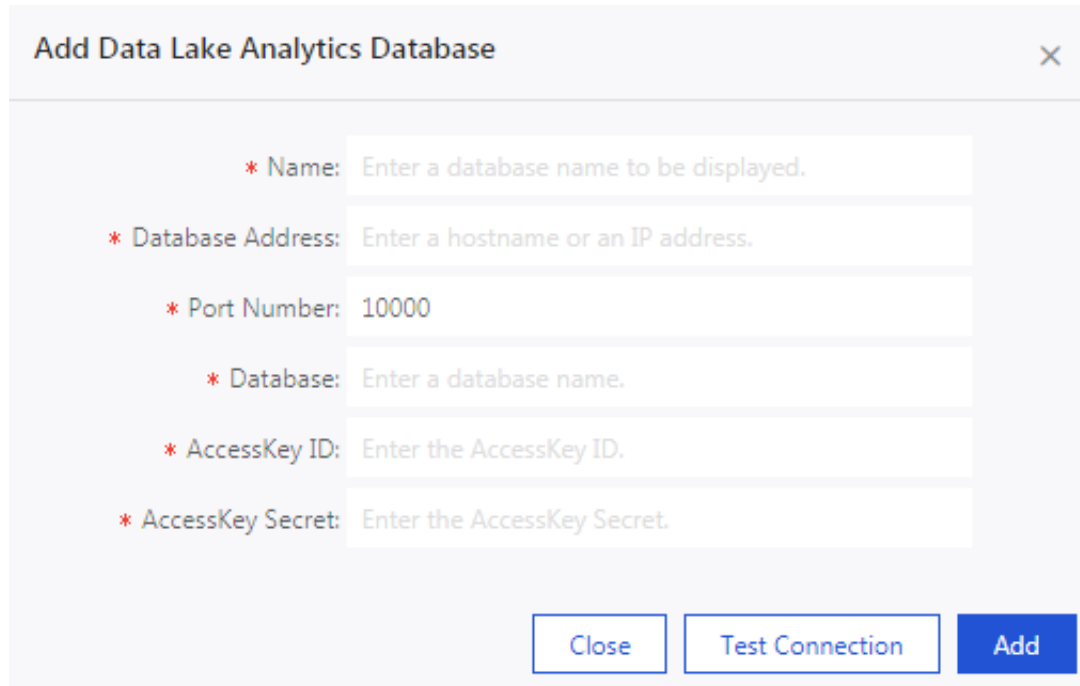
### 4.3.2.2.11 Data Lake Analytics

This topic describes how to add a cloud Data Lake Analytics data source.

#### Procedure

1. Click [Create Data Source](#).
2. Click **Data Lake Analytics** and enter the data source connection information.

Figure 4-14: Add a Data Lake Analytics data source



Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number. The default port is 10000.
Database	Enter the name of the database.
AccessKey ID	Enter the AccessKey ID of the account that purchased the data source instance.
AccessKey Secret	Enter the AccessKey Secret of the account that purchased the data source instance.

3. Click **Test Connection** to perform a data source connectivity test.
4. After the connection is established, click **Add**.

### 4.3.2.2.12 DRDS

This topic describes how to add a cloud DRDS data source.

#### Procedure

1. Click [Create Data Source](#).
2. Click **DRDS** and enter the data source connection information.

Figure 4-15: Add a DRDS data source

**Add DRDS Database**

\* Name: Enter a database name to be displayed.

\* Database Address: Enter a hostname or an IP address.

\* Port Number: 3306

\* Database: Enter a database name.

\* Username: Enter a username.

\* Password: Enter the password.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number. The default port is 3306.
Database	Enter the name of the database.
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click **Test Connection** to perform a data source connectivity test.
4. After the connection is established, click **Add**.

### 4.3.2.3 User-created data sources

#### 4.3.2.3.1 MySQL

This topic describes how to add a user-created MySQL data source. You can access the MySQL data source through an SSH tunnel.

#### Context

The procedure to add a user-created MySQL data source is similar to adding a cloud MySQL data source. You must use the following method to open the specified port of the firewall to allow Quick BI to access the MySQL database:

1. Run the following command to open the configuration file of the firewall.

```
vi /etc/sysconfig/iptables
```

2. Add the following command to the configuration file:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3306
-j
 ACCEPT
```

3. After the configuration is completed, run the following command to restart iptables:

```
service iptables restart
```

#### Procedure

1. Click [Create Data Source](#).



**2. Click MySQL and enter the data source connection information, as shown in**

*Figure 4-16: Add a MySQL data source.*

Figure 4-16: Add a MySQL data source

**Add MySQL Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: IP

\* Port Number: 3306

\* Database: Enter a database name.

\* Username: Enter a username.

\* Password: Enter the password.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 3306.
Database	Enter the name of the database.
Username	Enter the username of the database.
Password	Enter the password of the database.

**3. Click Test Connection to perform a data source connectivity test.****4. After the connection is established, click Add.**

### 4.3.2.3.2 SQL Server

This topic describes how to add a user-created SQL Server data source. You can access the data source through an SSH tunnel.

#### Context

The procedure to add a user-created SQL Server data source is similar to adding a cloud SQL Server data source. You must use the following method to open the specified port of the firewall to allow Quick BI to access the SQL Server database:

1. Run the following command to open the configuration file of the firewall.

```
vi /etc/sysconfig/iptables
```

2. Add the following command to the configuration file:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 1433
-j
ACCEPT
```

3. After the configuration is completed, run the following command to restart iptables:

```
service iptables restart
```

#### Procedure

1. Click [Create Data Source](#).

**2. Click SQL Server and enter the data source connection information, as shown in**

*Figure 4-17: Add a SQL Server data source.*

Figure 4-17: Add a SQL Server data source

**Add SQL Server Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: IP

\* Port Number: 1433

\* Database: Enter a database name.

Schema: dbo

\* Username: Enter a username.

\* Password: Enter the password.

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

[Close] [Test Connection] [Add]

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 1433.
Database	Enter the name of the database.
Schema	Enter the schema. The default schema is dbo.
Username	Enter the username of the database.
Password	Enter the password of the database.

**3. Click Test Connection to perform a data source connectivity test.**

4. After the connection is established, click Add.



**Note:**

After a data source has been added, you cannot add it again. If you attempt to add the same data source, an error message appears.

### 4.3.2.3.3 PostgreSQL

This topic describes how to add a user-created PostgreSQL data source. You can access the PostgreSQL data source through an SSH tunnel.

#### Context

The procedure to add a user-created PostgreSQL data source is similar to adding a cloud PostgreSQL data source. You must use the following method to open the specified port of the firewall to allow Quick BI to access the PostgreSQL database:

1. Run the following command to open the configuration file of the firewall.

```
vi /etc/sysconfig/iptables
```

2. Add the following command to the configuration file:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5432
-j
ACCEPT
```

3. After the configuration is completed, run the following command to restart iptables:

```
service iptables restart
```

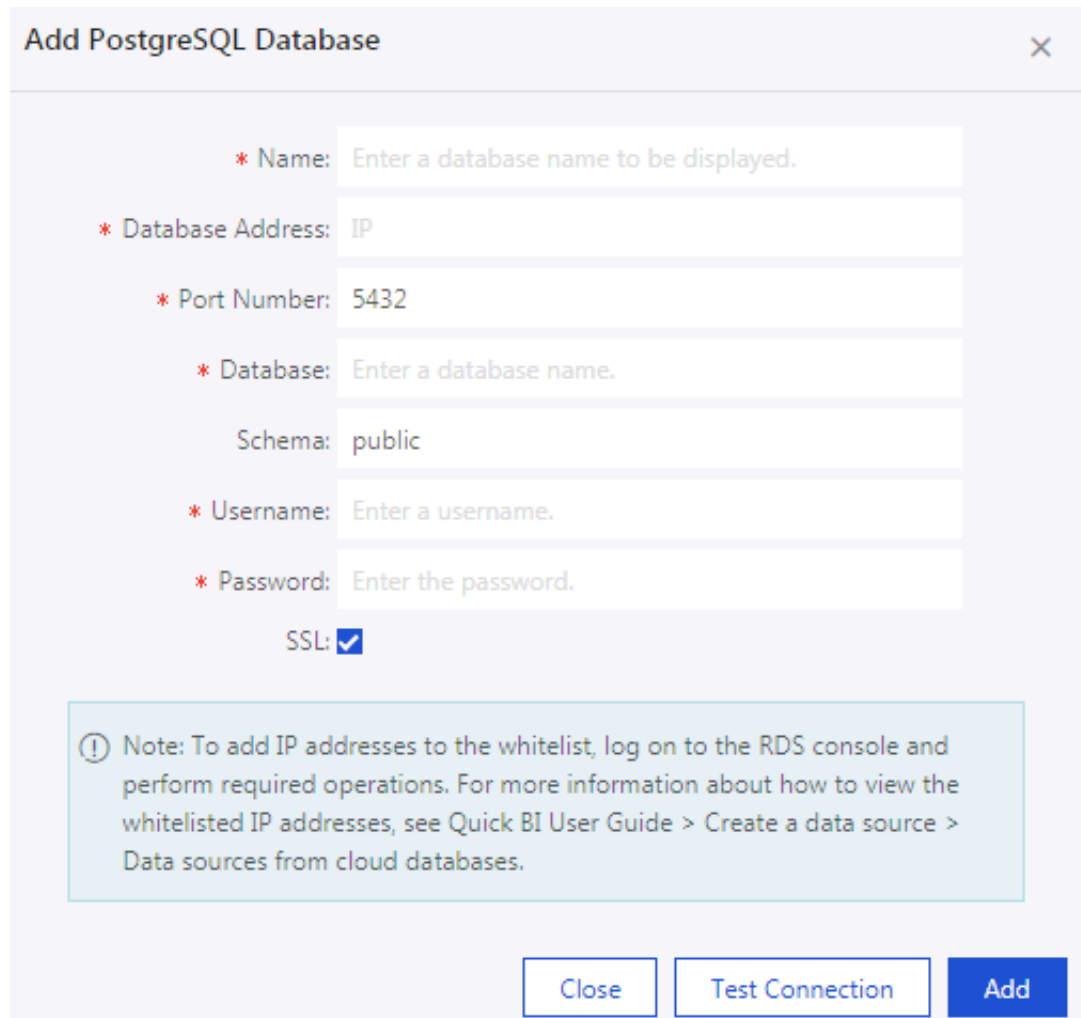
#### Procedure

1. Click [Create Data Source](#).

**2. Click PostgreSQL and enter the data source connection information, as shown in**

*Figure 4-18: Add a PostgreSQL data source.*

Figure 4-18: Add a PostgreSQL data source



**Add PostgreSQL Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: IP

\* Port Number: 5432

\* Database: Enter a database name.

Schema: public

\* Username: Enter a username.

\* Password: Enter the password.

SSL: ☒

ⓘ Note: To add IP addresses to the whitelist, log on to the RDS console and perform required operations. For more information about how to view the whitelisted IP addresses, see Quick BI User Guide > Create a data source > Data sources from cloud databases.

Close Test Connection Add

**Note:**

If you select the SSH check box, the data source supports MaxCompute Lightning, an interactive query service provided by MaxCompute.

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database .
Port Number	Enter the port number of the database. The default port is 5432.
Database	Enter the name of the database.

Parameter	Description
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click **Test Connection** to perform a data source connectivity test.

4. After the connection is established, click **Add**.

#### 4.3.2.3.4 Oracle

This topic describes how to add a user-created Oracle data source. You can access the Oracle data source through an SSH tunnel.

##### Procedure

1. Click *Create Data Source*.
2. Click **Oracle** and enter the data source connection information, as shown in *Figure 4-19: Add an Oracle data source*.

Figure 4-19: Add an Oracle data source

**Add Oracle Database** [X]

\* Name: Enter a database name to be displayed.

\* Database Address: IP

\* Port Number: 1521

\* Database: Enter a database name.

Schema: The default is the login username

\* Username: Enter a username.

\* Password: Enter the password.

[Close] [Test Connection] [Add]

Parameter	Description
Name	Specify a name for the data source.

Parameter	Description
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 1521.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click **Test Connection** to perform a data source connectivity test.
4. After the connection is established, click **Add**.

#### 4.3.2.3.5 Hive

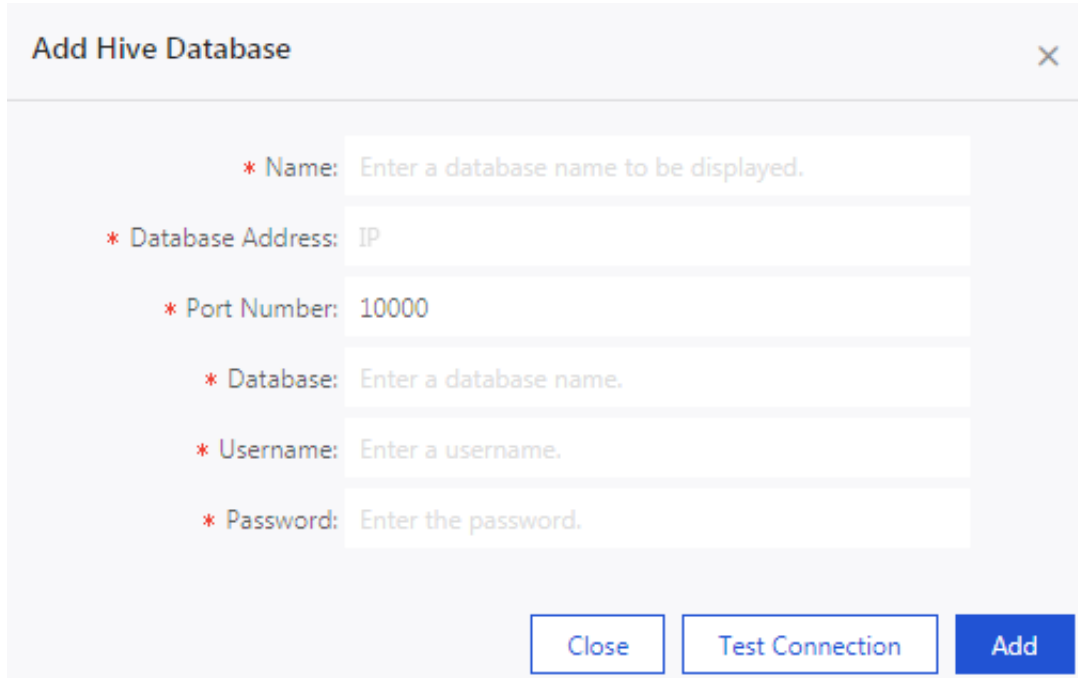
This topic describes how to add a user-created Hive data source.

##### Procedure

1. On the **Data Sources** page, click [Create Data Source](#) in the upper-right corner.

2. Select Hive and enter the data source connection information, as shown in [Figure 4-20: Add a Hive data source](#).

Figure 4-20: Add a Hive data source



Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 10000.
Database	Enter the name of the database.
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.
4. After the connection is established, click Add.

#### 4.3.2.3.6 Vertica

This topic describes how to add a user-created Vertical data source. You can access the Vertica data source through an SSH tunnel.

#### Procedure

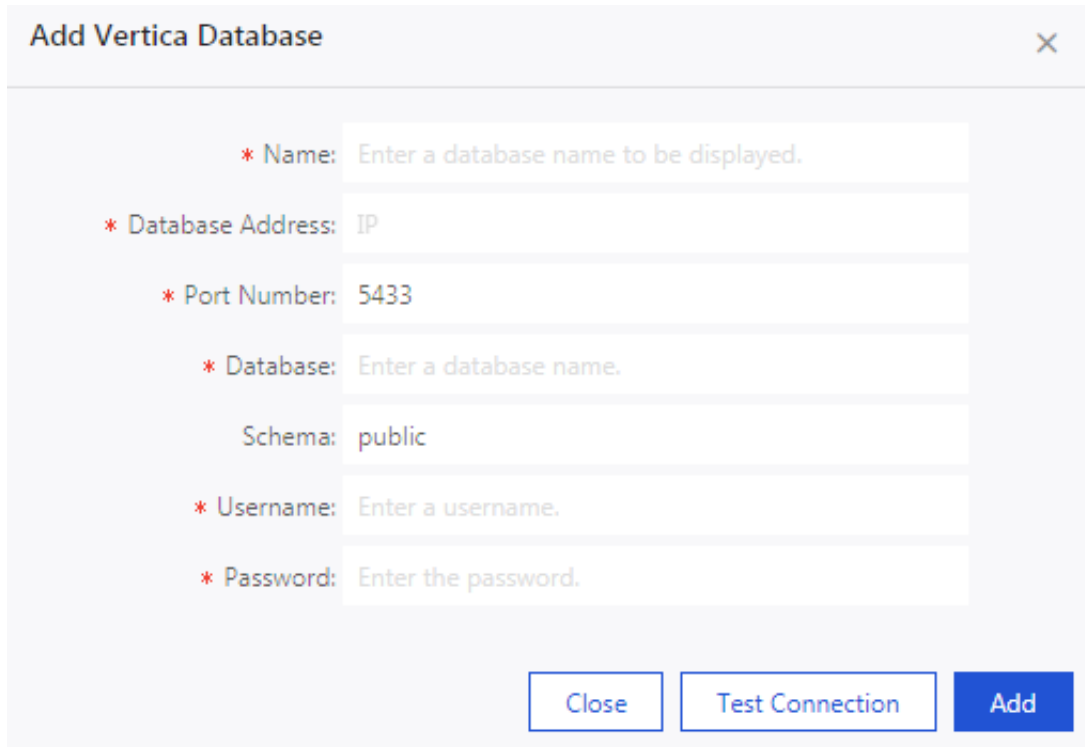
1. On the Data Sources page, click [Create Data Source](#) in the upper-right corner.



**2. Select Vertica and enter the data source connection information, as shown in**

*Figure 4-21: Add a Vertical data source.*

Figure 4-21: Add a Vertical data source



Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 5433.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.
Password	Enter the password of the database.

**3. Click Test Connection to perform a data source connectivity test.****4. After the connection is established, click Add.**

### 4.3.2.3.7 IBM DB2 LUW

This topic describes how to add a user-created IBM DB2 LUW data source. You can access the IBM DB2 LUW data source through an SSH tunnel.

#### Procedure

1. On the Data Sources page, click [Create Data Source](#) in the upper-right corner.
2. Select IBM DB2 LUW and enter the data source connection information, as shown in [Figure 4-22: Add an IBM DB2 LUW data source](#).

Figure 4-22: Add an IBM DB2 LUW data source

**Add IBM DB2 LUW Database**

\* Name: Enter a database name to be displayed.

\* Database Address: IP

\* Port Number: 50000

\* Database: Enter a database name.

Schema: DB2INST1

\* Username: Enter a username.

\* Password: Enter the password.

Close Test Connection Add

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 50000.
Database	Enter the name of the database.
Schema	DB2INST1
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.

4. After the connection is established, click **Add**.

#### 4.3.2.3.8 SAP IQ (Sybase IQ)

This topic describes how to add a user-created SAP IQ (Sybase IQ) data source. You can access the SAP IQ data source through an SSH tunnel.

##### Procedure

1. On the Data Sources page, click [Create Data Source](#) in the upper-right corner.
2. Select SAP IQ (Sybase IQ) and enter the data source connection information.

Figure 4-23: Add an SAP IQ (Sybase IQ) data source

The screenshot shows a dialog box titled "Add SAP IQ (Sybase IQ) Database" with a close button (X) in the top right corner. The dialog contains the following fields and labels:

- \* Name: Enter a database name to be displayed.
- \* Database Address: IP
- \* Port Number: 2638
- \* Database: Enter a database name.
- Schema: sybase
- \* Username: Enter a username.
- \* Password: Enter the password.

At the bottom right, there are three buttons: "Close", "Test Connection", and "Add". The "Add" button is highlighted in blue.

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 2638.
Database	Enter the name of the database.
Schema	sybase
Username	Enter the username of the database.
Password	Enter the password of the database.

3. Click **Test Connection** to perform a data source connectivity test.
4. After the connection is established, click **Add**.

#### 4.3.2.3.9 SAP HANA

This topic describes how to add a user-created SAP HANA data source. You can access the SAP HANA data source through an SSH tunnel.

##### Procedure

1. On the **Data Sources** page, click [Create Data Source](#) in the upper-right corner.
2. Select **SAP HANA** and enter the data source connection information.

Figure 4-24: Add an SAP HANA data source

The screenshot shows a dialog box titled "Add SAP HANA Database" with a close button (X) in the top right corner. The dialog contains the following fields and values:

- Name:** Enter a database name to be displayed.
- Database Address:** IP
- Port Number:** 30015
- Database:** Enter a database name.
- Schema:** public
- Username:** Enter a username.
- Password:** Enter the password.

At the bottom of the dialog, there are three buttons: "Close", "Test Connection", and "Add". The "Add" button is highlighted in blue.

Parameter	Description
Name	Specify a name for the data source.
Database Address	Enter the hostname or IP address of the database.
Port Number	Enter the port number of the database. The default port is 30015.
Database	Enter the name of the database.
Schema	public
Username	Enter the username of the database.

Parameter	Description
Password	Enter the password of the database.

3. Click Test Connection to perform a data source connectivity test.

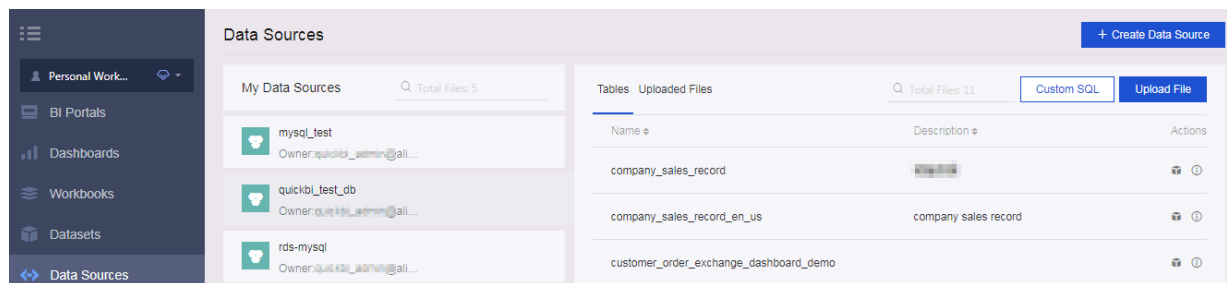
4. After the connection is established, click Add.

#### 4.3.2.4 Data sources

This topic describes the basic information about the Data Sources page.

You can manage data sources on the Data Sources page, including create, edit, and delete a data source, as shown in [Figure 4-25: The Data Sources page](#).

Figure 4-25: The Data Sources page



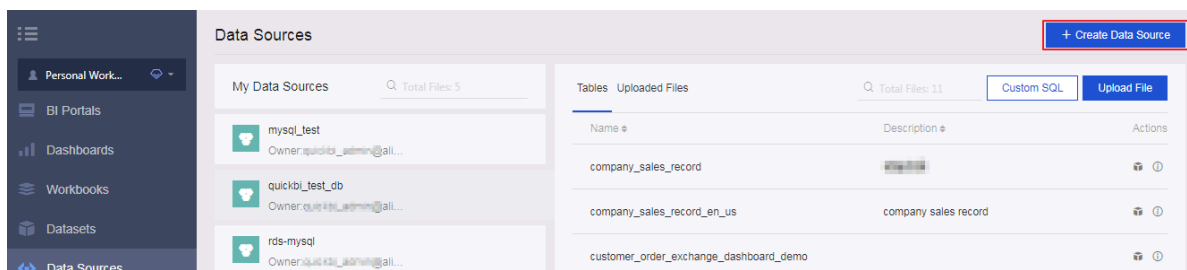
#### 4.3.2.5 Add a data source

Datasets, workbooks, dashboards, and BI portals are created based on data sources. This topic describes how to add a data source.

##### Procedure

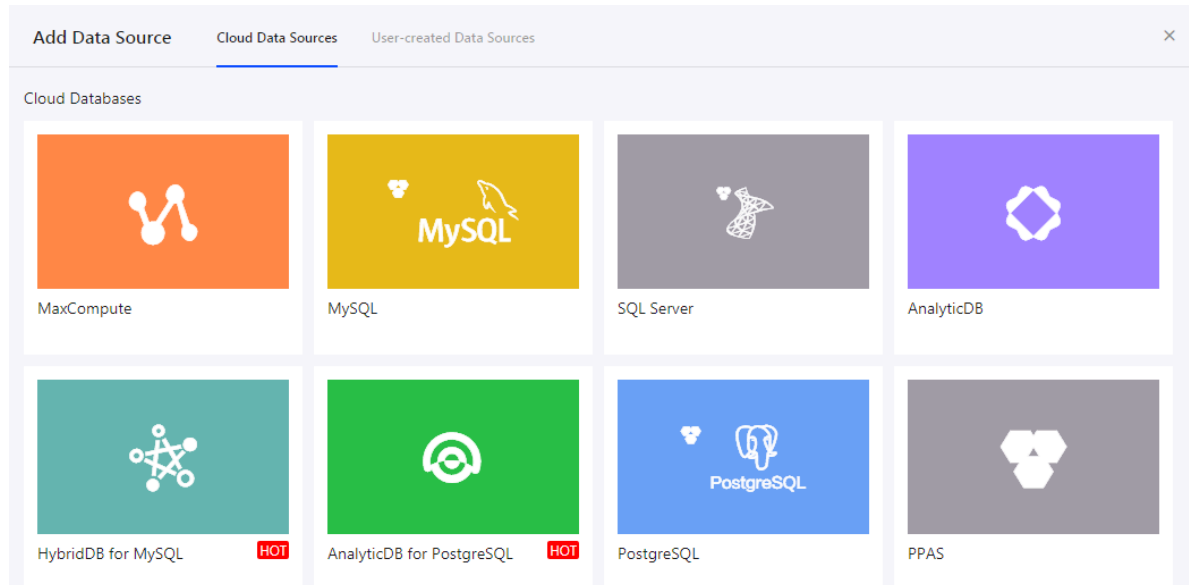
1. [Log on to the Quick BI console](#).
2. In the top navigation pane, click Workspace.
3. In the left-side navigation pane, click Data Sources.
4. On the Data Sources page, click Create Data Source in the upper-right corner, as shown in the following figure [Figure 4-26: Add a data source](#).

Figure 4-26: Add a data source



5. In the dialog box that appears, select a data source type, as shown in the following figure *Figure 4-27: Select a data source type*.

Figure 4-27: Select a data source type



6. Enter the data source connection information as required, and click Add.

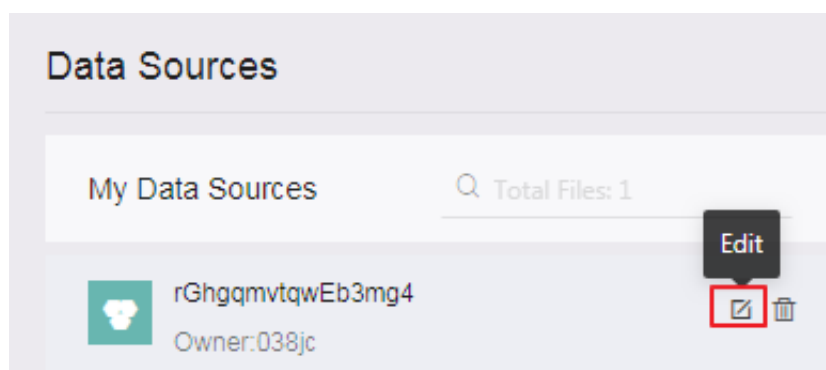
#### 4.3.2.6 Edit a data source

This topic describes how to edit a data source.

##### Procedure

1. Log on to the *Quick BI console*.
2. In the left-side navigation pane, click Data Sources.
3. Select the target data source from the Data Sources list.
4. Click the Edit icon to edit the data source, as shown in *Figure 4-28: Edit a data source*.

Figure 4-28: Edit a data source



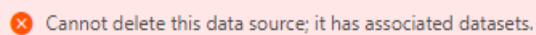
### 4.3.2.7 Delete a data source

This topic describes how to delete a data source and lists the issues that may occur during this process.

#### Context

If you have created a dataset based on a data source, then you cannot delete the data source, as shown in [Figure 4-29: Delete a data source](#).

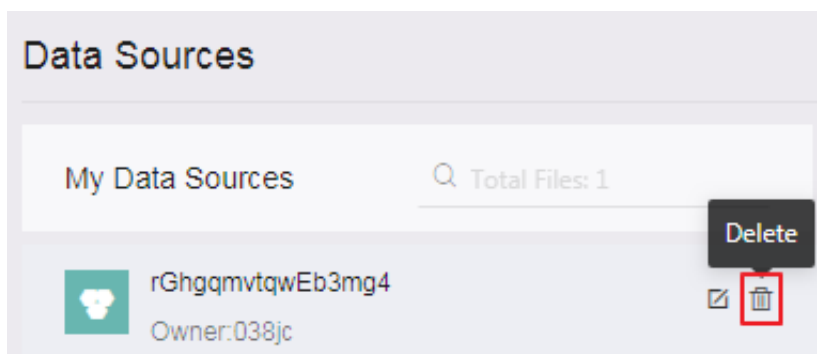
Figure 4-29: Delete a data source



#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Data Sources.
3. Select the target data source from the Data Sources list.
4. Click the Delete icon to delete the data source, as shown in [Figure 4-30: Delete the data source](#).

Figure 4-30: Delete the data source



### 4.3.2.8 Search for a data source

This topic describes how to search for a specific data source.

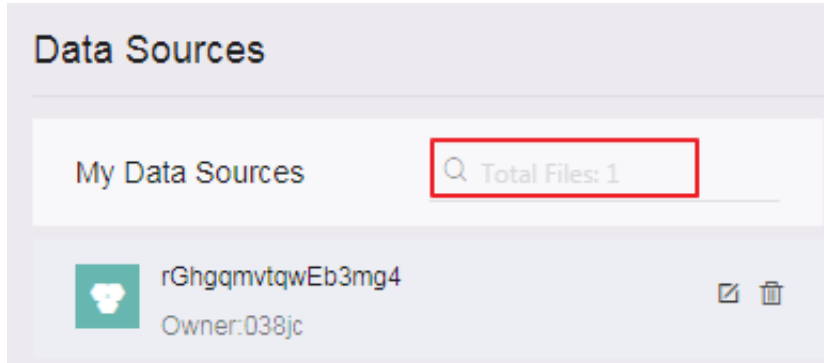
#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Data Sources.

**3. Enter a keyword into the search box to search for the data source, as shown in**

*Figure 4-31: Search for a data source.*

Figure 4-31: Search for a data source

**4. Click the Search icon to search for the data source.**

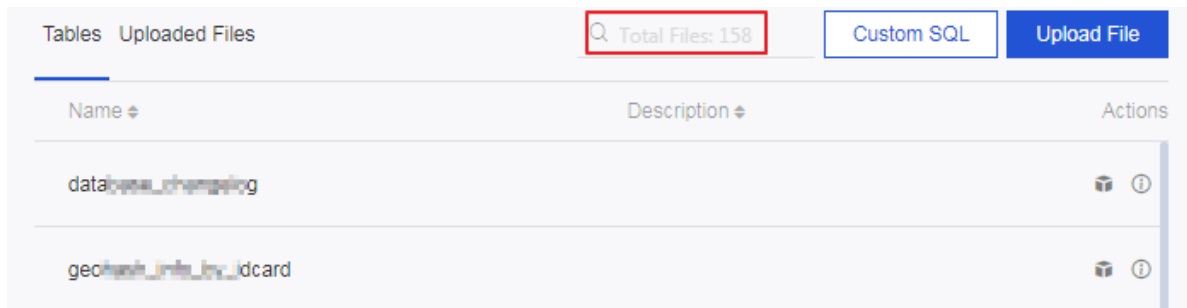
#### 4.3.2.9 Search for tables under a data source

This topic describes how to search for a specific table under a data source.

##### Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click Data Sources.
3. Select the target data source. All tables under the data source are listed on the right side of the page.
4. Enter a keyword in the search box to search for a table, as shown in *Figure 4-32: Search for a table under a data source.*

Figure 4-32: Search for a table under a data source

**5. Click the Search icon to search for the table.**



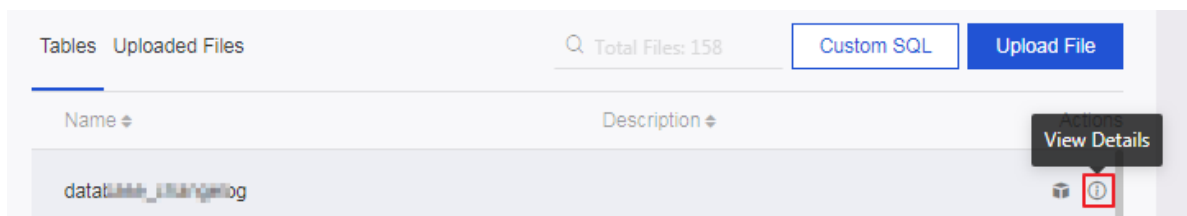
### 4.3.2.10 Query table details

Quick BI allows you to query tables and table details under a data source.

#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Data Sources.
3. Select the target data source. All tables under the data source are listed on the right side of the page.
4. Find the target table and click the View Details icon to view the table details and fields, as shown in [Figure 4-33: Query table details](#).

Figure 4-33: Query table details



## 4.3.3 Datasets

### 4.3.3.1 Overview

You can use tables from data sources to create datasets. The following topics describe common operations on datasets, for example, create, edit, and query a dataset.

You can create a dataset using the following methods:

- Create a dataset from a data source
- Upload CSV files to create datasets
- Use custom SQL statements to create datasets

### 4.3.3.2 Create datasets

#### 4.3.3.2.1 Create datasets from data sources

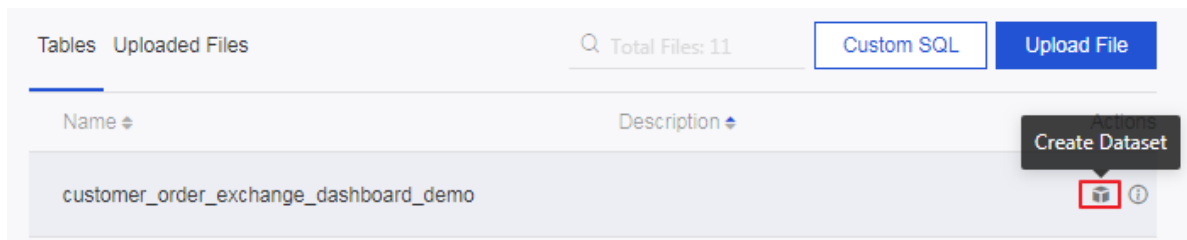
This topic describes how to create a dataset from a data source.

#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Data Sources.

3. Select the target data source. All tables under the data source are automatically listed on the right side of the page.
4. Select a table and click the Create Dataset icon, as shown in [Figure 4-34: Create a dataset from a data source](#).

Figure 4-34: Create a dataset from a data source



After the dataset is created, you are redirected to the Datasets page. Newly created datasets are marked with New. This allows you to quickly find newly created datasets, as shown in [Figure 4-35: The dataset is created](#).

Figure 4-35: The dataset is created



#### 4.3.3.2.2 Upload CSV files to create datasets

This topic describes how to create datasets by uploading CSV files.

##### Procedure

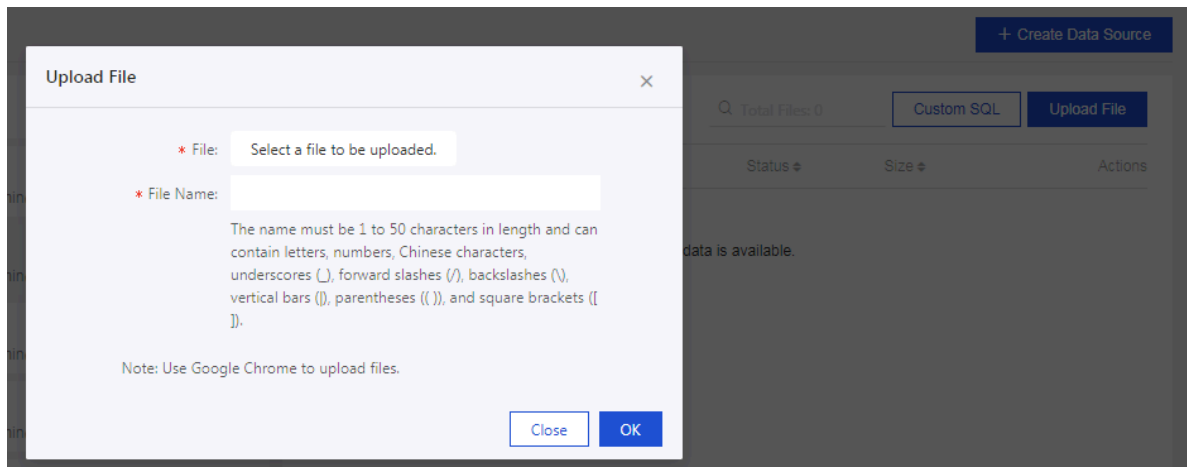
1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Data Sources.
3. Click Upload File in the upper-right corner.
4. In the Upload File dialog box that appears, select the target file, enter the file name, and click OK, as shown in [Figure 4-36: Upload a file](#).



**Note:**

**After the file is uploaded, you are redirected to the Uploaded Files page.**

Figure 4-36: Upload a file



5. On the Uploaded Files page, select the file that you have uploaded and click the **Create Dataset** icon to create a dataset.

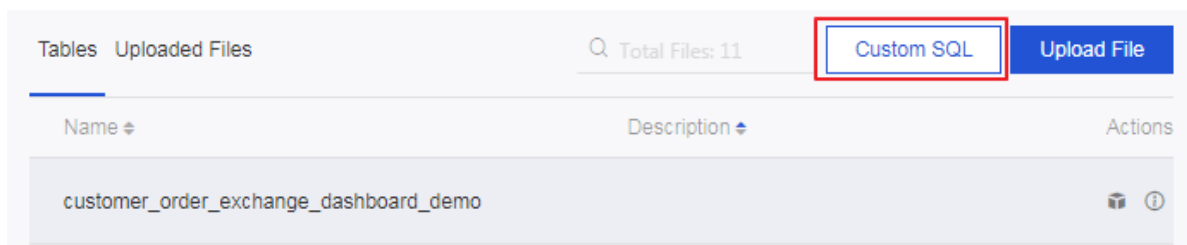
#### 4.3.3.2.3 Use custom SQL statements to create datasets

This topic describes how to use custom SQL statements to create a dataset.

##### Procedure

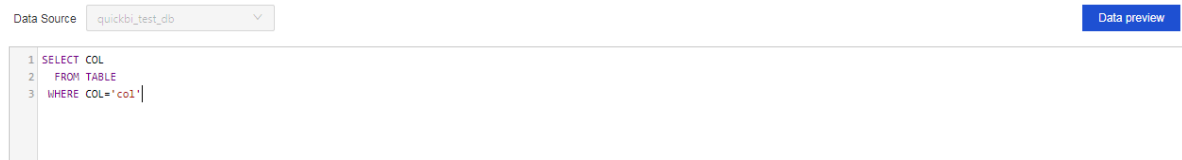
1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Data Sources**.
3. Select the target data source from the **Data Source** list.
4. On the right side of the page, click **Custom SQL** to create a dataset, as shown in [Figure 4-37: Custom SQL statements.](#)

Figure 4-37: Custom SQL statements



5. On the edit page, enter the custom SQL statements, as shown in [Figure 4-38: Enter SQL statements](#).

Figure 4-38: Enter SQL statements



**Note:**

You can click Data Preview to preview the first 100 rows of data.

6. Click Save in the upper-right corner to save the data source as a dataset.

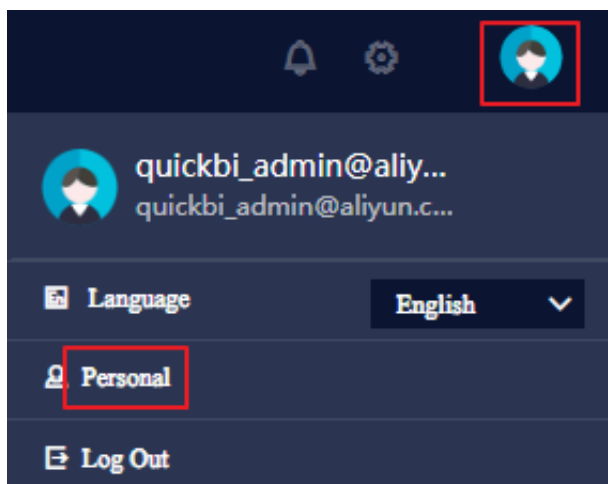
### 4.3.3.3 Specify a method to name dimensions and measures

Quick BI automatically creates datasets based on the metadata of physical tables and converts fields in the physical tables to dimensions or measures in datasets. Dimensions and measures are automatically named after the names or description of physical table fields. On the Datasets page, hover over your avatar and select Personal. In the dialog box that appears, set the parameters based on your needs. After you set the parameters, dimensions and measures will be named based on the settings.

#### Procedure

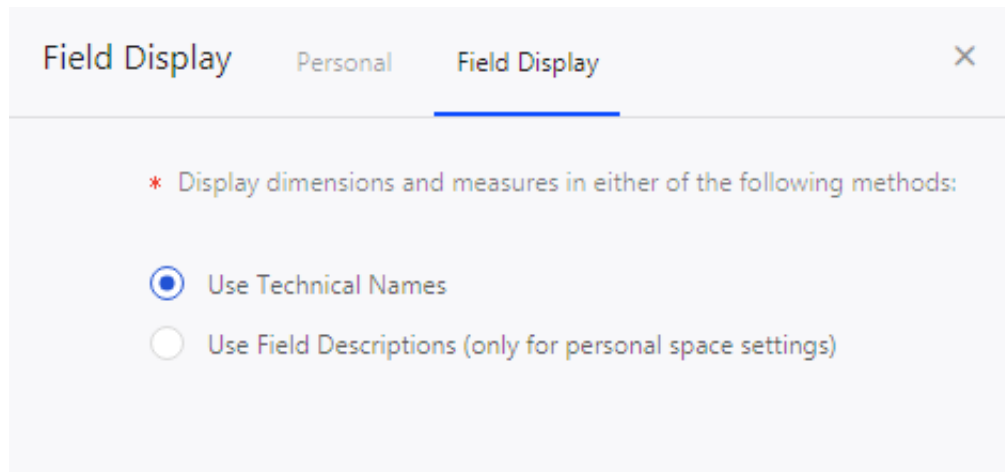
1. [Log on to the Quick BI console](#).
2. Hover over your avatar and select Personal, as shown in [Personal](#).

Figure 4-39: Select Personal



3. In the Personal dialog box that appears, click **Field Display** and select a method to name dimensions and measures in a dataset, as shown in *Figure 4-40: Select a method to name dimensions and measures*.

Figure 4-40: Select a method to name dimensions and measures



#### 4.3.3.4 Edit a dataset

##### 4.3.3.4.1 Edit a dimension

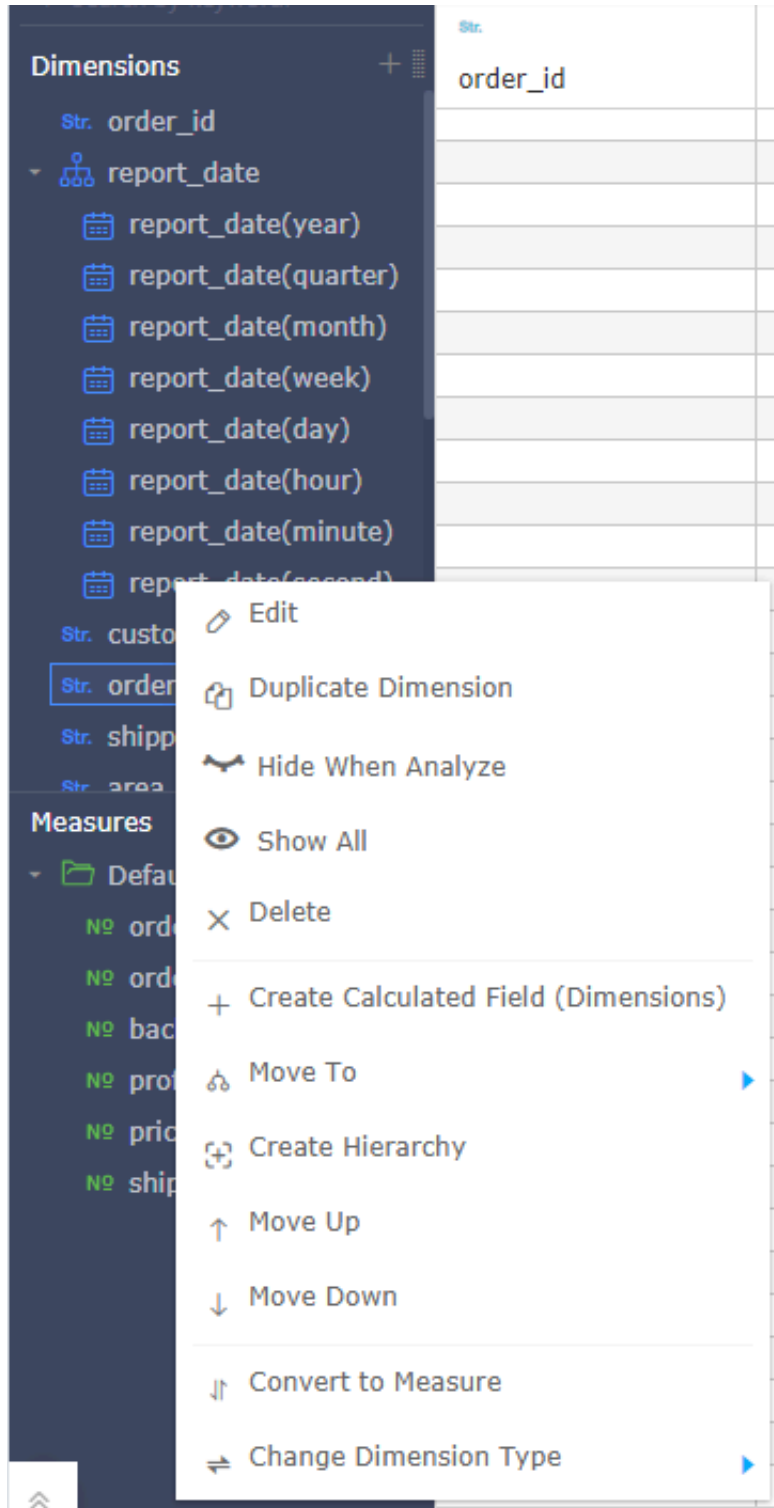
By default, if the data type of a table is character or other types, the field is classified as a dimension by the system. You can edit fields in the Dimensions or Measures list.

#### Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click **Datasets**.
3. In the Datasets list, click the target dataset to go to the dataset edit page.
4. Select a field from the Dimensions list.

5. Right-click the field, and a shortcut menu appears, as shown in the following figure.

Figure 4-41: Shortcut menu



- **Edit:** You can edit the dimension name and description.

- **Duplicate Measure:** You can copy a dimension. The name of the duplicate field ends with Duplicate.
- **Hide When Analyze:** You can hide specific dimensions as needed.
- **Show All:** Displays all dimensions.
- **Delete:** You can delete a dimension.
- **Create Calculated Field (Measures):** You can create a new dimension and customize the calculation method.
- **Move To:** You can move a measure to an existing hierarchy for drilling.
- **Create Hierarchy:** You can create a hierarchy and move a dimension to this hierarchy.
- **Move Up or Move Down:** You can select Move Up or Move Down to move a dimension upwards or downwards, or directly drag a dimension to the target position.
- **Convert to Measure:** You can convert a dimension to a measure.
- **Change Dimension Type:** You can change the type of a dimension to Default, Date/Time, Geo, String, and Number.

For example, when you create a geo bubble map or geo map, you must change the dimension type to Geo. Otherwise, you cannot create geo maps.

#### 4.3.3.4.2 Edit a measure

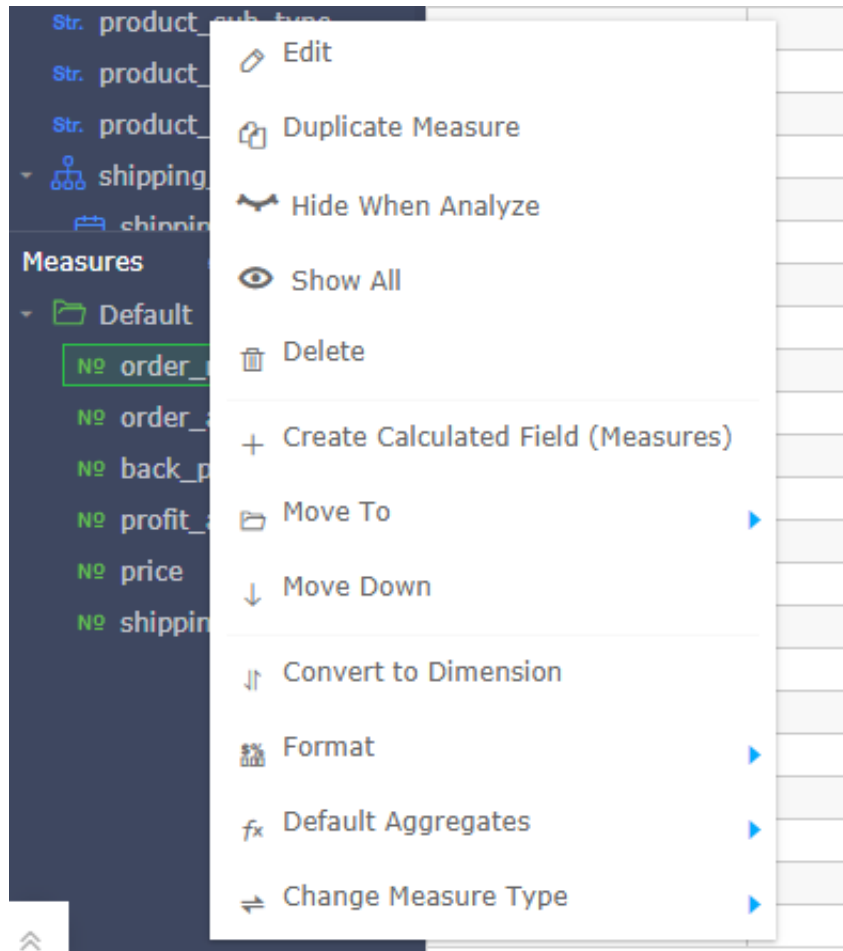
By default, if the data type of a table is numeric, then the field is classified as a measure by the system. You can edit fields in the Measures or Dimensions list.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. In the Datasets list, click the target dataset to go to the dataset edit page.
4. Select a field from the Measures list.

5. Right-click the field, and a shortcut menu appears, as shown in [Figure 4-42: Shortcut menu](#).

Figure 4-42: Shortcut menu



- **Edit:** You can edit the measure name and description.
- **Duplicate Measure:** You can copy a measure. The name of the duplicate field ends with Duplicate.
- **Hide When Analyze:** You can hide specific measures as needed.
- **Show All:** Displays all measures.
- **Delete:** You can delete a measure.
- **Create Calculated Field (Measures):** You can create a new measure and customize the calculation method.
- **Move To:** You can move a measure to an existing folder.
- **Move Up or Move Down:** You can select Move Up or Move Down to move a measure upwards or downwards, or directly drag a measure to the target position.



- **Convert to Dimension:** You can convert a measure to a dimension.
- **Format:** You can specify the display format of numbers.
- **Default Aggregates:** You can specify an aggregate function. Aggregate functions include Sum, Count, Count Distinct, Maximum, Minimum, and Average.
- **Change Measure Type:** You can change the type of the measure to String or Number.

#### 4.3.3.4.3 Toolbar

The dataset edit page in the Preview mode provides a toolbar and a shortcut menu, as shown in [Figure 4-43: Toolbar](#) and [Figure 4-44: Shortcut menu](#).

Figure 4-43: Toolbar



Figure 4-44: Shortcut menu



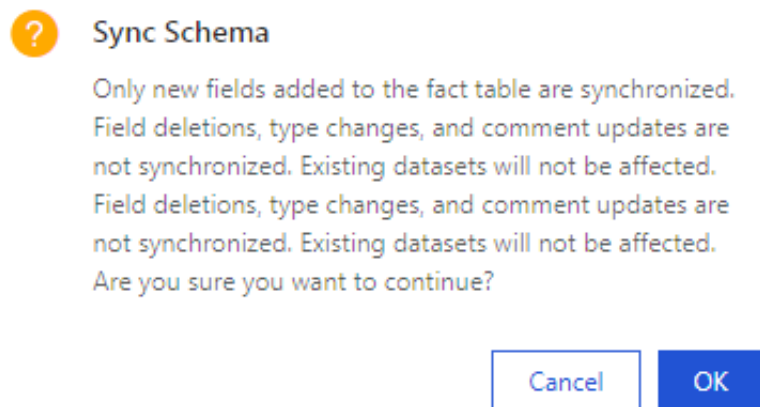
#### Toolbar

- **Lock:** Multiple users can edit a dataset at the same time, but only one user at a time can save the changes. To save a dataset when multiple users are editing it, a user must lock the dataset, refresh the page to update the data, apply the changes to the dataset, and then save the dataset. If a user locks and then edits a dataset without refreshing the page, the changes made by the previous user will be overwritten.

- **Sync Schema:** Synchronizes the physical table with the dataset. For example, changes made to the physical table, such as new fields, will be synchronized to the dataset.

If a field is deleted from the physical table, a field is renamed, a comment is modified, or the table schema is changed, the system will not delete the relevant data in the dataset.

Figure 4-45: Synchronize the schema



- **Refresh Preview:** Refreshes the dataset and displays the data in the Preview mode. If you want to view the latest data in real time, save the dataset and then refresh the page.
- **Set Filter Criteria:** Filters data in the dataset to avoid full table scan.
- **Save As:** Saves the current dataset as a new dataset. This operation allows you to quickly duplicate a dataset or backup data.
- **Save:** Saves the dataset.

After you add new fields, delete fields, convert dimensions to measures, or convert measures to dimensions, you need to save the dataset. After you save the dataset, you can refresh the page to view the updated dataset.

#### Shortcut menu

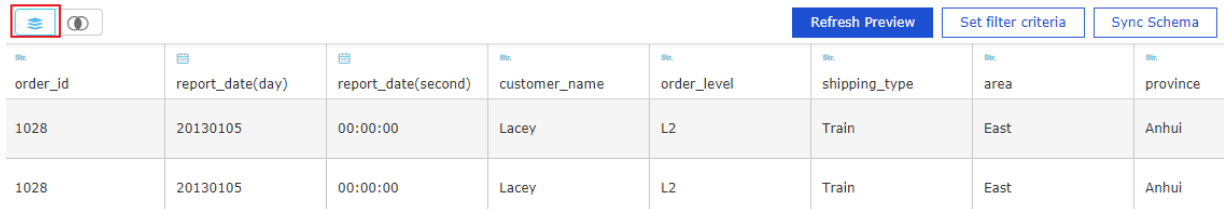
- **Dashboard:** You can click Dashboard to go to the Create Dashboard page.
- **Workbook:** You can click Workbook to create a workbook and edit it.
- **Dataset:** You can click Dataset to create a dataset.
- **Data Portal:** You can click BI Portal to create a BI portal and edit it.
- **Retrieve Data:** You can click Retrieve Data to go the Add Data Source page.

#### 4.3.3.4.4 Preview data

This topic describes how to preview data on the dataset edit page.

To preview data, click the Preview icon, as shown in [Figure 4-46: Preview data](#).

Figure 4-46: Preview data



order_id	report_date(day)	report_date(second)	customer_name	order_level	shipping_type	area	province
1028	20130105	00:00:00	Lacey	L2	Train	East	Anhui
1028	20130105	00:00:00	Lacey	L2	Train	East	Anhui

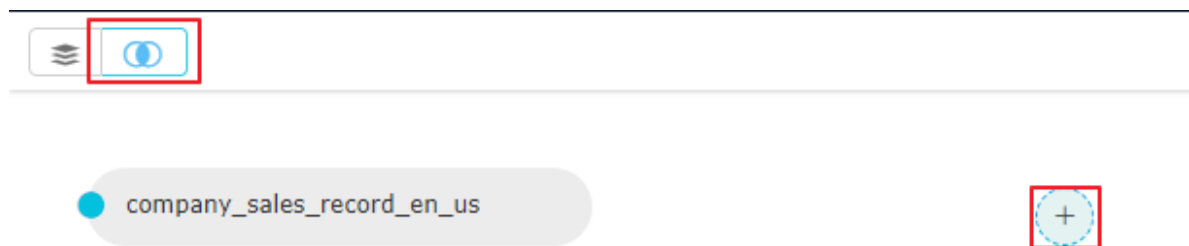
#### 4.3.3.4.5 Table join and examples

This topic describes how to join tables when you edit a dataset.

##### Procedure

1. Log on to the [Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. In the Datasets list, click the target dataset, for example, `company_sales_record`, to go to the dataset edit page.
4. Click the Table Join icon to switch the dataset to the Table Join mode, as shown in [Figure 4-47: Table join](#).

Figure 4-47: Table join



5. Click the plus sign (+) and a Build Table Join for `company_sales_record` Associated Model dialog box appears.

6. Select the field that is used to join tables, and a join type, as shown in [Figure 4-48](#):

*Edit the table join model.*

Figure 4-48: Edit the table join model

Build Table Join for company\_sales\_record\_en\_us Associated Model

Dataset Field	Join Type	Join Table	Join Field	Actions
order_id	Inner Join	company_sales_rec...	order_id	

Add Join Field

Quick BI supports the following join types:

- **Inner Join:** Returns data records that match the specified join field in the two tables.
- **Left Outer Join:** Returns all data records in the left table and data records that match the specified join field in the two tables.
- **All Join:** Returns all data records in the two tables.



**Note:**

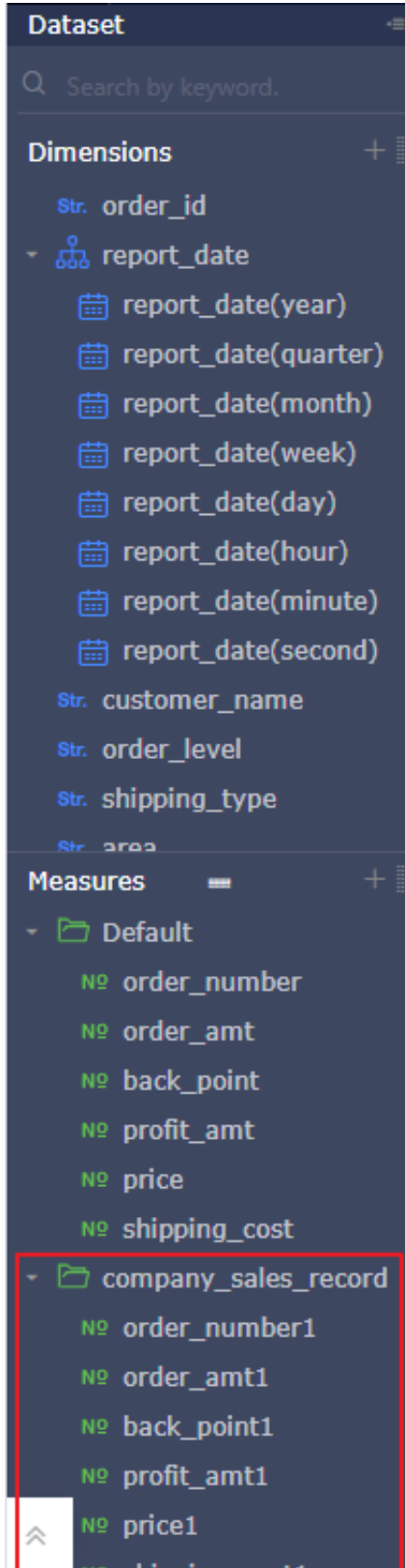
Currently, MySQL data sources do not support the All Join type.

7. Click Add Join Field to add multiple join fields.
8. Click OK to save the model.
9. Click the Preview icon to preview the data.
10. In the Preview mode, click Save to save the dataset.

11. Click **Refresh Preview** to view the data after table join, as shown in [Figure 4-49](#):

*Table join results.*

Figure 4-49: Table join results



### 4.3.3.4.6 Calculated fields

#### 4.3.3.4.6.1 Overview

Quick BI allows you to use existing fields and functions supported by SQL to generate new columns. These new columns are called calculated fields, and they must meet the SQL syntax that is used to define columns.

If you want to perform calculations based on existing data in the data source, you can add calculated fields. When you create a calculated field, you can use measure or dimension names that are understandable to sales personnel as expression parameters. When the Quick BI executes the SQL statements, semantic logic expressions will be translated into column expressions consisting of the physical field names.

To add a calculated field, click the plus sign (+) next to Dimensions or Measures. In the Edit Calculated Field dialog box that appears, specify the field name and select a supported function.

Fields selected from the Dimensions list will be automatically classified as calculated dimensions. Fields selected from the Measures list will be automatically classified as calculated measures.

In the Expression edit box, you can select any functions and expressions that are supported by the current data source.

You must enter the function name. You can manually enter the field name. Format : [field name]. You can also enter "[" by using an English IME, and then select the target field from the list, or double-click the target field in the left-side Dimensions or Measures list to insert the dimension or measure name to the edit box. Correct SQL statements will be colored in the edit box.

Do not mix Chinese and English punctuation marks when you write expressions , such as quotation marks, commas, and parentheses. These mistakes will cause syntax parsing errors. Typically, only English punctuation marks are allowed in SQL statements. If an error occurs, check whether you have used Chinese punctuation marks in the expression.

After you have added the calculated field, save the dataset before you refresh it.

Currently, you cannot use calculated fields in the expression of another calculated field. If the physical table field that corresponds to the calculated field is deleted, the calculated field becomes invalid.

#### ***4.3.3.4.6.2 Rules for using calculated fields***

This topic describes the rules for using calculated fields.

Non-aggregate calculated fields can be used as dimensions, or as measures after you specify the aggregate method. Aggregate calculated fields can be used as measures only, and cannot be converted to dimensions.

You can set a data type for a calculated field. Currently, you can select Number, Text, or Date/Time.

If you set the data type of a calculated field to Text, and the actual content of the field is text data, then using the SUM or AVG aggregate method will cause an error. The error is caused because these aggregate methods do not support text data.

Similar to dimensions and measurements generated by the original fields in the data source, you can use the calculated dimensions or measures when you set rows, columns, and filters, and select fields on the Data tab page for charts and maps. You can also convert a calculated field from dimension to measure, or from measure to dimension.

#### ***4.3.3.4.6.3 Types of calculated measures***

Calculated measures can be classified into common measures and aggregate measures.

Calculated measures whose expressions exclude aggregate functions are common measures. Calculated measures whose expressions include aggregate functions are aggregate measures.

You can use the count() or count(distinct) function that has dimensions as its parameters to form a deduplicated aggregate measure.

Examples of aggregate measures: Average purchase amount per user  $\text{sum}(\text{purchase amount})/\text{countd}(\text{user ID})$ . Order cost proportion  $\text{sum}(\text{order cost})/\text{sum}(\text{order amount})$ . However,  $\text{avg}(\text{order cost}/\text{order amount})$  is incorrect, and an error will occur while calculating based on the field.

Do not use common measures together with aggregate measures. For example,  $\text{sum}(\text{order cost})/\text{order amount}$  is incorrect.

You can change the aggregate function of common measures. However, you cannot change the aggregate function of aggregate measures. You cannot convert aggregate measures to dimensions.

Aggregate measures support the following aggregate functions: SUM, AVG, MIN, MAX, COUNT, and COUNT distinct.

#### ***4.3.3.4.6.4 Examples of using a calculated field***

Use functions or arithmetic operations in a calculate field. Examples:

- To calculate the total order amount: `sum([order_amt])`
- To calculate the average order amount: `avg([order_amt])`
- To calculate the maximum order amount: `max([order_amt])`
- To calculate the minimum order amount: `min([order_amt])`
- To count the number of customers: `count([customer_name])`
- To count the number of unique customers: `count(distinct [customer_name])`

Use elementary arithmetic operations

`order_cost ([order_amt] – [profit_amt])/100`

Truncate strings

`Substring([customer_name],1,1)`

Classify value ranges of a measure with the CASE statement

- Classify orders into different groups based on the order amount

```
CASE WHEN [order_amt] < 500 THEN 'small order' WHEN [order_amt] >= 500
AND [order_amt] < 2000 Then 'medium order' WHEN [order_amt] >= 2000 AND [
order_amt] < 5000 THEN 'big order' ELSE 'large order' END
```

- Classify dimension members with the CASE WHEN statement. In this example, provinces are classified into a specific region.

```
CASE WHEN [province] in ('Heilongjiang', 'Liaoning', 'Jilin') THEN 'Northeast'
ELSE [province] END
```

- Calculate a measure by using a complex expression: order amount per customer

```
sum([order_amt])/count(distinct[customer_name])
```

- Add a Unix timestamp

```
from_unixtime([order_id] + 1234567890)
```



- Locate different days in a month

`day([report_date])`

Returns a number in the range of 1 to 31.

- Locate different hours in a day

`hour([report_date])`

Returns a number in the range of 0 to 23.

- Calculate the advertisement effectiveness conversion rate

`CASE WHEN sum([Views]) > 0 THEN sum([Conversion times])/sum([Views]) ELSE 0 END`

The following expression is incorrect to calculate the conversion rate: `sum(CASE WHEN [Views] > 0 THEN [Conversion times]/[Views] ELSE 0 END)`. For metrics that indicate rates, you cannot perform the division operation before you perform the sum operation. You must perform the sum operation before the division operation.

#### 4.3.3.4.6.5 Add a calculated field

This topic describes how to add a calculated field.

##### Context

For more information about calculated fields, see [Rules for using calculated fields](#) and [Examples of using a calculated field](#). Calculated fields are classified into calculated dimensions and calculated measures. For more information about measure types, see [Types of calculated measures](#). The following example uses the `company_sales_record` to calculate the average profit of orders.

##### Procedure

1. In the Quick BI console, click Dataset in the left-side navigation pane.
2. Click the `company_sales_record` dataset.

3. On the dataset edit page, find the Measures list and click the plus sign (+). An Add Calculated Field dialog box appears, as shown in *Figure 4-50: Add a calculated field*.

Figure 4-50: Add a calculated field

The screenshot shows the 'Edit Calculated Field (Measures)' dialog box. On the left, a sidebar lists available measures under 'Default' and 'company\_sales\_record'. The main panel has the following sections:

- \*Name:** A text input field with a placeholder 'Enter a name.' and a red border. Below it, a note states: 'The name must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (()), and square brackets ([ ]).'.
- \*Expression:** A large text area for entering the calculated expression. Below it, a note states: 'Enter the open bracket ([) to call up the dimension or measure list and then select a required field.'
- \*Data Type:** Radio buttons for 'String' and 'Number' (selected).
- Format:** A text input field. Below it, a note states: 'The format expression must be 1 to 50 characters in length and can contain letters, numbers, underscores (\_), number signs (#), commas (,), periods (.), and percent signs (%). Example: #,##0.00%'.
- Description:** A text input field.
- Functions (MySQL):** A panel on the right with a search bar and a list of functions: ABS, CEIL, FLOOR, RAND, SIGN, and PI. Each function has a 'Show' link and a 'Usage' note.

At the bottom right are 'Cancel' and 'OK' buttons.

4. Enter a name for the target calculated measure and specify an expression, as shown in *Figure 4-51: Specify a calculated field expression*.

For example, if you want to calculate the average profit of orders, enter an expression to divide the total profit of orders by number of orders.



**Note:**

**You must use an English IME to enter the expression.**

Figure 4-51: Specify a calculated field expression

Edit Calculated Field (Measures)

\*Name

Enter a name.

The name must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (( )), and square brackets ([ ]).

\*Expression

Enter the open bracket ([) to call up the dimension or measure list and then select a required field.

\*Data Type

String

☒ Number

Format

The format expression must be 1 to 50 characters in length and can contain letters, numbers, underscores (\_), number signs (#), commas (,), periods (.), and percent signs (%).  
Example: #,##0.00%

Description

Functions (MySQL)

Search by function name.

ABS

Show

Usage: ABS(x)

CEIL

Show

Usage: CEIL(x)

FLOOR

Show

Usage: FLOOR(x)

RAND

Show

Usage: RAND()

RAND

Show

Usage: RAND(x)

SIGN

Show

Usage: SIGN(x)

PI

Show

Usage: PI()

Cancel

OK

**5. Select a data type.**

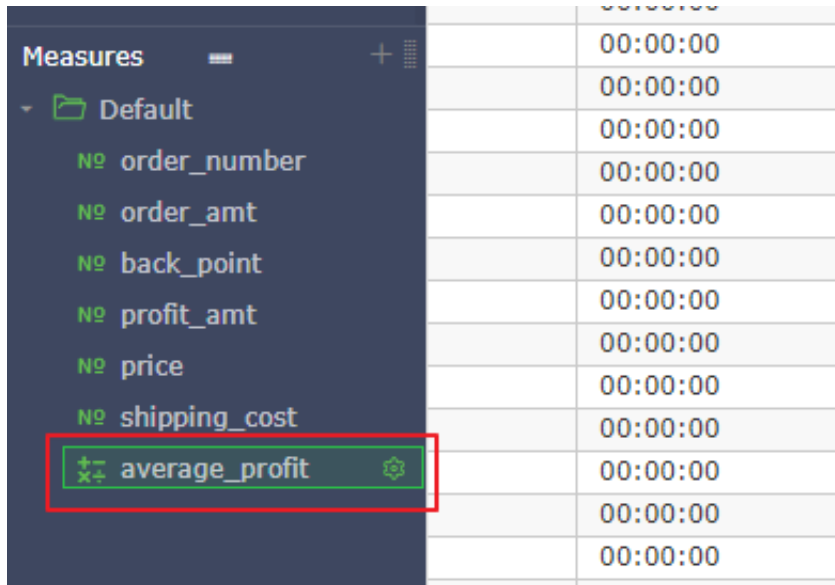
**For example, the average profit is a numeric value. Therefore, you need to select Number as the data type.**

**6. Click OK to add the field.**

**7. Click Save in the upper-right corner to save the dataset.**

8. Click Refresh Preview to view the calculated field, as shown in [Figure 4-52: Refresh the dataset](#).

Figure 4-52: Refresh the dataset



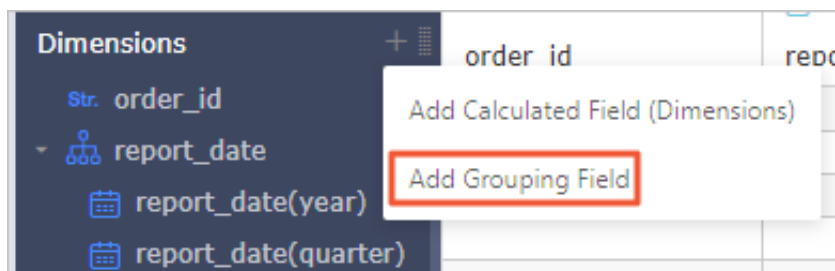
Measures	report_date
order_number	00:00:00
order_amt	00:00:00
back_point	00:00:00
profit_amt	00:00:00
price	00:00:00
shipping_cost	00:00:00
average_profit	00:00:00
	00:00:00
	00:00:00

#### 4.3.3.4.7 Grouping fields

On the dataset edit page, you can add a grouping field to classify data into different groups and add group information.

1. On the Datasets page, click the target dataset.
2. On the dataset edit page, choose + > Add Grouping Field.

Figure 4-53: Add a grouping field



3. In the Edit Grouping Field dialog box, enter the required information and click OK.

Figure 4-54: Edit the grouping field

Field Name:  ⓘ

The field name must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (()), and square brackets ( []).

Grouping Field:  Group By:

Groups +

- Group1
- Ungrouped

Select an item or manually enter an item.

Cancel OK

4. Click Save and then click Refresh Preview. The Dimensions list shows the grouping field.

#### 4.3.3.5 Rename a dataset

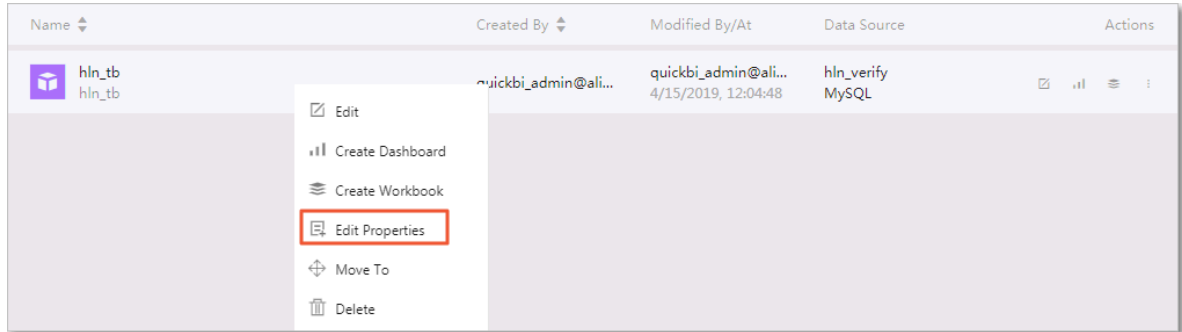
This topic describes how to rename a dataset.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, select the target dataset. Right-click the dataset or click the More icon in the Actions column.

4. Select **Edit Properties** and rename the dataset, as shown in [Figure 4-55: Rename a dataset](#).

Figure 4-55: Rename a dataset



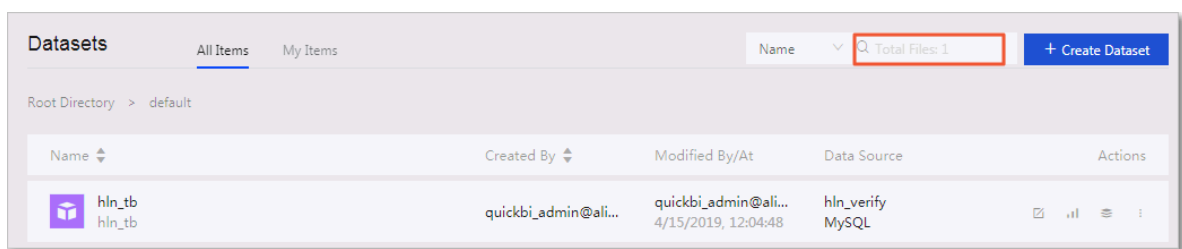
#### 4.3.3.6 Search for a dataset

This topic describes how to search for a specific dataset.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. On the Datasets page, find the search box.
4. Enter a keyword and click the Search icon to search for the target dataset, as shown in [Figure 4-56: Search for a dataset](#).

Figure 4-56: Search for a dataset



#### 4.3.3.7 Transfer a dataset

This topic describes how to transfer a dataset to another user.

##### Context

You can transfer your datasets to other Apsara Stack tenant accounts. After a dataset is transferred to another user, you can no longer view the dataset.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Datasets**.
3. On the **Datasets** page, right-click the target dataset or click the **More** icon in the **Actions** column.
4. Select **Edit Properties**.
5. On the **Edit Properties** tab page that appears, enter the target account and click **Save**, as shown in [Figure 4-57: Transfer a dataset](#).

Figure 4-57: Transfer a dataset

The screenshot shows the 'Edit Properties' dialog box for a dataset. The title bar is 'Edit Properties'. The form contains the following fields and options:

- Name:** A text input field containing 'company\_sales\_record\_en\_us'.
- Owner:** A dropdown menu showing 'quickbi\_admin@aliyun.com' with a downward arrow.
- description:** A large text area with the placeholder text 'Please add an object description'.
- Security Level:** Two radio button options:
  - ☒ Private (Allow Only Workspace Owner to Edit)
  - ☐ Protected (Allow Other Workspace Members to Edit)

#### 4.3.3.8 Create a dataset folder

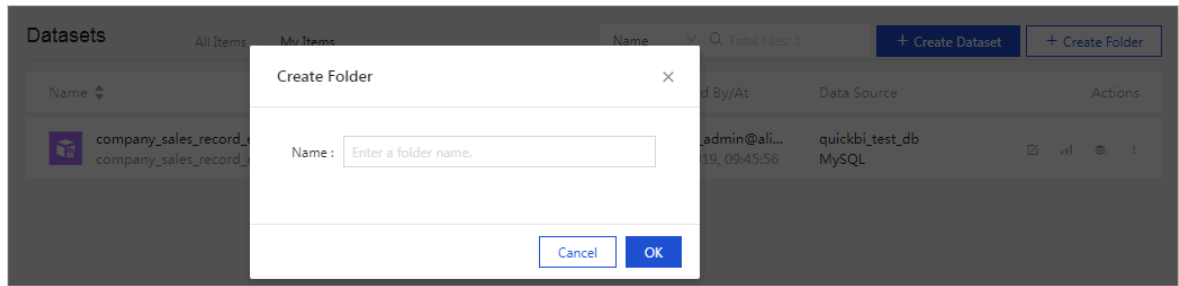
This topic describes how to create dataset folders on the **Datasets** page to manage your datasets.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Datasets**.
3. Click **Create Folder** in the upper-right corner.

4. In the Create Folder dialog box that appears, specify a name for the folder and click OK, as shown in [Figure 4-58: Create a dataset folder](#).

Figure 4-58: Create a dataset folder



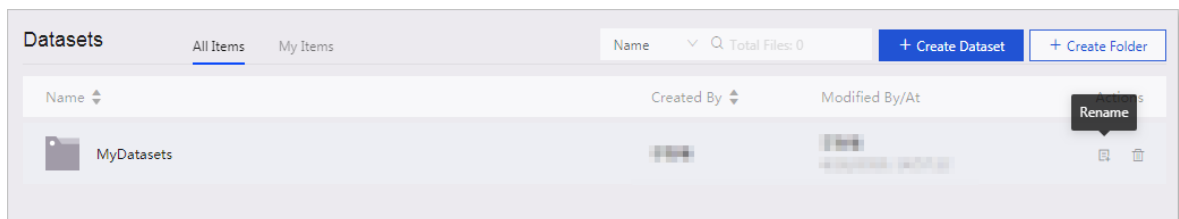
#### 4.3.3.9 Rename a dataset folder

This topic describes how to rename a dataset folder.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the target folder.
4. Right-click the folder and select Rename, as shown in [Figure 4-59: Rename a dataset folder](#).

Figure 4-59: Rename a dataset folder



5. Enter a name and click OK.

#### 4.3.3.10 Delete a dataset

This topic describes how to delete a dataset and issues that may occur during this process.

##### Context

If workbooks have been created based on the dataset, a notification occurs when you attempt to delete the dataset, as shown in [Figure 4-60: Notification](#).



After you delete a dataset that has dashboards created based on it, an error occurs when you access the dashboard, as shown in [Figure 4-61: Error message](#).

Figure 4-60: Notification

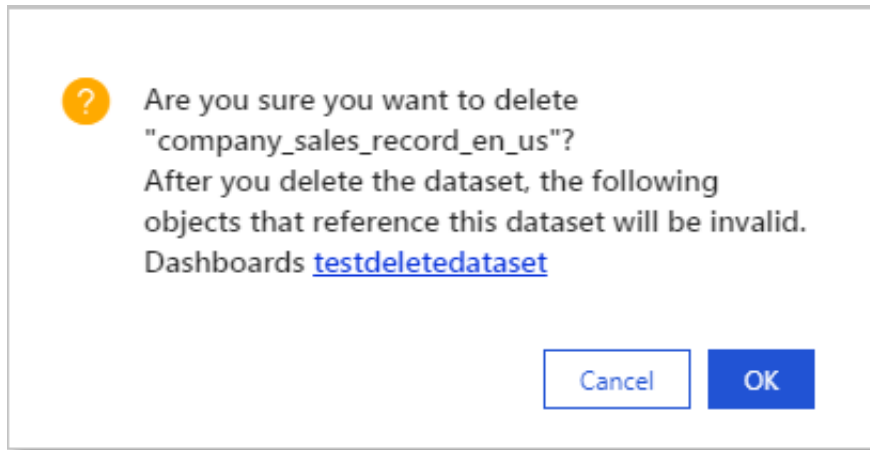
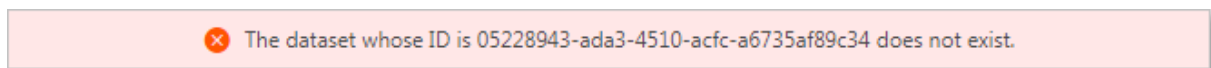


Figure 4-61: Error message

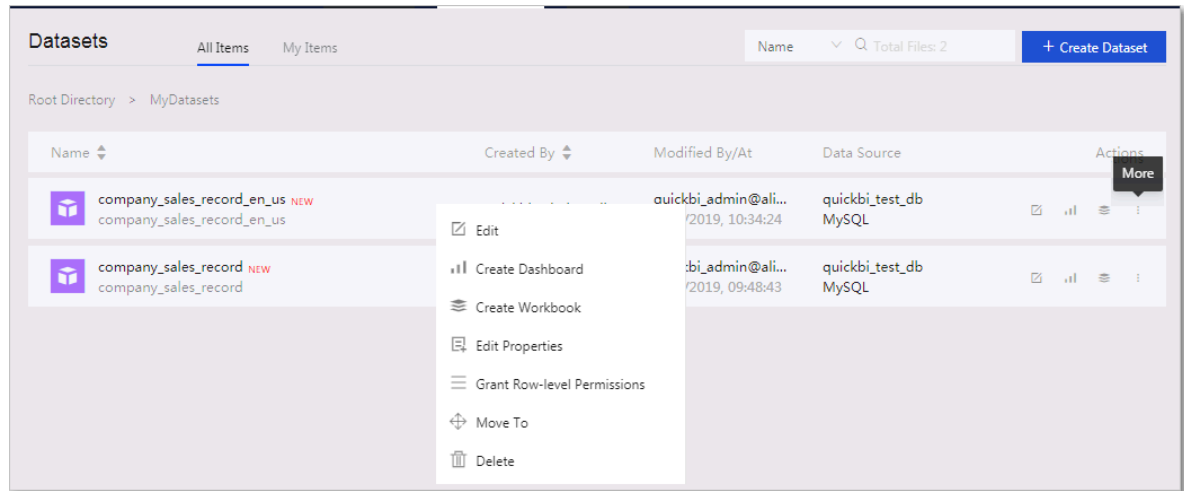


## Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, select the target dataset.

4. Right-click the dataset and select **Delete**, or click the **More** icon and select **Delete** to delete the dataset, as shown in [Figure 4-62: Delete Datasets](#).

Figure 4-62: Delete Datasets



#### 4.3.3.11 Set row-level permissions

For more information about configuring row-level permissions, see [Row-level permission management](#).

## 4.4 Dashboards

### 4.4.1 Dashboard overview

#### 4.4.1.1 Dashboard features

This topic describes the features of a dashboard.

Quick BI provides widgets that you can drag and drop to create reports for various products.

Quick BI also provides a variety of dashboard components. You can select components when you create charts. You can also select the Standard or Full Screen mode when you create a dashboard.

Dashboards use a more flexible tile layout to show interactions between report data. A dashboard not only visualizes data but also allows you to filter and query data and use multiple data display modes to highlight the key fields of data.

You can drag, drop, and click fields in a dashboard to display data more clearly.

You can follow the instructions on the pages to analyze data. This improves user experience.

You can query dynamic data on the dashboard edit page. This increases data visualization performance.

#### 4.4.1.2 Chart types and scenarios

You need to use the corresponding charts to display different types of data. Quick BI currently supports 32 types of charts, including line charts, vertical bar charts, bubble maps, and funnel charts.

For more information about creating charts, see [Create a dashboard](#).

The following table lists the analysis types and scenarios for each chart.

Table 4-1: Chart types and scenarios

Analysis type	Description	Sample scenario	Applicable chart
Comparison	Compares the differences between values, or compares the measures based on the dimensions.	Compares the sales and income differences between different countries or regions.	Vertical bar charts , radar charts, funnel charts, cross table charts , polar diagrams , tornado-leaned funnel charts, and word clouds
Percentage	Displays the percentage of a part or the ratio of a value to the whole.	Displays the sales of the salesperson who has the greatest percentage of total sales.	Pie charts, funnel charts, gauges, and treemaps.
Relation	Displays the relation between values, or between measures.	You can view the relation between two measures and learn about the influence of the first measure on the second measure.	Scatter charts, treemaps, kanban , hierarchy charts , and flow analysis charts.

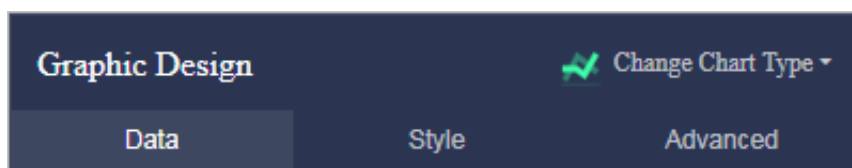
Analysis type	Description	Sample scenario	Applicable chart
Trend	Displays the trend of a value, especially trend changes by time , for example, by year, month, or day , or the progress of data indicators and other possible patterns.	You can view trends in sales or revenue for a product over a period of time.	Line charts
Geographic maps	Displays the size and distribution of data indicators of a country or region on a map. The datasets that you use must include geographic data.	You can view income information about different regions in a country.	Geo maps and geo bubble maps

#### 4.4.1.3 Data elements of a chart

This topic describes the data elements of a chart.

Each chart has three tabs: Data, Style, and Advanced, as shown in [Figure 4-63: Chart tabs](#).

Figure 4-63: Chart tabs



- On the Data tab, you can specify the rows and columns to be displayed in the chart.
- On the Style tab, you can set the chart layout and items to be displayed in the chart.
- On the Advanced tab, you can configure the filter interaction feature to dynamically compare and display data.

Each chart has its unique core data elements. For example, in a geo map, a latitude field is required. Otherwise, data cannot be displayed.

The following table lists the core data elements of each type of chart.

Table 4-2: Data elements of data charts

Chart type	Required data element	Data element description
Line charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Stacked line charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Area charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Stacked area charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
100% stacked area charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Vertical bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.

Chart type	Required data element	Data element description
Stacked vertical bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
100% stacked vertical bar chart	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Circular bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Horizontal bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Stacked horizontal bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
100% stacked horizontal bar charts	Category axis and value axis	The category axis must have at least one dimension. The value axis must have at least one measure.
Pie charts	Slice label and central angle	You can add one dimension to indicate slice labels and one measure to indicate central angles.

Chart type	Required data element	Data element description
Geo bubble maps	Geo location and the bubble size	You can add one dimension to indicate geographic locations. You can add up to five measures to indicate bubble sizes.
Colored maps	Geo location and colorscale	You can add one dimension to indicate geographic locations. You must add at least one measure. You can add up to five measures to indicate the colorscale.
Geo maps	Geo location and colorscale	You can add one dimension to indicate geographic locations. The dimension must consist of geographic information. You can add at least one measure and at most five measures to indicate the colorscale.
Geo bubble maps	Geo location and colorscale	You can add one dimension to indicate geographic locations. The dimension must consist of geographic information. You can add one measure to indicate the colorscale.
Cross tables	Rows and columns	Add dimensions to indicate rows. The number of dimensions that you can add is unlimited. Add measures to indicate columns. The number of measures that you can add is unlimited.

Chart type	Required data element	Data element description
Pivot tables	Rows and values	Add dimensions to indicate rows. The number of dimensions that you can add is unlimited. Add measures to indicate values. The number of measures that you can add is unlimited.
Gauges	Pointer angle	You can add one measure to indicate the pointer angle.
Progress bar charts	Pointer	You can add at least one measure and at most five measures to indicate the progress.
Radar charts	Label and label length	You can add one or two dimensions to indicate labels. You must add at least one measure to indicate the label length.
Scatter charts	Color legend, X axis, and Y axis	The color legend can have one dimension only, and the dimension can contain up to 1,000 members. The X axis must have at least one measure and can have up to three measures. The Y axis must have one measure only.
Bubble charts	X axis, Y axis, and bubble size	The X axis must have at least one measure. The Y axis must have one measure only. You can specify one measure only to indicate the bubble size.
Funnel charts	Tier labels and tier areas	You can add one dimension to indicate tier labels and one measure to indicate tier areas.



Chart type	Required data element	Data element description
<b>Kanban</b>	<b>Labels and metrics</b>	<b>You can add one dimension to indicate labels. You must add at least one measure. You can add up to ten measures to indicate metrics.</b>
<b>Treemaps</b>	<b>Rectangle labels and rectangle sizes</b>	<b>You can add one dimension to indicate rectangle labels and one measure to indicate rectangle sizes.</b>
<b>Polar diagrams</b>	<b>Arc radius and labels</b>	<b>You can add one dimension to indicate labels and one measure to indicate the radius of an arc.</b>
<b>Word clouds</b>	<b>Word size and word</b>	<b>You can add one dimension to indicate word sizes and one measure to indicate words .</b>
<b>Tornado-leaned funnel charts</b>	<b>Metrics for measures and dimensions</b>	<b>Add one dimension and one measure for comparing data.</b>
<b>Hierarchy charts</b>	<b>Node label and node metric</b>	<b>The node label must have at least two dimensions . The node metric must have at least one dimension.</b>
<b>Flow analysis charts</b>	<b>Previous page, current page, next page, previous page PV, previous page UV , current page PV, current page UV, next page PV, next page UV, conversion rate, and bounce rate.</b>	<b>Add one dimension or measure for each of the element as required.</b>

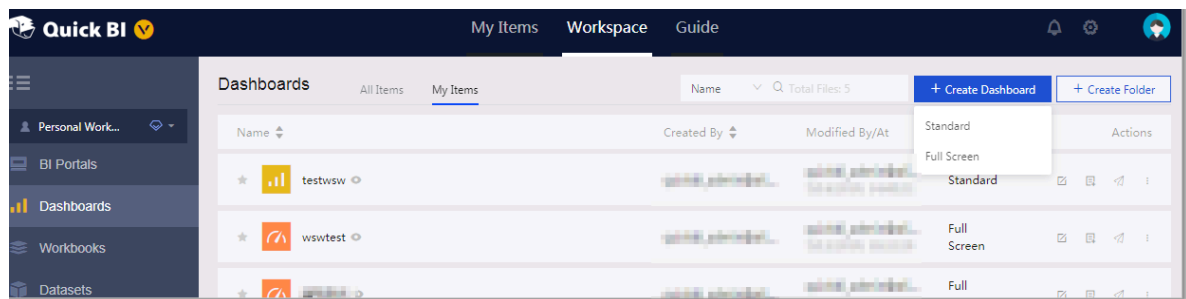
## 4.4.2 Access a dashboard

This topic describes how to access a dashboard.

### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Dashboards** to go to the Dashboards page.
3. Choose **Create dashboards > Standard** as shown in [Figure 4-64: Create a dashboard](#).

Figure 4-64: Create a dashboard



## 4.4.3 Areas of a dashboard

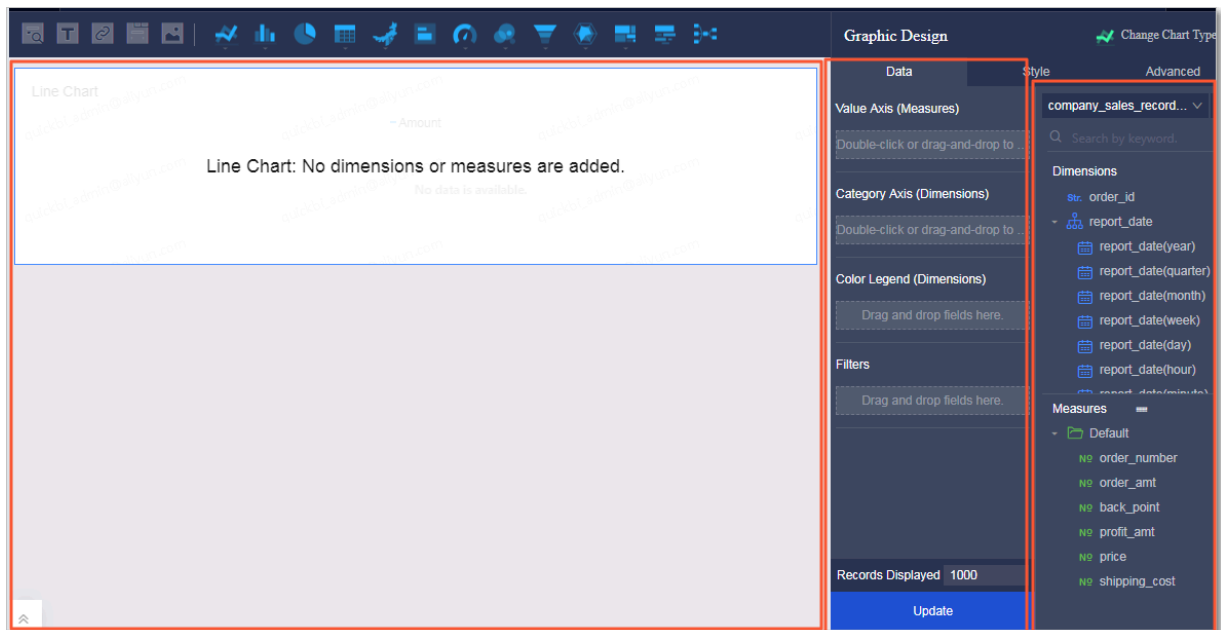
### 4.4.3.1 Overview

The dashboard edit page contains three areas, as shown in [Figure 4-65: Dashboard](#).

- Dataset selection area
- Dashboard configuration area

- **Dashboard display area**

Figure 4-65: Dashboard



- **Dataset selection area:** In this area, you can switch the current dataset to another dataset. The fields of each dataset are displayed in the Dimensions and Measures lists based on the data types preset in the system. You can select dimensions and measures based on the data elements in the chart.
- **Dashboard configuration area:** In this area, you can select chart types and edit the title, layout, and legends of a chart as needed. On the Advanced tab page, you can associate the current chart with other charts and display analysis results from multiple perspectives. You can also filter data by using filters, or inserting a filter bar widget to query key data in a chart.
- **Dashboard display area:** In this area, you can drag charts to adjust their positions, and change chart types. For example, you can change a bar chart to a bubble map. Based on the elements of different charts, the system displays information about missing or error elements. In the dashboard display area, you can save, preview, and create a dashboard. The dashboard provides instructions to help you learn how to create dashboards.

### 4.4.3.2 Dataset selection area

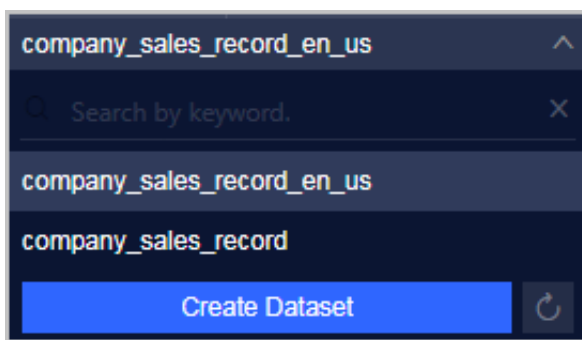
#### 4.4.3.2.1 Switch datasets

If you cannot find the target dataset in the drop-down list, navigate to the Datasets page and check whether the dataset exists. For more information about creating datasets, see [Create a dataset](#).

#### Procedure

1. [Go to the target dashboard](#).
2. On the Data tab page, click the Switch Datasets icon.
3. Select the target dataset from the drop-down list that appears, as shown in [Figure 4-66: Switch to another dataset](#).

Figure 4-66: Switch to another dataset



#### 4.4.3.2.2 Search for a dimension or measure

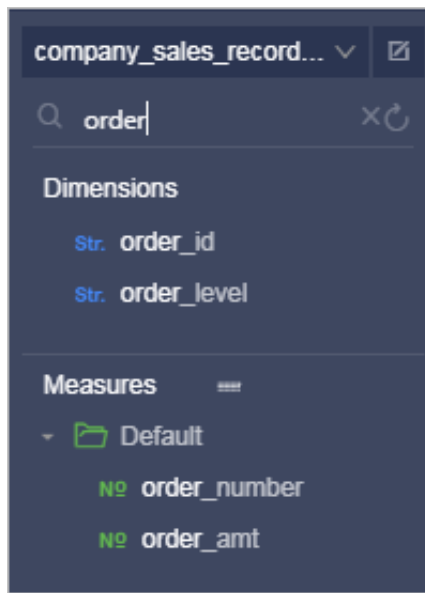
This topic describes how to search for a specific dimension or measure.

#### Procedure

1. [Go to the target dashboard](#).
2. Enter a keyword of a field, for example, order, in the search box.

3. Click the Search icon, as shown in [Figure 4-67: Search for a dimension or measure](#).

Figure 4-67: Search for a dimension or measure



For more information about editing dimensions and measures, see [Edit a dimension](#) and [Edit a measure](#).

### 4.4.3.3 Dashboard graphic design area

#### 4.4.3.3.1 Select fields

This topic describes how to select fields.

#### Context

Before you create a chart, make sure that you have selected and edited the target dataset. For more information about editing a dataset, see [Edit a dataset](#).

#### Procedure

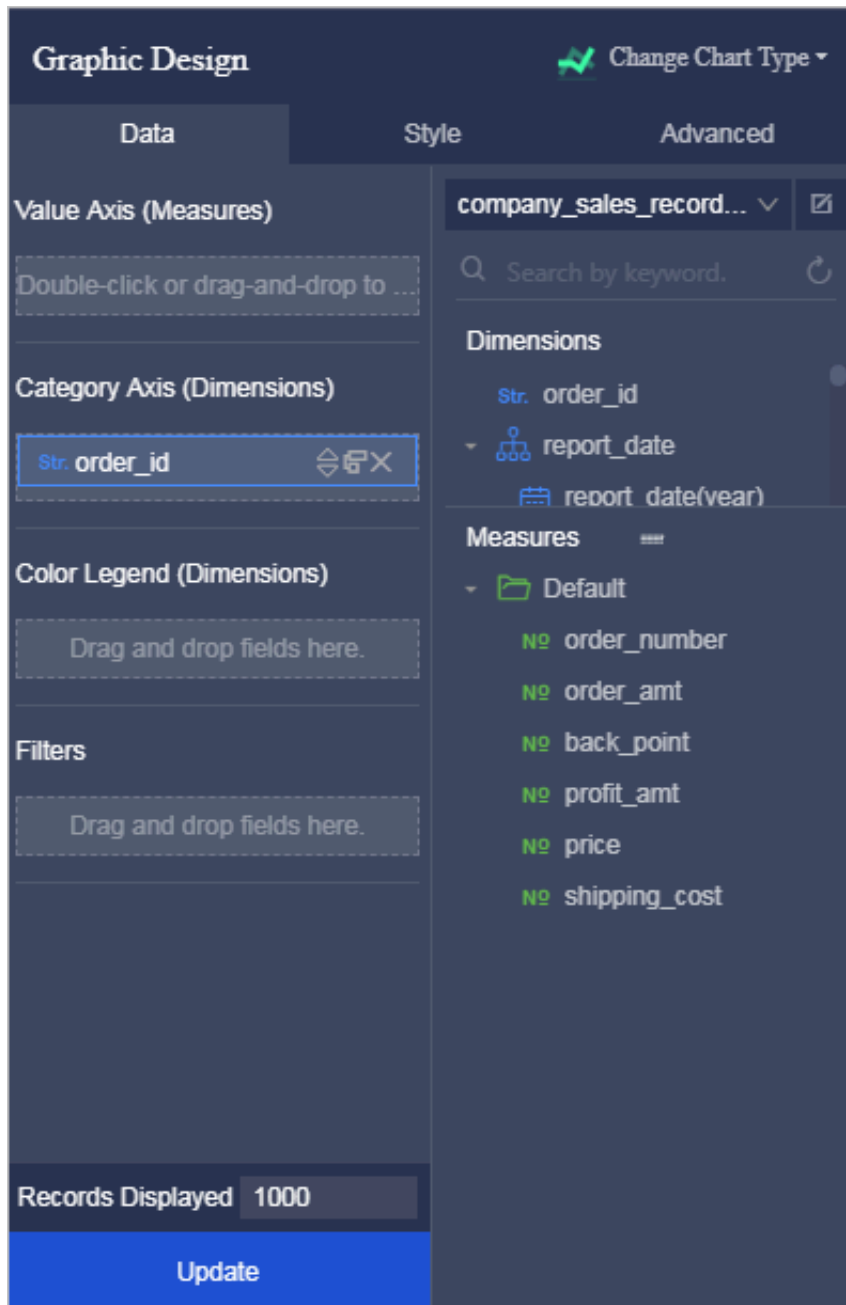
1. [Go to the target dashboard](#).
2. Select a chart from the toolbar on the top of the dashboard.
3. Click the chart icon to create a chart. The chart is displayed in the display area of the dashboard.

To change the chart type, click Change Chart Type in the Graphic Design area, and select another chart type.

4. On the Data tab page, select the required field, as shown in [Figure 4-68: Select fields](#).

Double-click a field to add it to the Rows or Columns area, or you can drag a field to the target area.

Figure 4-68: Select fields



- To delete a field, click the Delete icon next to the field.
- To sort values of a field, click the ascending or descending icon after the field.

5. Click Update. The system then creates a chart.

### 4.4.3.3.2 Color legend

This topic describes how to use the color legend.

#### Context

The color legend function displays the values of the selected field in different colors in a chart.

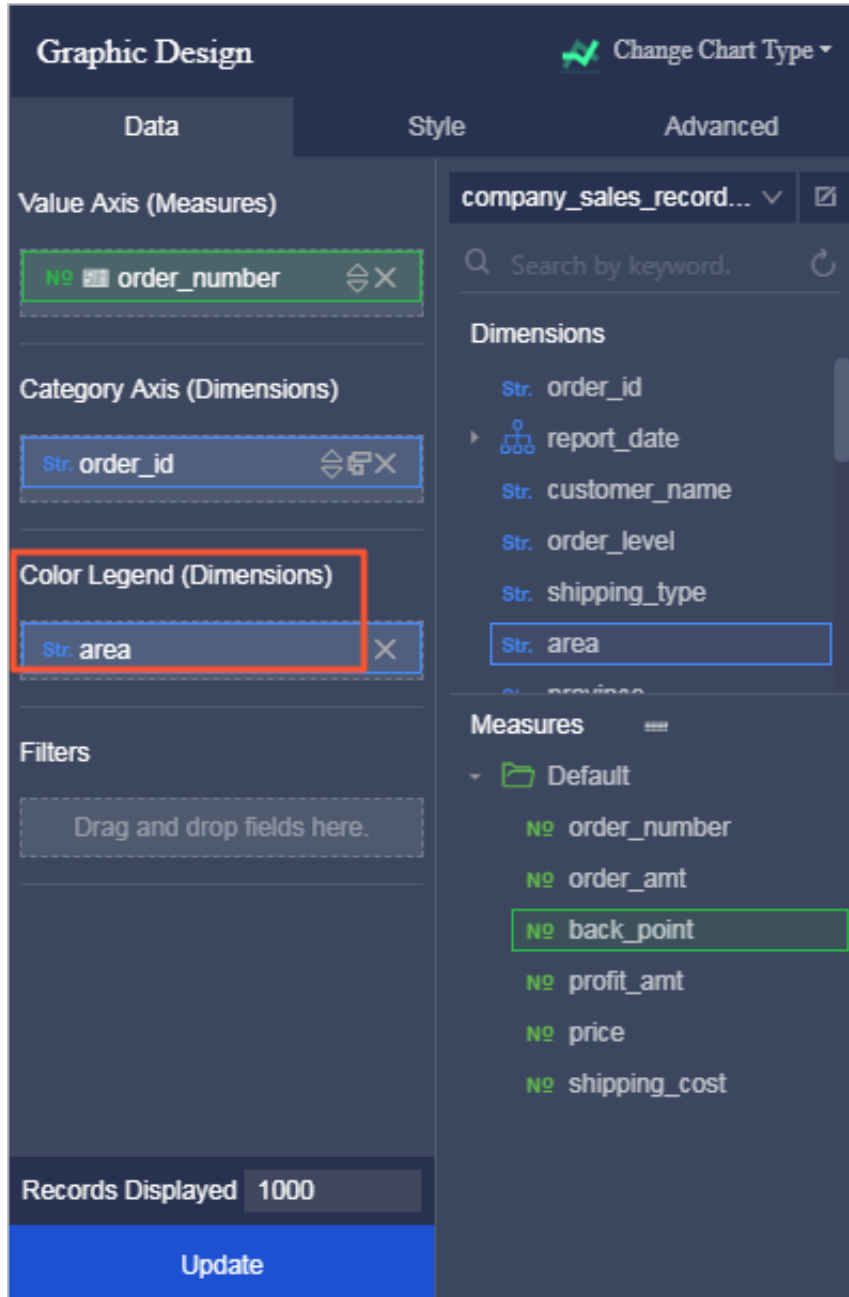
Only dimensions can be added to the color legend area.

#### Procedure

1. *Go to the target dashboard.*

2. Drag a dimension, for example, `product_type`, to the Color Legend area, as shown in [Figure 4-69: Color legend](#).

Figure 4-69: Color legend



3. Click Update. The specified fields are displayed in the chart in different colors, as shown in [Figure 4-70: Result](#).

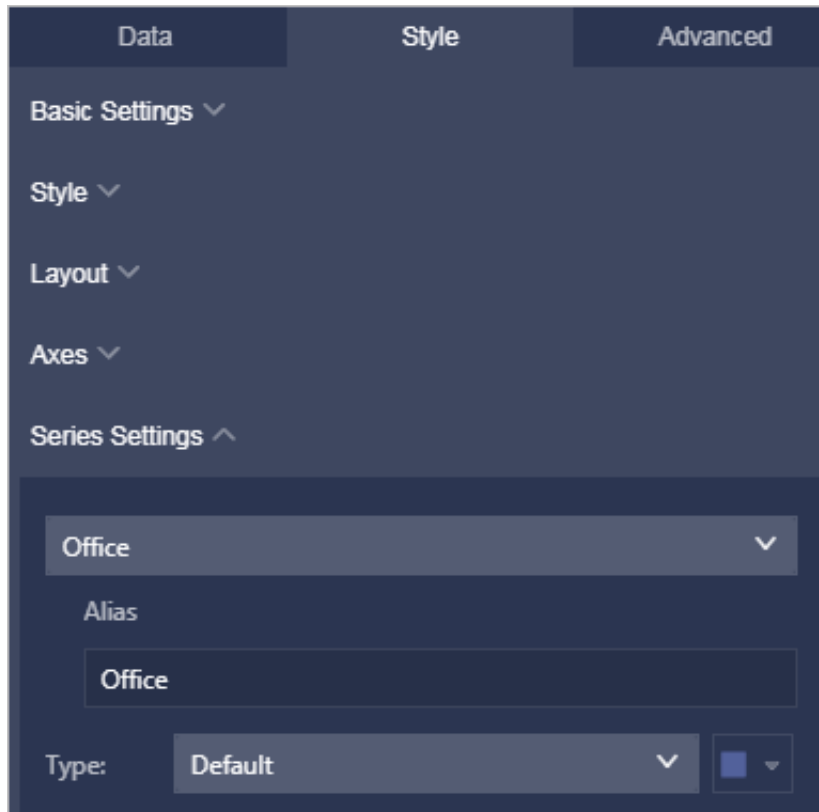
Figure 4-70: Result





4. Click the Series Settings under the Style tab, you can change the color of each specified filed, as shown in [Figure 4-71: Change colors](#).

Figure 4-71: Change colors



#### 4.4.3.3.3 Sorting

This topic describes how to sort data based on the specified dimension and measure on the Data tab page.

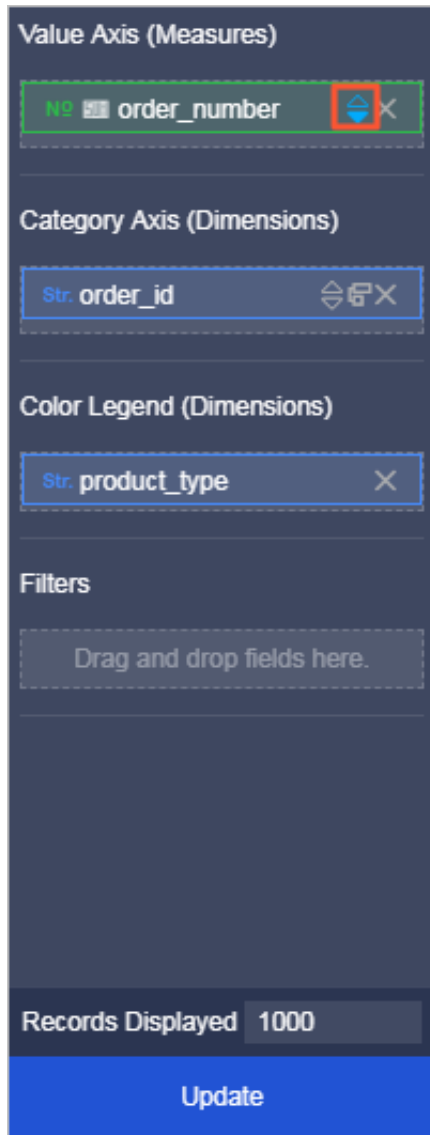
##### Procedure

1. [Go to the target dashboard](#).
2. Select a field, for example, order\_amt.

3. Click the ascending icon after the field, as shown in [Figure 4-72: Set the sorting order](#).

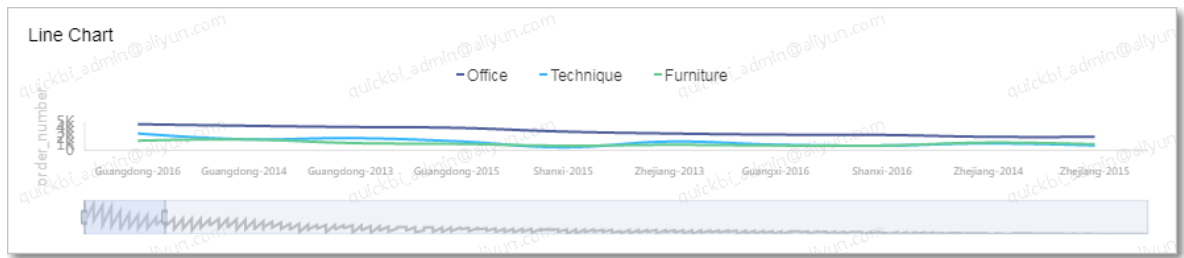
The up arrow represents the ascending order, and the down arrow represents the descending order.

Figure 4-72: Set the sorting order



4. Click Update to update the chart, as shown in [Figure 4-73: Sorting result](#).

Figure 4-73: Sorting result



#### 4.4.3.3.4 Filter by field

This topic describes how to filter data based on specified fields.

##### Context

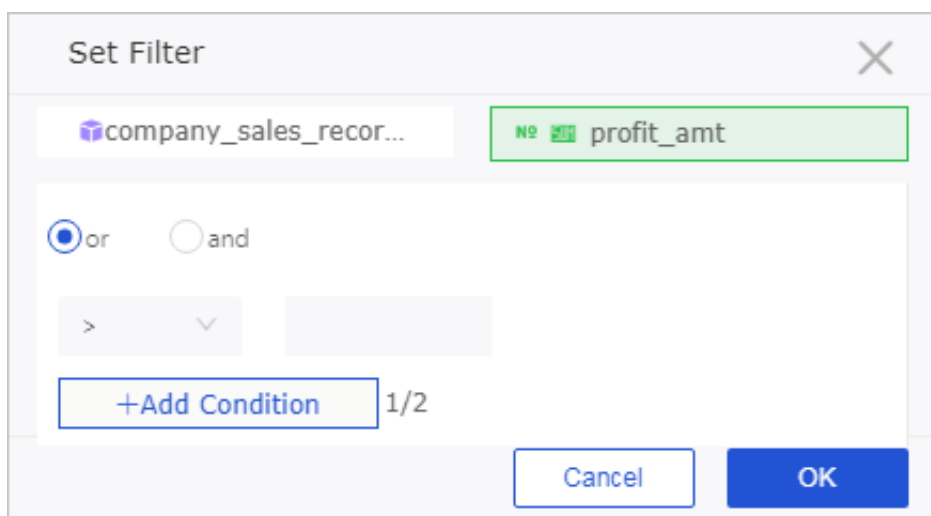
Drag a dimension or measure to the Filters area to specify the fields used to filter data.

In the following example, the filed `profit_amt` is selected.

##### Procedure

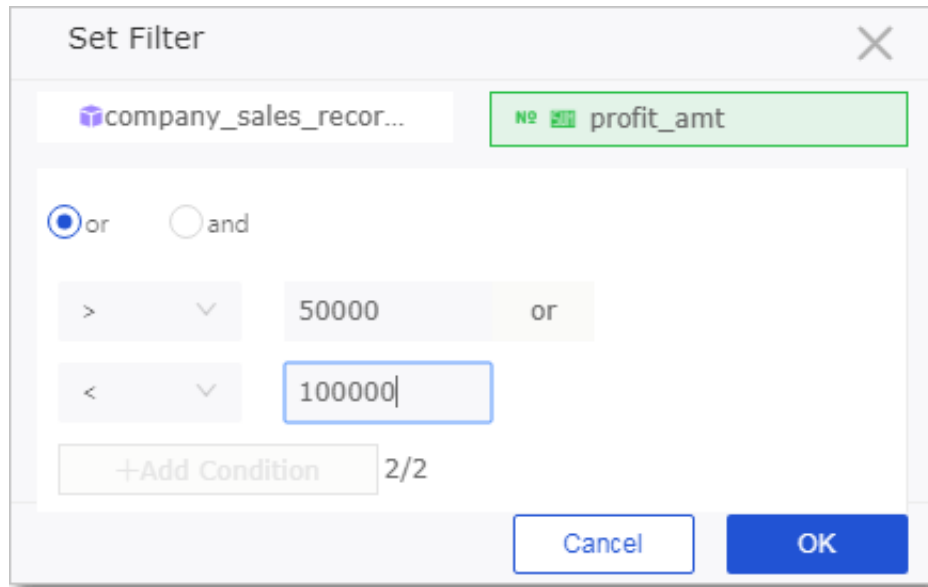
1. [Go to the target dashboard](#).
2. Drag the `profit_amt` field to the Filters area.
3. Click the Filter icon and set the parameters in the Set Filter dialog box, as shown in [Figure 4-74: Set the filter](#).

Figure 4-74: Set the filter



4. Select the filter condition, for example, >, <, or =, as shown in [Figure 4-75: Specify a value range](#).

Figure 4-75: Specify a value range



5. After you set the parameters, click OK.
6. Click Update. The system then updates the chart based on the parameters of the filter.

#### 4.4.3.3.5 Filter interaction

You can use the filter interaction feature when you have created multiple charts on a dashboard. This topic describes how to use the filter interaction feature.

##### Context

You can configure the filter interaction feature on the Advanced tab page.

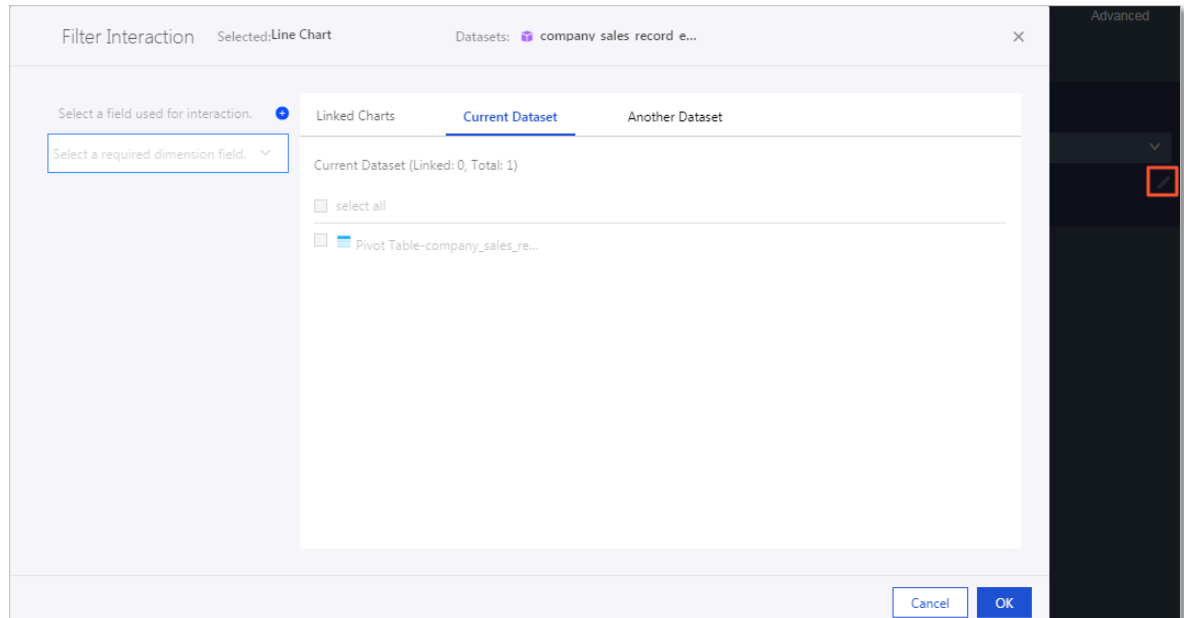
Before you use this feature, make sure that you have created at least two charts in the current dashboard.

##### Procedure

1. [Go to the target dashboard](#).
2. Select a chart, for example, a funnel chart.
3. In the Graphic Design area, click the Advanced tab.

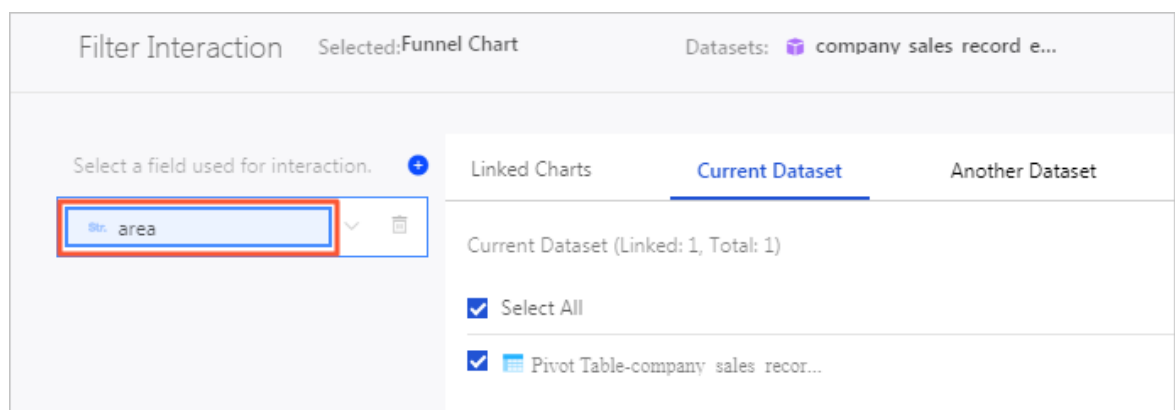
4. On the Advanced tab page, click the Filter Interaction icon. The system then displays all available charts, as shown in *Figure 4-76: The Advanced tab*.

Figure 4-76: The Advanced tab



5. Select fields the same as the filter fields from the available charts to associate these charts, as shown in *Figure 4-77: Filter interaction settings*.

Figure 4-77: Filter interaction settings



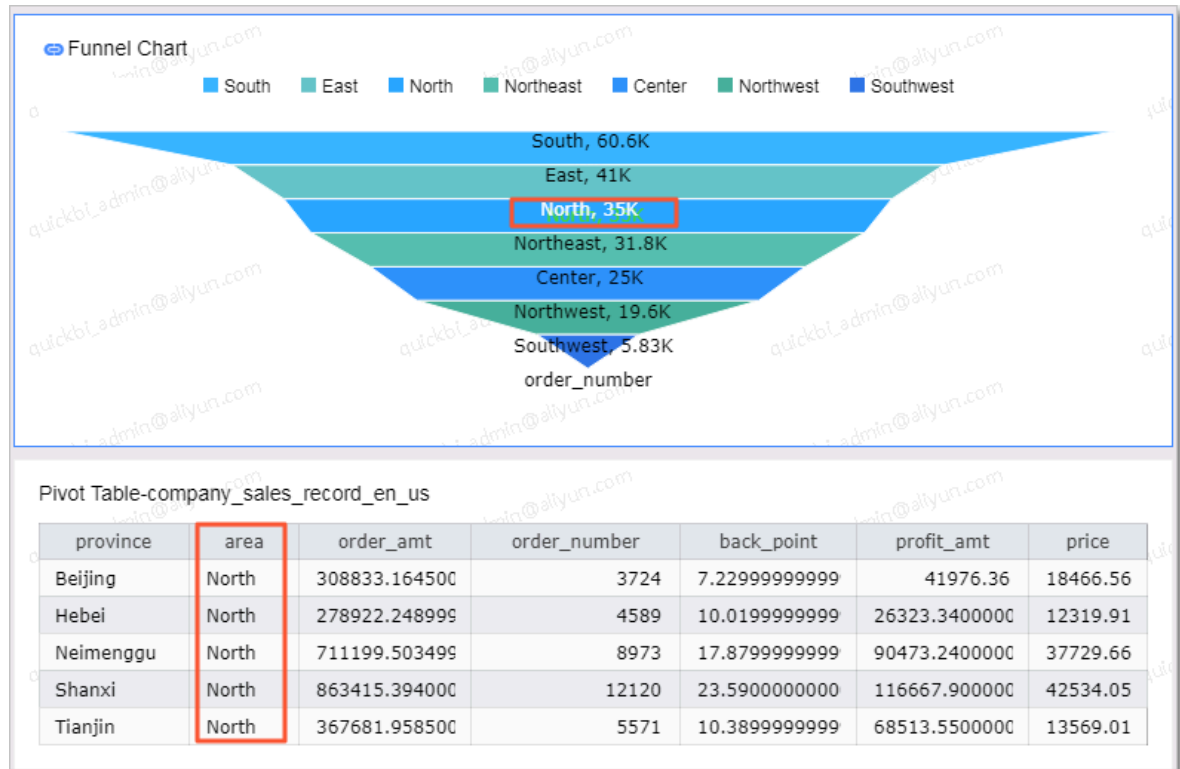
6. In the upper-right corner of the dashboard, click Preview to preview the current dashboard.

Figure 4-78: Click Preview



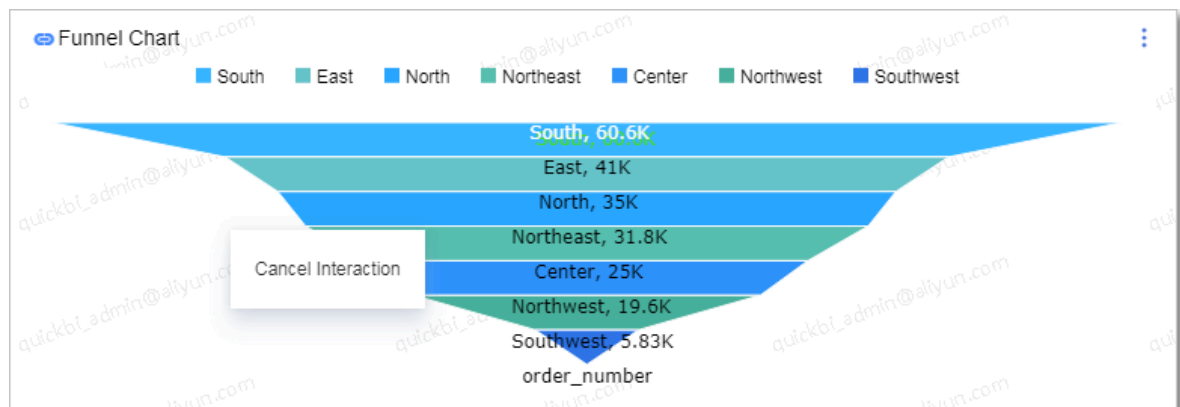
7. Click China North in the funnel chart, and the associated cross tab chart displays data of China North, as shown in *Figure 4-79: Result*.

Figure 4-79: Result



8. Hover over any blank area in the funnel chart, and click the Cancel Interaction notification to disable the filter interaction feature.

Figure 4-80: Click the notification



### 4.4.3.4 Dashboard display area

#### 4.4.3.4.1 Overview

In the display area of a dashboard, you can perform the following operations on one or multiple charts:

- Manage the dashboard
- Adjust chart positions
- View chart data
- Delete a chart

#### 4.4.3.4.2 Toolbar

The toolbar of a dashboard allows you to save, preview, and edit the dashboard, as shown in *Figure 4-81: Dashboard toolbar*.

Figure 4-81: Dashboard toolbar



#### 4.4.3.4.3 Adjust chart position

Multiple charts may be displayed on the same dashboard. In this circumstance, you can drag the charts to adjust their positions.

##### Procedure

1. *Go to the target dashboard.*
2. Select a chart or widget.
3. Drag the chart or widget to the specified position.



##### Note:

You can drag a chart or widget to anywhere within the display area of the dashboard.

#### 4.4.3.4.4 View chart data

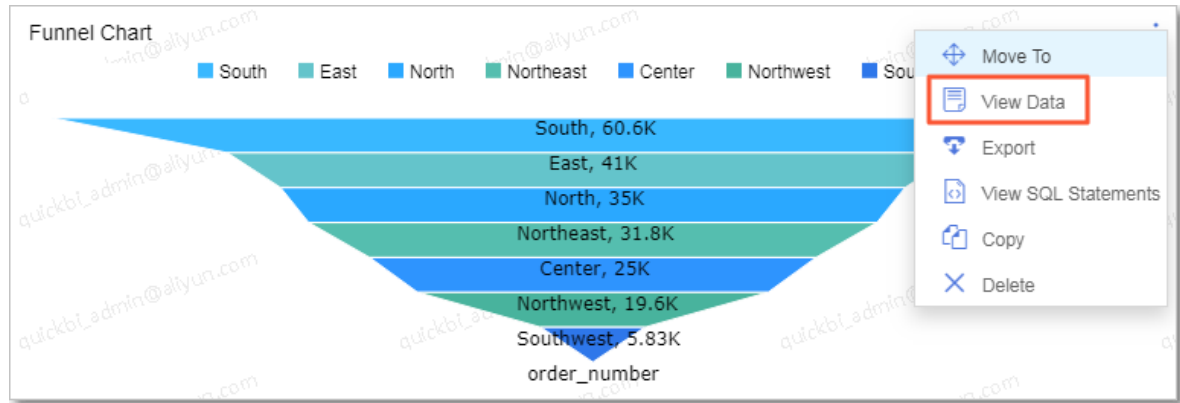
This topic describes how to view chart data.

##### Procedure

1. *Go to the target dashboard.*

2. Select a chart, for example, a funnel chart.
3. Click the More icon in the upper-right corner of the chart.
4. Select View Data, as shown in [Figure 4-82: View chart data](#).

Figure 4-82: View chart data



5. You can also click Export to export chart data to a local device, as shown in [Figure 4-83: Export chart data](#).

Figure 4-83: Export chart data

Figure 4-83 shows a 'View Data' dialog box with a table of chart data. The table has two columns: 'area' and 'order\_number'. The data is as follows:

area	order_number
South	60646.0
East	40954.0
North	34977.0
Northeast	31839.0
Center	25004.0
Northwest	19623.0
Southwest	5828.0

At the bottom right of the dialog box, there are two buttons: 'Export' and 'Cancel'.

#### 4.4.3.4.5 Change chart types

You can select different chart types from the toolbar on the top of the dashboard.

This topic describes how to change chart types.

#### Procedure

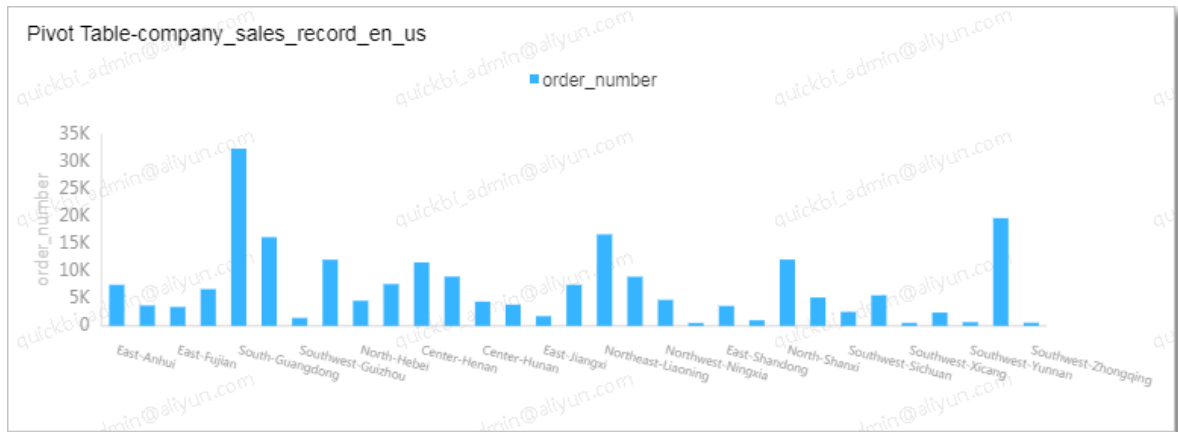


1. *Go to the target dashboard.*
2. Select a chart, for example, a cross table.
3. In the Graphic Design area, click Change Chart Type and select another chart type, for example, a vertical bar chart.

#### 4. Click the vertical bar chart icon to change the chart type.

The system then converts the cross table to a vertical bar chart, as shown in [Figure 4-84: Change the chart type](#).

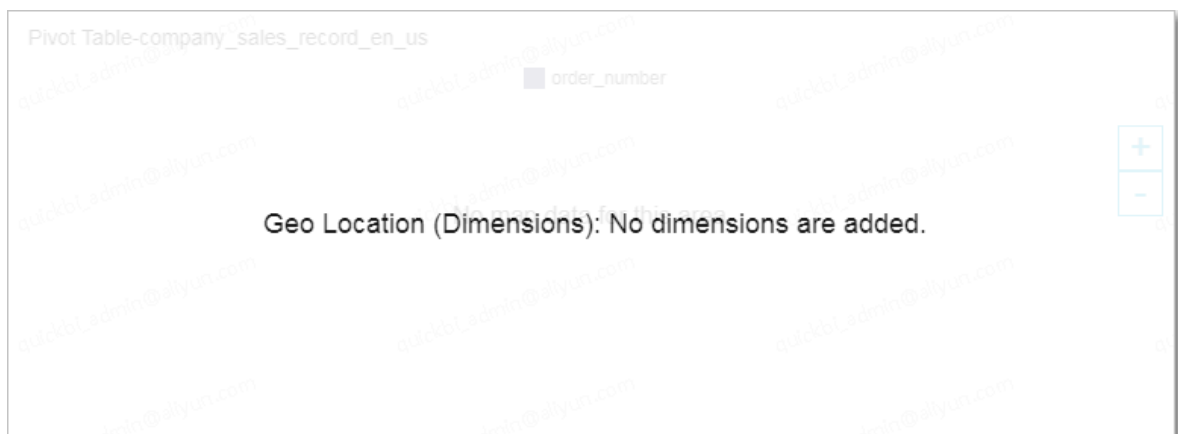
Figure 4-84: Change the chart type



If you fail to change the chart type, it indicates that the elements of the current chart do not match those of the target chart. You need to manually adjust the elements before you change the chart type.

The system provides instructions to help you adjust the elements based on the current and target chart types, as shown in the following figure.

Figure 4-85: System instructions



You can follow the instructions to adjust the dimensions and measures to change the chart type.

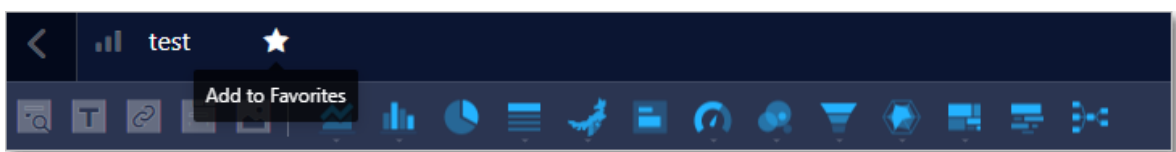
#### 4.4.3.4.6 Add to favorites

You can add a dashboard to the Favorites tab by clicking the Add to Favorites icon on the top of the dashboard.

##### Procedure

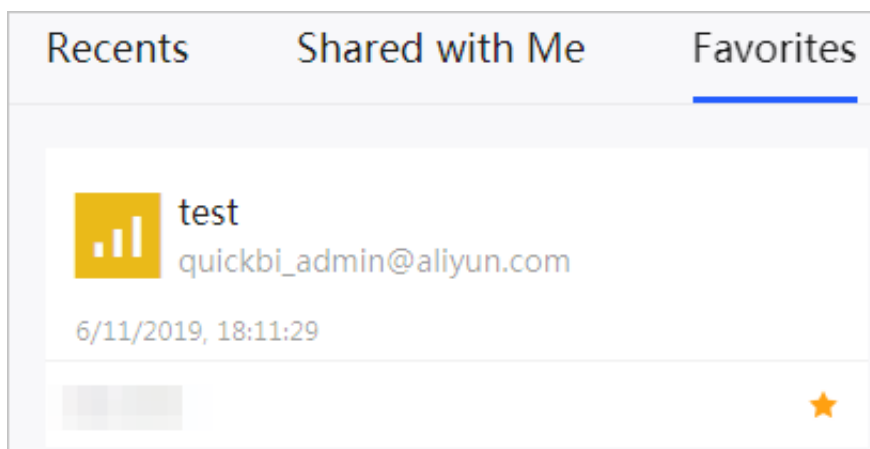
1. *Go to the target dashboard.*
2. On the top of the dashboard, click the Add to Favorites icon, as shown in *Figure 4-86: The Add to Favorites icon.*

Figure 4-86: The Add to Favorites icon



3. On the Quick BI homepage, you can click the Favorites tab to view the dashboards that you have added, as shown in *Figure 4-87: Favorites.*

Figure 4-87: Favorites



#### 4.4.3.4.7 Delete a chart

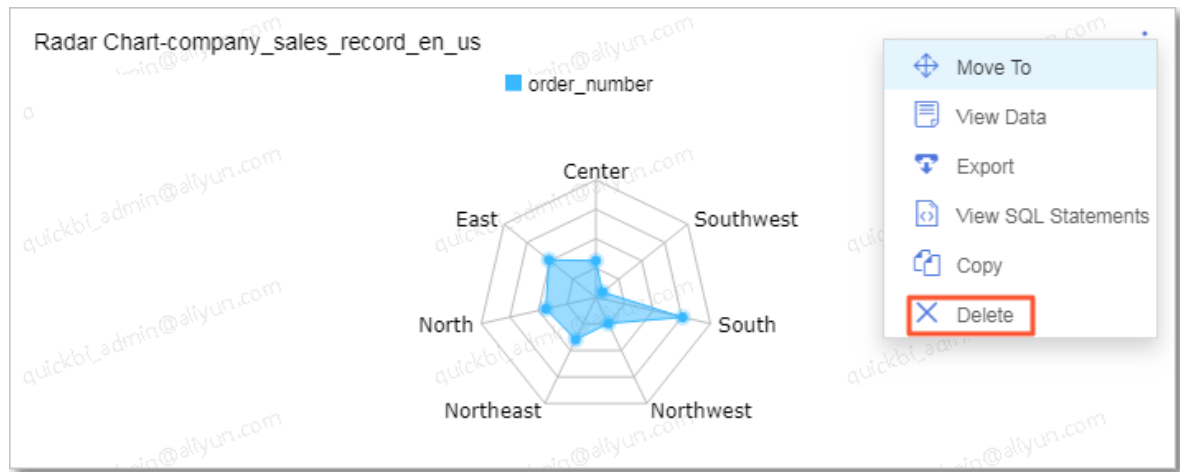
If you no longer need a chart, you can delete it. This topic describes how to delete charts.

##### Procedure

1. *Go to the target dashboard.*
2. Select a chart, for example, a radar chart.
3. Click the More icon in the upper-right corner of the chart.

#### 4. Select Delete, as shown in *Figure 4-88: Delete a chart.*

Figure 4-88: Delete a chart



### 4.4.3.4.8 Widgets

#### 4.4.3.4.8.1 Overview

The display area of a dashboard provides the following five types of widgets:

- Filter bar
- Text area
- IFrame
- Tab
- Image

#### 4.4.3.4.8.2 Filter bar

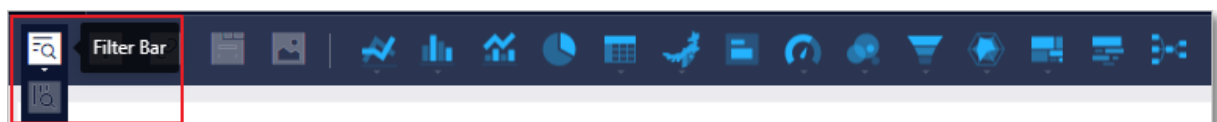
##### 4.4.3.4.8.2.1 Add filter conditions

In a dashboard, you can select filter conditions to query data in one or multiple charts.

#### Context

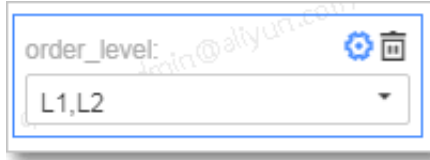
*Go to the target dashboard*, click the Filter Bar icon, and specify filter conditions, as shown in *Figure 4-89: Filter conditions.*

Figure 4-89: Filter conditions



Click the Settings icon and the Set Filter dialog box appears, as shown in [Figure 4-90](#): *The Set Filter dialog box*.

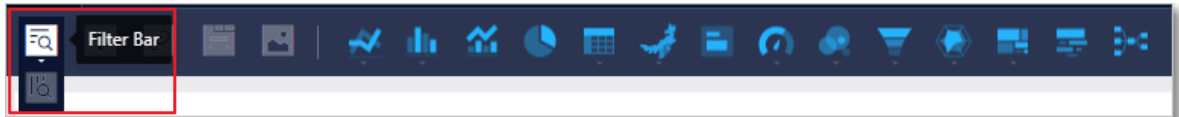
Figure 4-90: The Set Filter dialog box



## Procedure

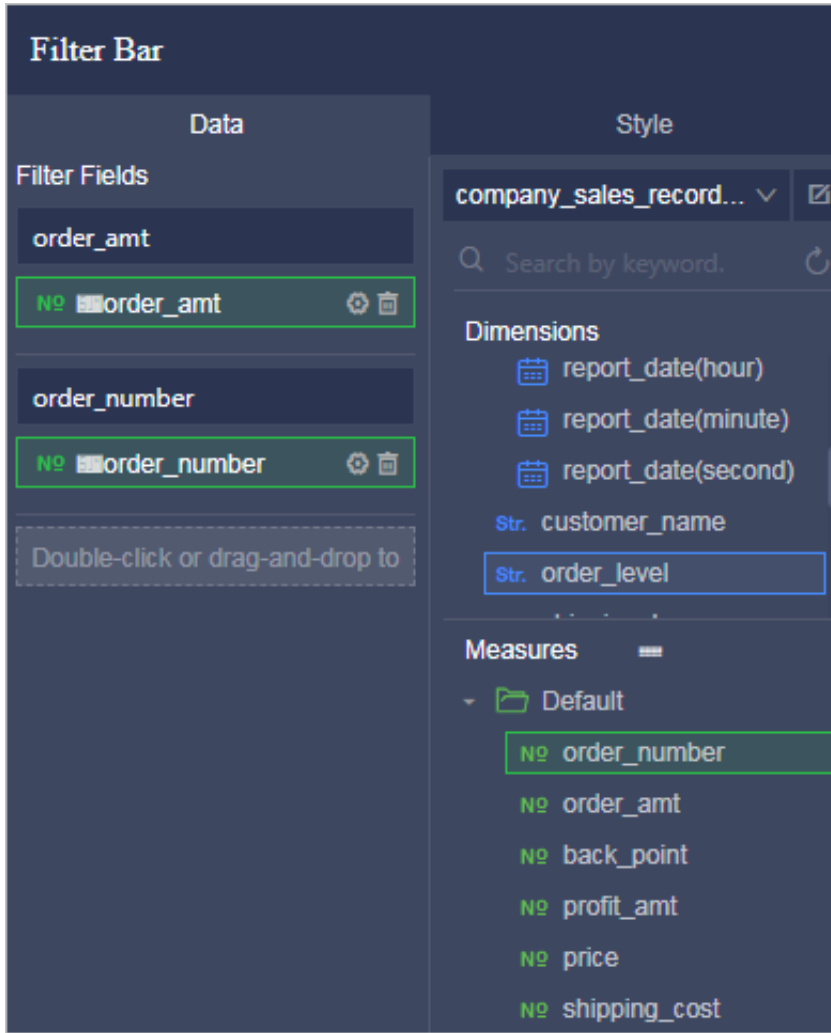
1. Click the Filter Bar icon, as shown in [Figure 4-91](#): *Example*.

Figure 4-91: Example



2. On the Data tab page, select a dataset and filter fields, as shown in [Figure 4-92: Set filter conditions](#).

Figure 4-92: Set filter conditions



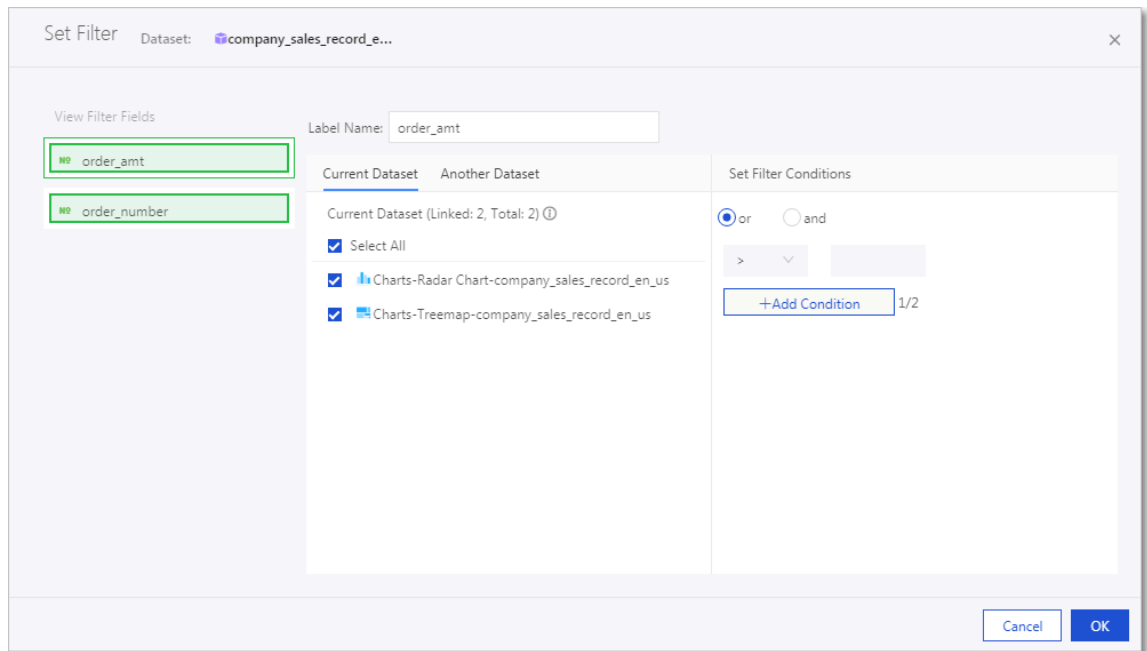
Currently, you can associate the current chart with charts from the same dataset or other datasets.

### 3. Select Current Dataset or Another Dataset.

The following example selects Current Dataset.

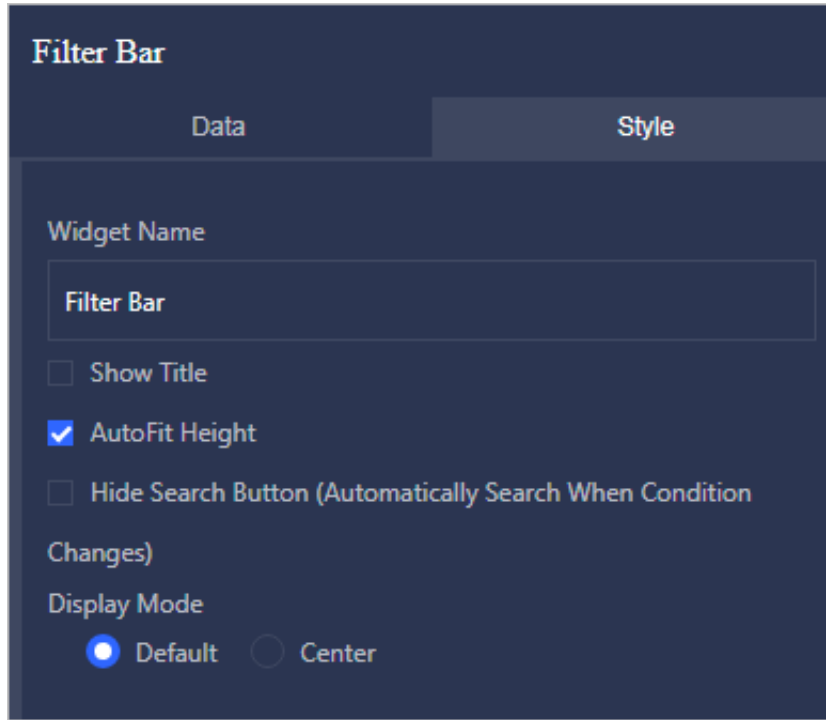
- a) Select Current Dataset and choose a chart type based on the fields that you have selected, as shown in [Figure 4-93: Associate charts from the same dataset](#).

Figure 4-93: Associate charts from the same dataset



- b) On the Style tab page, specify a title for the widget and set the position of the Query button, as shown in [Figure 4-94: Edit filter conditions](#).

Figure 4-94: Edit filter conditions



The following figure [Figure 4-95: Filter conditions](#) shows the Filter Bar widget after you have specified the filter conditions:

Figure 4-95: Filter conditions



- c) Select a field, for example, `order_amt`.  
d) Set the value to 50,000, as shown in [Figure 4-96: Set filter conditions](#).

Figure 4-96: Set filter conditions





- e) Click Query. The charts that the filter conditions apply to are then updated, as shown in *Figure 4-97: Query result*.

In the treemap, products with an order amount less than 50,000 are filtered out.

Figure 4-97: Query result



The following example selects Another Dataset.

- a) Select a dataset. This example associates the order\_level field in charts from different datasets.



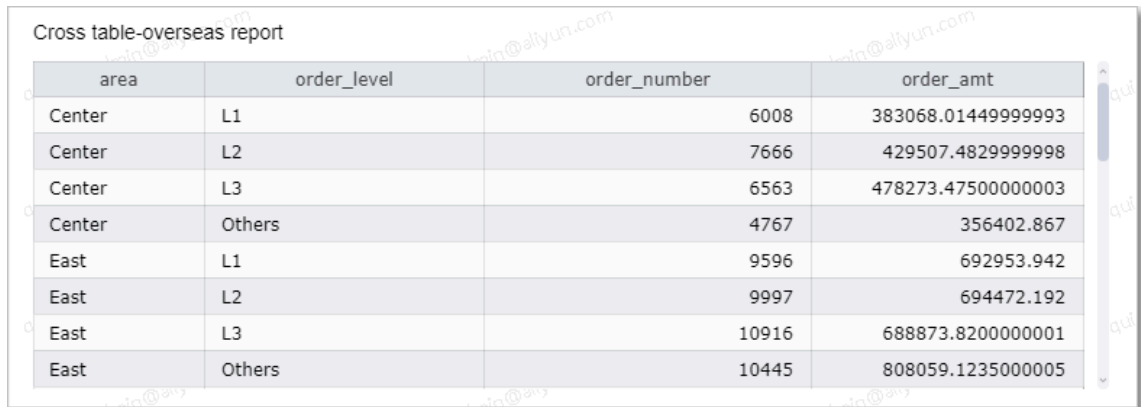
Note:

You can use filter conditions to associate charts from different datasets. Make sure that the filter fields associating charts from different datasets exist in these datasets. Otherwise, you cannot associate the datasets.

- b) Edit dimensions and measures.
- c) Select a chart, for example, a cross table.
- d) Select the target fields as the rows and columns of the cross table.
- e) Click Update to update the cross table.
- f) On the Style tab page, you can change the title and layout of the cross table.

For example, set the title of the cross table chart to Overseas Report, as shown in [Figure 4-98: Overseas Report](#).

Figure 4-98: Overseas Report

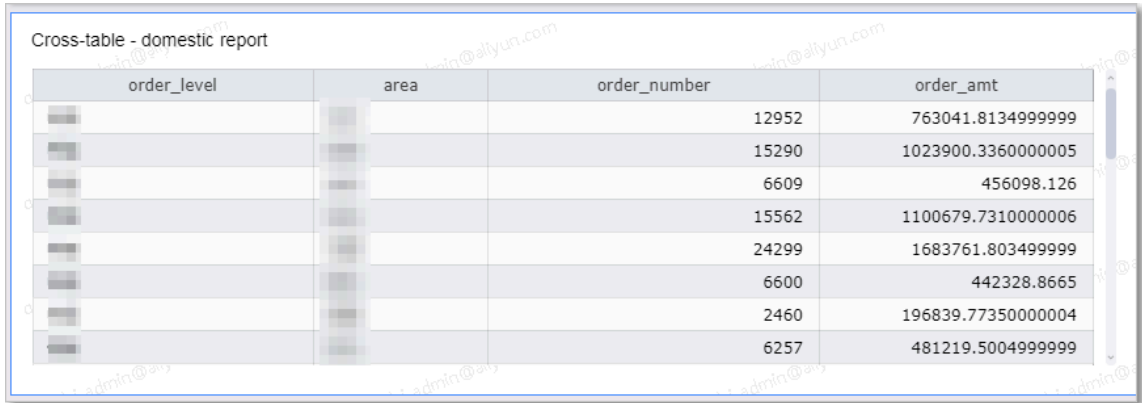


area	order_level	order_number	order_amt
Center	L1	6008	383068.01449999993
Center	L2	7666	429507.48299999998
Center	L3	6563	478273.47500000003
Center	Others	4767	356402.867
East	L1	9596	692953.942
East	L2	9997	694472.192
East	L3	10916	688873.8200000001
East	Others	10445	808059.1235000005

- g) To switch to another dataset, click the Edit Dataset icon.
- h) Edit dimensions and measures.
- i) Select a chart, for example, a cross table.
- j) Select the target fields as the rows and columns of the cross table.
- k) Click Update to update the chart.
- l) On the Style tab page, you can change the title and layout of the cross table.

For example, set the title of the cross chart to Domestic Report, as shown in [Figure 4-99: Domestic Report](#).

Figure 4-99: Domestic Report

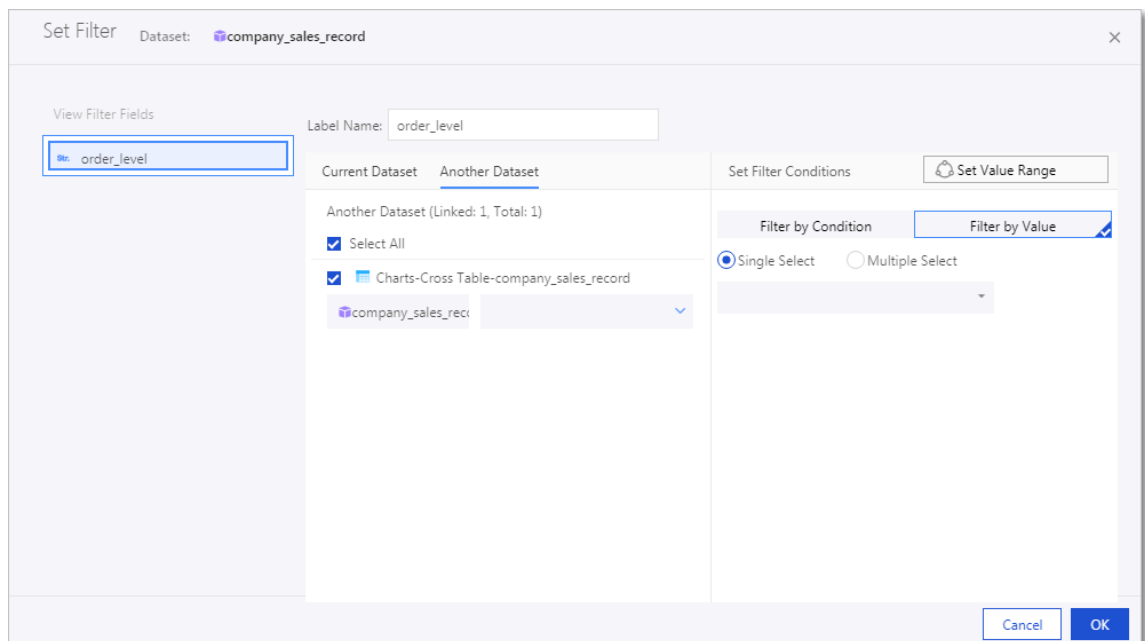


order_level	area	order_number	order_amt
		12952	763041.8134999999
		15290	1023900.3360000005
		6609	456098.126
		15562	1100679.7310000006
		24299	1683761.8034999999
		6600	442328.8665
		2460	196839.77350000004
		6257	481219.5004999999

- m) Click the Filter Bar icon and select a dataset and filter fields.
- n) Select Another Dataset and select the fields that are used to associate the current chart to charts from other datasets.

In the following example, the order\_level field is selected, as shown in [Associate data across datasets](#).

Figure 4-100: Associate data across datasets



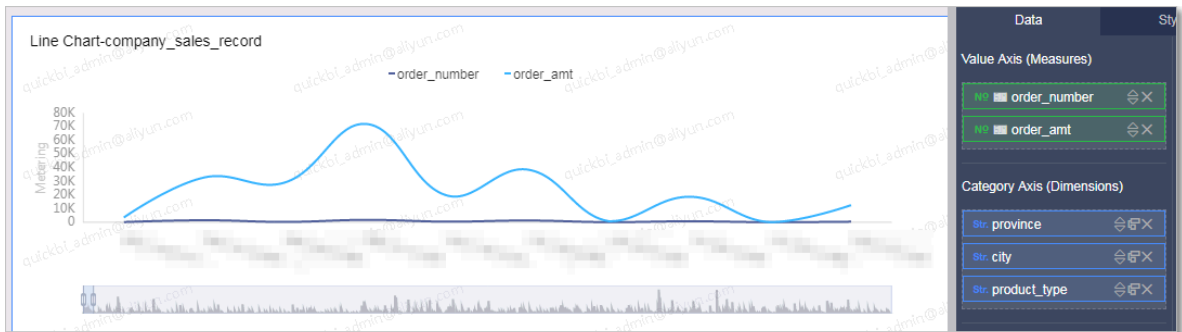
Click OK. You can use the widget to query table data from different datasets.

#### 4.4.3.4.8.2.2 Cascade filter

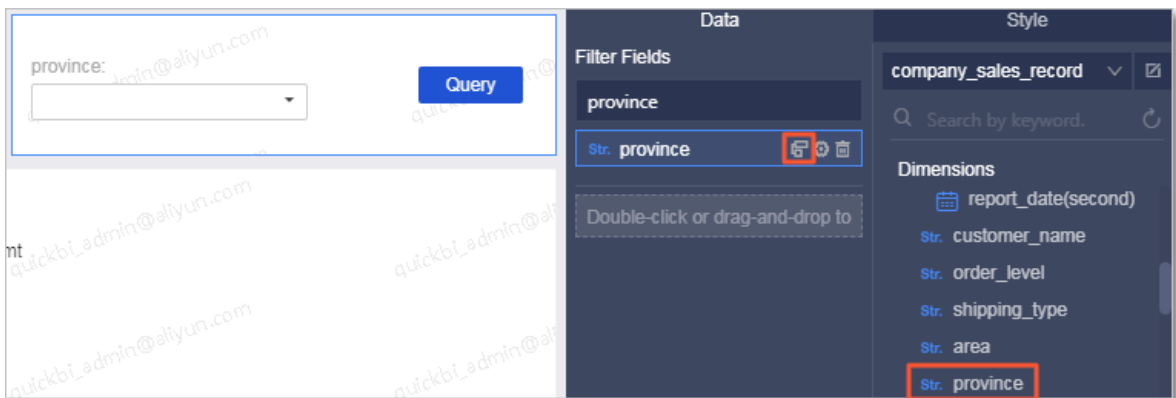
The filter bar widget supports cascade filter, which makes it easier for you to specify multiple filter conditions.

1. Create a line chart in the dashboard, as shown in the following figure.

Figure 4-101: Create a line chart



2. Click the Filter Bar widget, and select a dataset and filter fields. In this example, the province field is selected, as shown in the following figure.



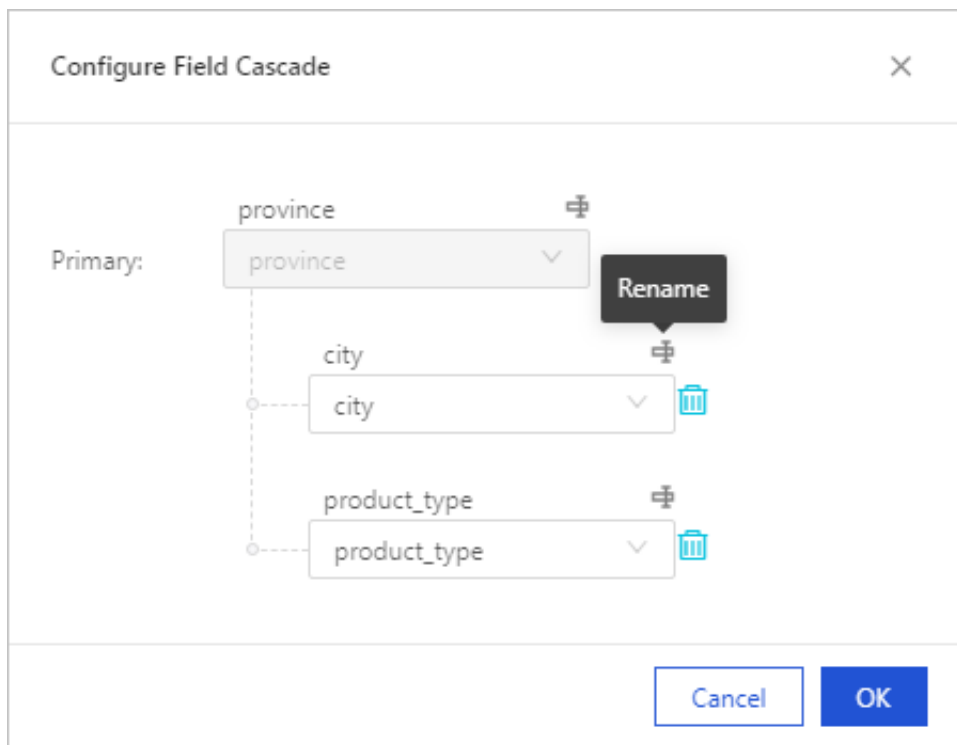
3. Click the cascade filter icon. In the Configure Field Cascade dialog box, click Add Cascade, select fields, and then click OK. In this example, the city and product\_type fields are selected.



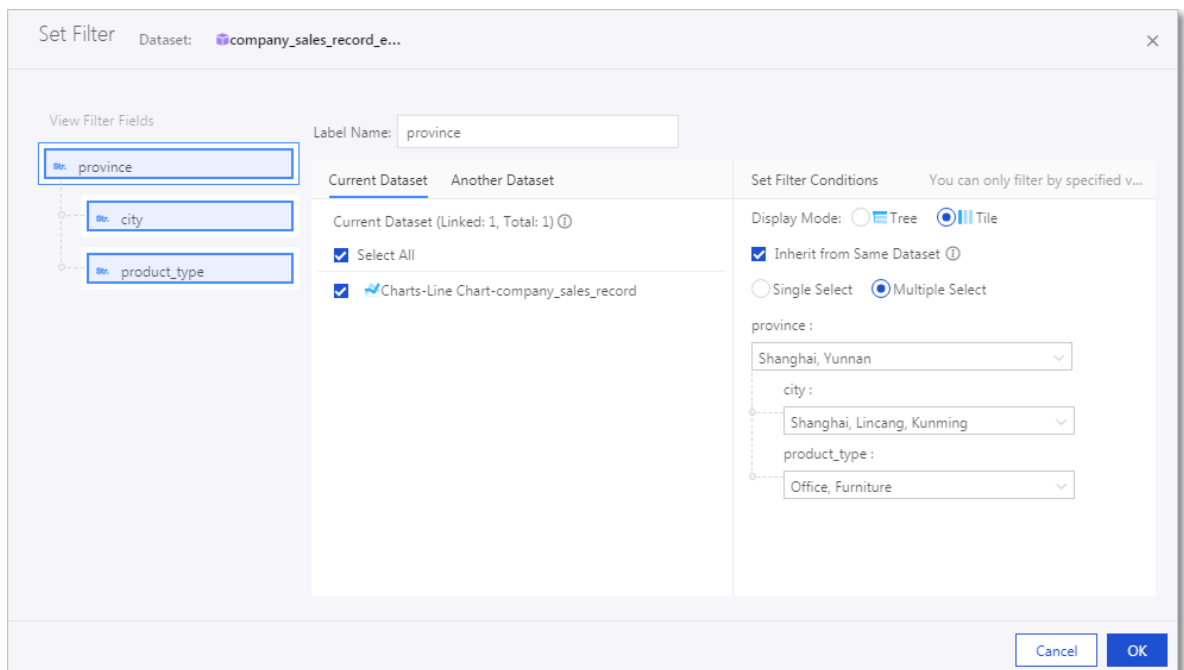
**Note:**

- The cascade filter supports three-level cascades, with lines connecting the parent node and child nodes.

- You can rename the cascade fields.



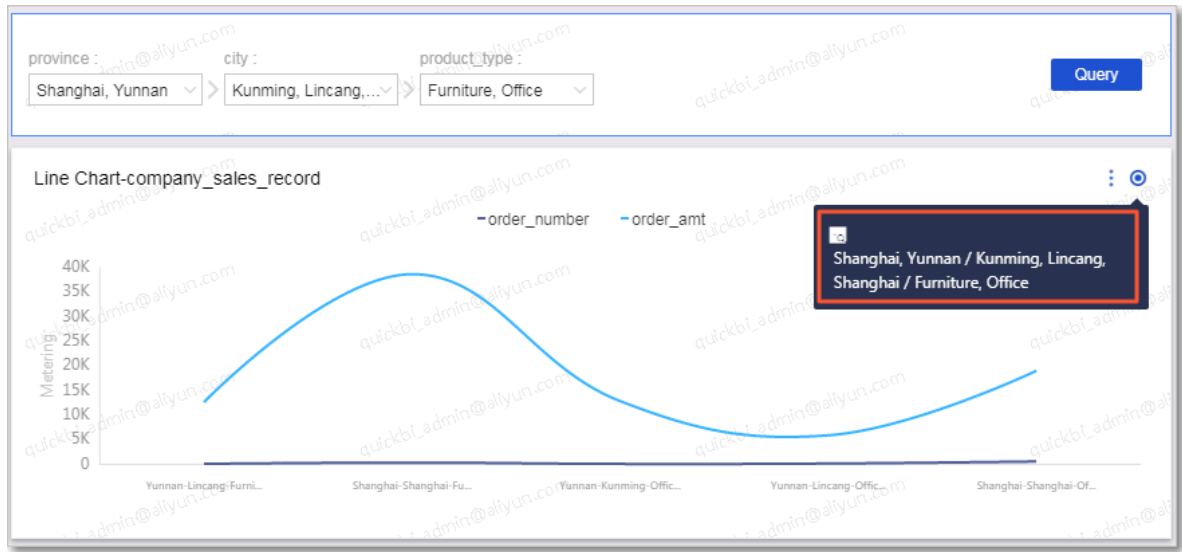
4. Click the Set Filter icon to set filter conditions and then click OK. This example selects Tile and Multiple Select, as shown in the following figure.



**Note:**

Cascade filter supports the Tree and Tile display options. You can also select Inherit from Same Dataset.

5. Click Query, as shown in the following figure.



**Note:**

To view details of the cascade, hover over the cascade icon in the upper-right corner of the line chart.

The filter bar supports cascade filter, which makes it easier for you to specify multiple filter conditions.

#### 4.4.3.4.8.2.3 Query data by date

Filter bars allow you to filter data by date.

#### Procedure

1. On the Data tab page, select a dataset and filter fields, for example, report\_date.
2. Select the target chart.

3. Click `report_date` to open the query page, as shown in [Figure 4-102: Query data by time duration](#).

Figure 4-102: Query data by time duration



4. Specify a time duration and click Query.

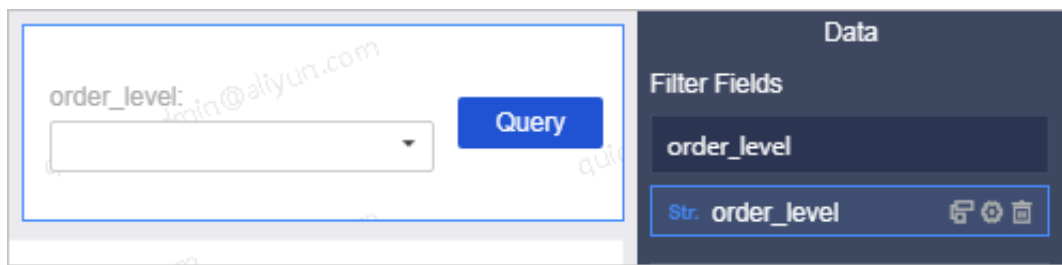
#### 4.4.3.4.8.2.4 Filter bars

You can query text data using the filter bar widget.

#### Procedure

1. On the Data tab page, select a dataset and filter fields, for example, `order_level`.
2. Select the target chart.
3. Click `order_level` to open the query page, as shown in [Figure 4-103: Query text data](#).

Figure 4-103: Query text data

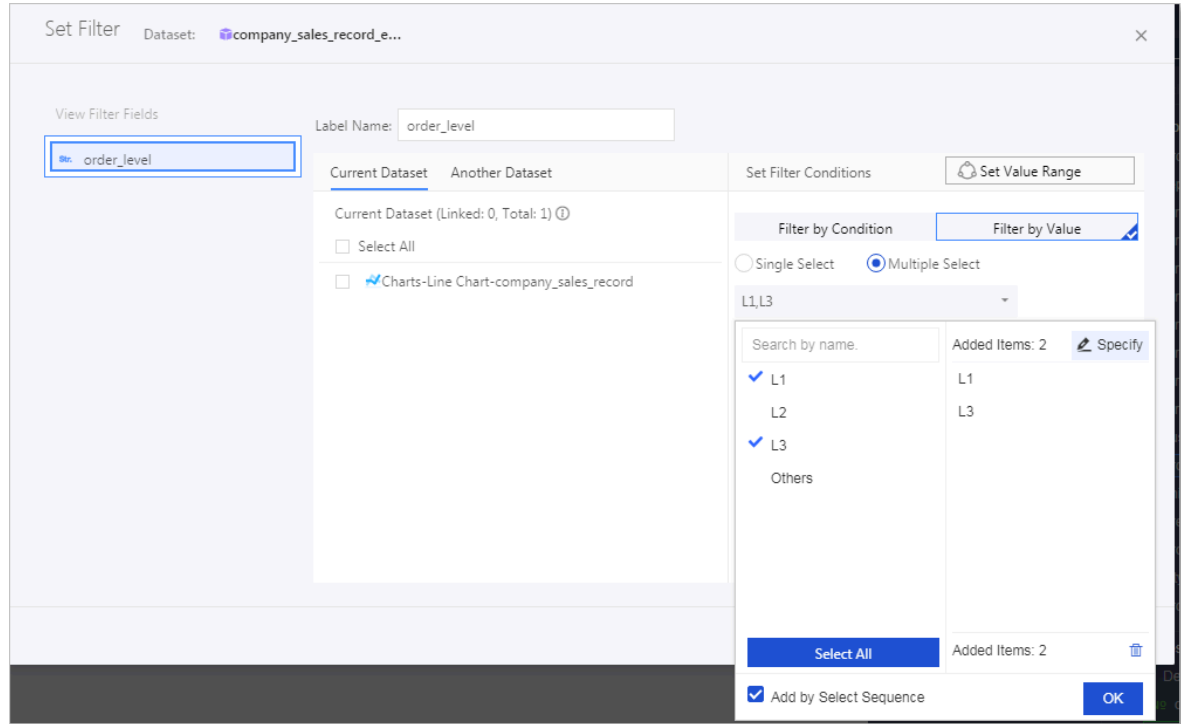


4. Set the filter condition, for example, filter by value.

The system automatically displays all options for the `order_level` field in the drop-down list. You can select Single Select or Multiple Select.

5. Click the drop-down arrow and select a value, as shown in [Figure 4-104: Filter by value](#).

Figure 4-104: Filter by value



6. Click OK.
7. Click Query, as shown in [Figure 4-105: Query by value](#).

Figure 4-105: Query by value



#### 4.4.3.4.8.3 Text area

The text area widget allows you to enter text into a text area, for example, add a title for a chart.

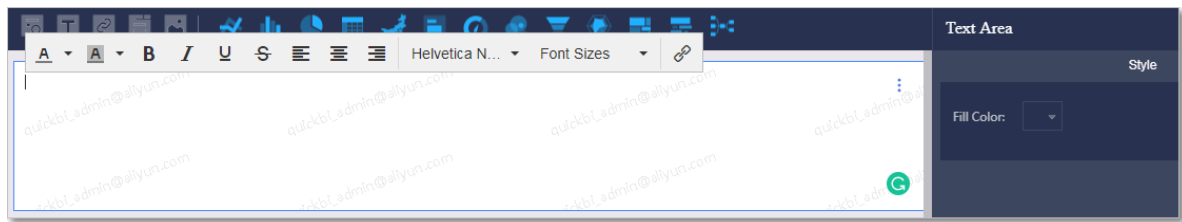
#### Procedure

1. [Go to the target dashboard](#).
2. Click the Text Area icon.



3. Enter text into the text box, as shown in [Figure 4-106: Text area](#).

Figure 4-106: Text area



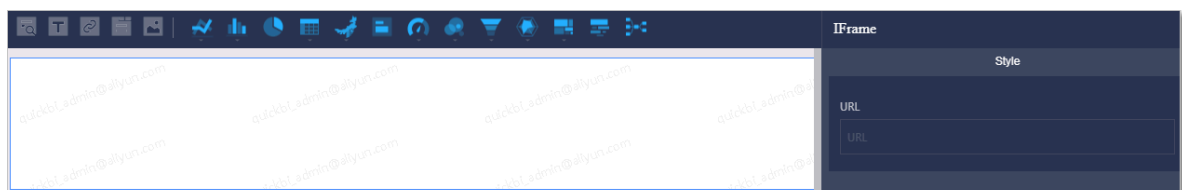
#### 4.4.3.4.8.4 IFrame

You can use the IFrame widget to insert webpages to query Internet data or browse webpages or websites related to the data on the current dashboard in real time.

#### Procedure

1. [Go to the target dashboard](#).
2. Click the IFrame icon.
3. In the URL input box, enter the address of the target webpage, as shown in [Figure 4-107: IFrame](#).

Figure 4-107: IFrame



#### Note:

The webpage address must use HTTPS.

In the Show Title input box, you can specify a title for the current IFrame widget.

To delete the current IFrame widget, click the More icon in the upper-right corner of the IFrame widget and select Delete.

#### 4.4.3.4.8.5 Tab

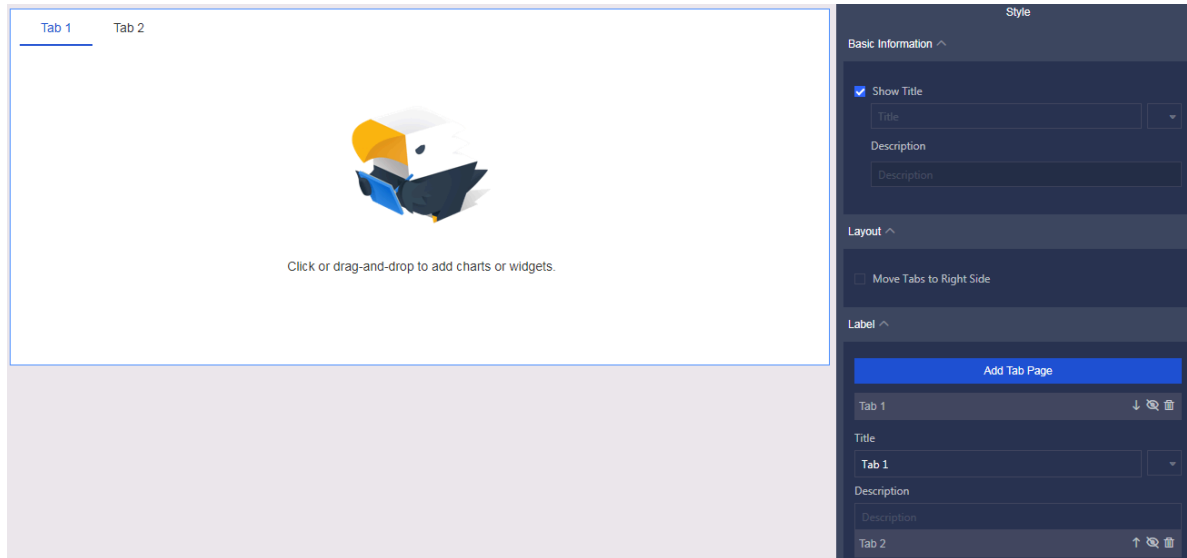
The tab widget enables you to display charts as tab pages on a dashboard.

#### Procedure

1. [Go to the target dashboard](#).

2. Click the Tab icon.
3. Click Add Tab Page to add a tab page, as shown in [Figure 4-108: Tab settings](#).

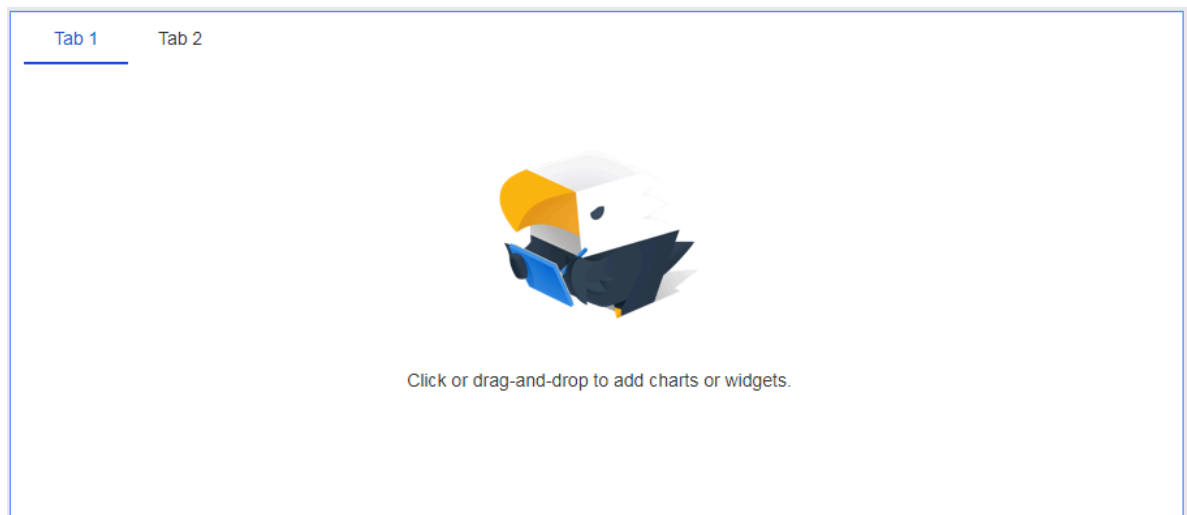
Figure 4-108: Tab settings



4. Select a tab page where you want to add charts, as shown in [Figure 4-109: Tab pages](#).

Click Tab 1 and the tab name Tab 1 becomes blue.

Figure 4-109: Tab pages



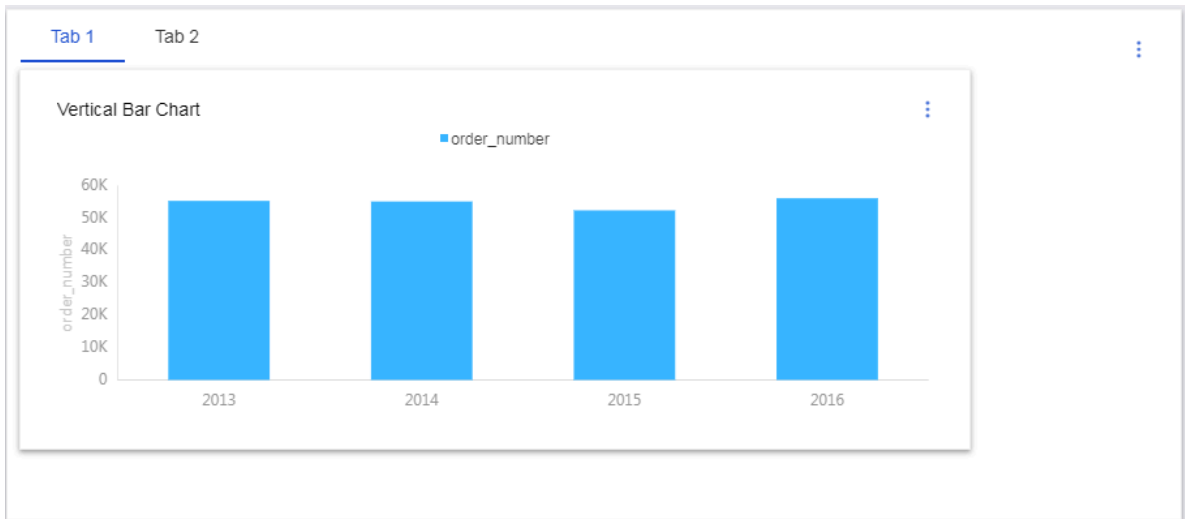
5. Click a chart icon to add a chart to Tab 1, as shown in [Figure 4-110: Add charts](#).

Figure 4-110: Add charts



Add and configure a chart. The following figure [Figure 4-111: Tab pages](#) shows the tab pages.

Figure 4-111: Tab pages



To delete the current tab widget, click the More icon in the upper-right corner of the tab page, and click Delete.

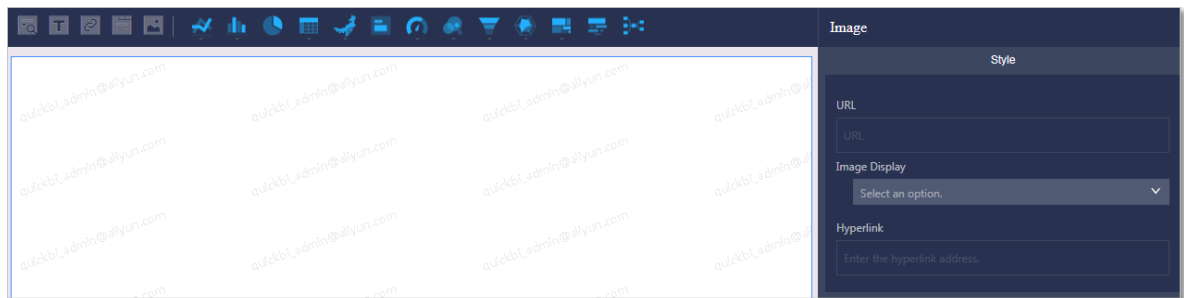
#### 4.4.3.4.8.6 Image

The Image widget allows you to insert images into a dashboard, and adjust the image position and display effects as needed.

#### Procedure

1. *Go to the target dashboard.*
2. **Click the Image icon.**
3. **Enter the URL of the target image.**
4. **Select a display effect from the Image Display drop-down list, as shown in *Figure 4-112: Edit the image on the Style tab page.***

Figure 4-112: Edit the image on the Style tab page



## 4.4.4 Create a dashboard

### 4.4.4.1 Line charts

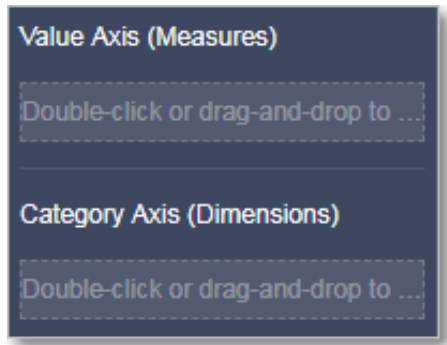
A line chart shows data changes as a series of data points connected by straight line segments and displays continuous data that changes over time. It is suitable for analyzing and displaying data trends during equal time periods. You can also use a line chart to analyze the interactions among multiple groups of data that change over time. For example, you can use a line chart to analyze the sales volume of one or multiple types of products to predict the future sales volume.

#### Context

A line chart consists of the category axis and value axis. The category axis is horizontal and determined by dimensions, such as date, province, and product type. The value axis is vertical and determined by measures, such as order amount and performance metrics.

The system automatically matches dimensions and measures with the category axis and value category. Follow the instructions to add fields, as shown in [Figure 4-113: The category axis and value axis in a line chart](#).

Figure 4-113: The category axis and value axis in a line chart



You must specify at least one dimension to determine the category axis, and at least one measure to determine the value axis. If you need to use the color legend, only one dimension can be specified for the color legend.



**Note:**

The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.

The following example uses the `company_sales_record` dataset to describe how to use a line chart to demonstrate the order amount of each type of products in each province per year.

### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. On the **Datasets** page, find the `company_sales_record` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Line Chart** icon.
5. On the **Data** tab page, select the target dimensions and measures.

In the **Dimensions** list, find and add the `report_date` (year), `province`, and `product_type` dimensions to the **Category Axis** area. In the **Measures** list, find

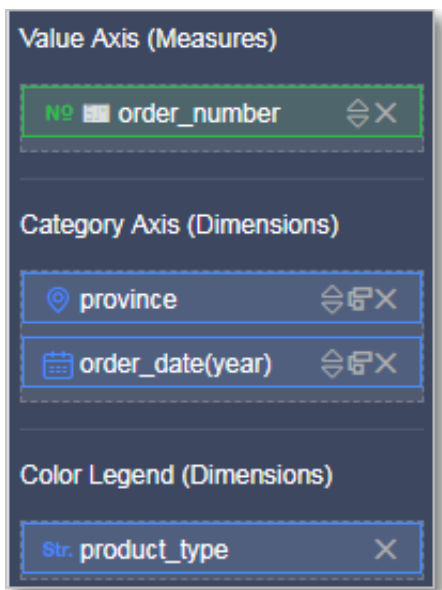
and add the order\_amt measure to the Value Axis area, as shown in the following figure *Figure 4-114: Select fields for the line chart*:

**Note:**

Make sure that you have converted the province dimension from String to Geo.

For more information about converting dimensions to another type, see [Edit a dimension](#).

Figure 4-114: Select fields for the line chart

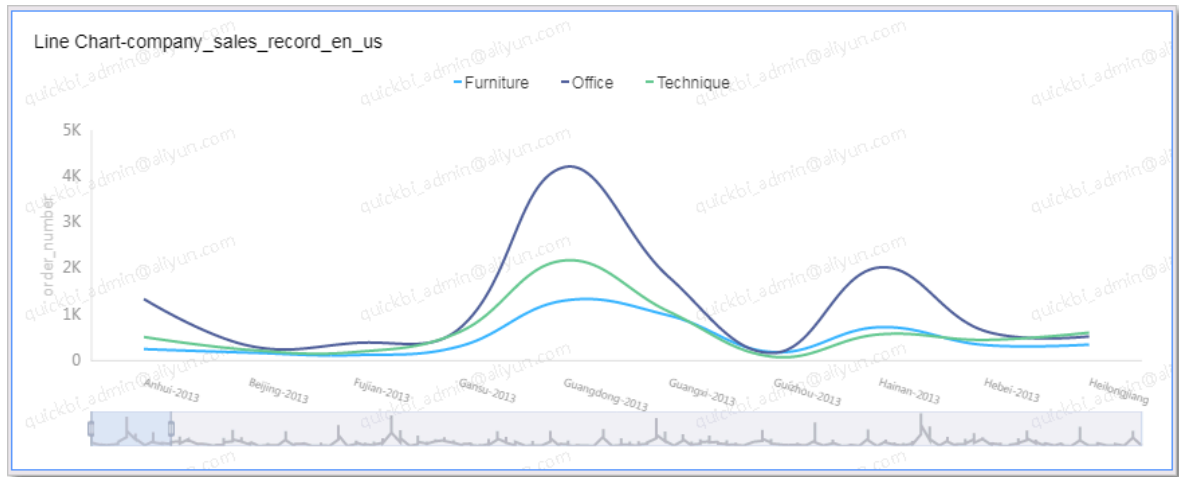


6. Drag the product \_type dimension from the Category Axis area to the Color Legend area.
7. Click Update and the system then updates the chart.

8. On the Style tab, you can change the title, layout, and legend mode, as shown in

*Figure 4-115: The line chart.*

Figure 4-115: The line chart



9. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.

10. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.2 Area charts

An area chart shows data changes and trends with different sizes of areas. It also displays continuous data that changes over time. Area charts are suitable for analyzing and displaying data trends during equal time periods. You can also use an area chart to analyze the interactions among multiple groups of data that change over time. For example, you can use an area chart to analyze the sales volume of one or multiple types of products to predict the future sales volume.

An area chart consists of the category axis and the value axis. The category axis is horizontal and determined by dimensions, such as date, province, and product type. The value axis is vertical and determined by measures, such as order amount and performance metrics.

The system automatically matches dimensions and measures with the category axis and value category. Follow the instructions to add fields.

#### Notes

You must specify at least one dimension to determine the category axis, and at least one measure to determine the value axis. If you need to use the color legend, only one dimension can be specified for the color legend.



**Note:**

The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.

The following example uses the `company_sales_record` dataset to describe how to use an area chart to demonstrate the order amount of each type of products in different provinces.

1. Log on to the Quick BI console.
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. Click the Area Chart icon. An area chart is created in the display area of the dashboard.
5. On the Data tab page, select the target dimensions and measures.

In the Dimensions list, find and add the province dimension to the Category Axis area. In the Measures list, find and add the `order_amt` measure to the Value Axis area.



**Note:**

Make sure that you have converted the province dimension from String to Geo.

6. Find and add the `product_type` dimension to the Color Legend area and click Update.

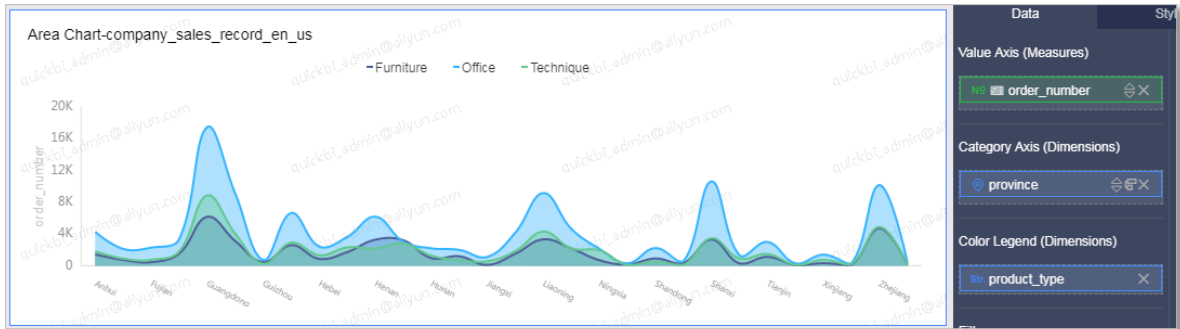


**Note:**

The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.



7. On the Style tab page, you can change the title, layout, legend mode, and axis style, as shown in the following figure:



#### Note:

You can switch to other area chart types, such as stacked area charts, 100% stacked area charts, and stacked line charts.

8. Click Save in the upper-right corner to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

### 4.4.4.3 Vertical bar charts

A vertical bar chart can be used to demonstrate the differences among multiple objects. It shows data changes over a specific period of time, or the comparison among objects, for example, the traffic counts of a road crossing during different periods of time.

#### Context

Similar to a [line chart](#), a vertical bar chart consists of the category axis and value axis. You can use a vertical bar chart to show data changes over a specific period of time, or compare the differences among multiple objects.

This topic uses the following scenarios to describe how to filter data and use the Dual Y-Axis function in a vertical bar chart.

- **Scenario 1:** Compare the shipping costs of different products in provinces of East China.
- **Scenario 2:** Compare the order amount and average profits of different products in different provinces.

You must specify at least one dimension to determine the category axis, such as province or product type. You must specify at least one measure to determine

the value axis, such as order amount or profit amount. You can only specify one dimension to determine the color legend.



**Note:**

The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.

## Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.

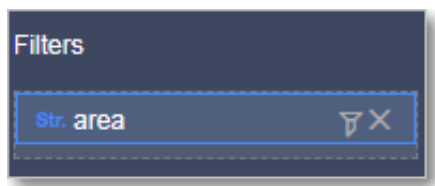
**4. On the dashboard edit page, click the Vertical Bar Chart icon.**

**Scenario 1:** The following example uses the `company_sales_record` dataset to describe how to use a vertical bar chart to compare the shipping cost of different products in provinces of East China.

- a) In the Dimensions list, find and add the area dimension to the Filters area, as shown in *Figure 4-116: The filter*.

**You can use the filter to filter out East China data from the area dimension.**

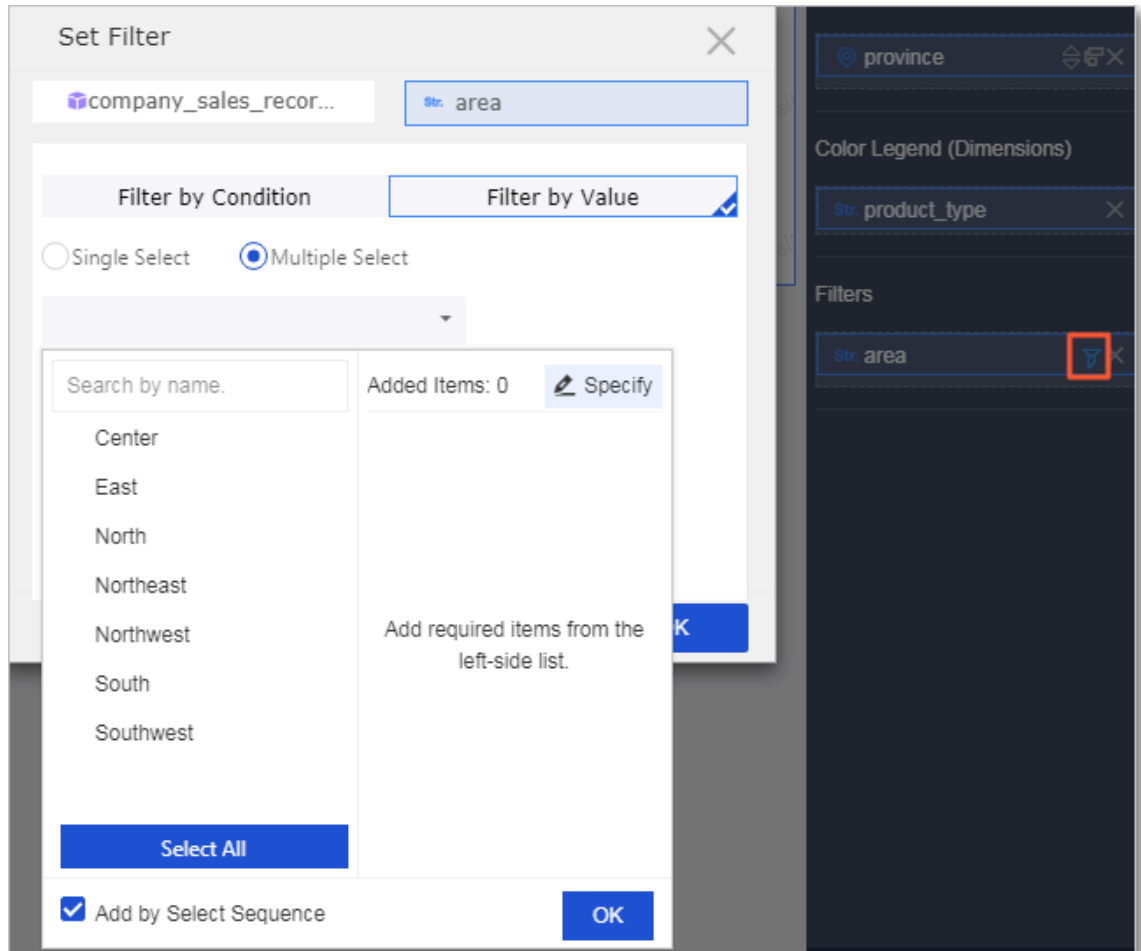
Figure 4-116: The filter



- b) Click the Filter to set filter conditions.

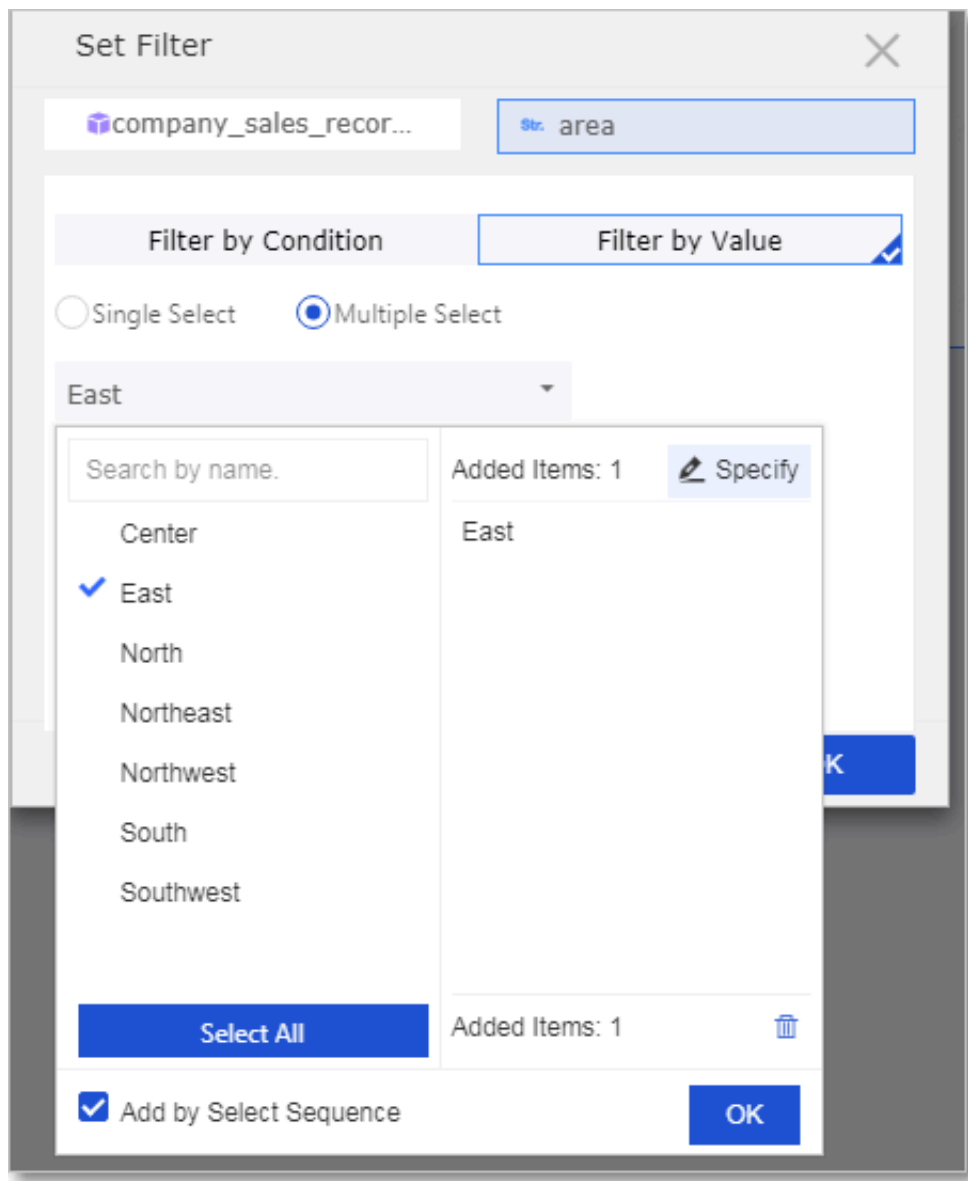
- c) **Select Filter by Value.** The system automatically lists all available options, as shown in *Figure 4-117: Filter by value.*

Figure 4-117: Filter by value



- d) Select East and click OK, as shown in *Figure 4-118: Select the target region.*

Figure 4-118: Select the target region



- e) In the Dimensions list, find and add the province and product\_type dimensions to the Category Axis area and Color Legend area, respectively.
- f) In the Measures list, find and add the shipping\_cost measure to the Value Axis area.



**Note:**

**Make sure that you have converted the province dimension from String to Geo.**

For more information about converting dimensions to another type, see [Edit a dimension](#).

- g) Drag the `product_type` dimension from the Category area to the Color Legend area, as shown in [Figure 4-119: Color legend](#).

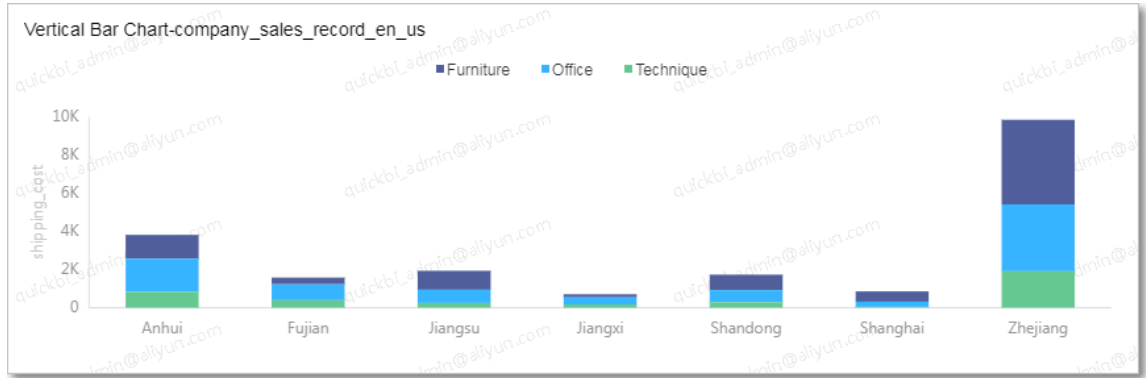
Figure 4-119: Color legend



- h) Click Update and the system then updates the chart.

- i) On the Style tab page, select the Stacked check box, as shown in [Figure 4-120: The vertical bar chart](#).

Figure 4-120: The vertical bar chart



**Scenario 2:** The following example uses the `company_sales_record` dataset to describe how to use a vertical bar chart to compare the order amount and average profits of different products in different provinces.



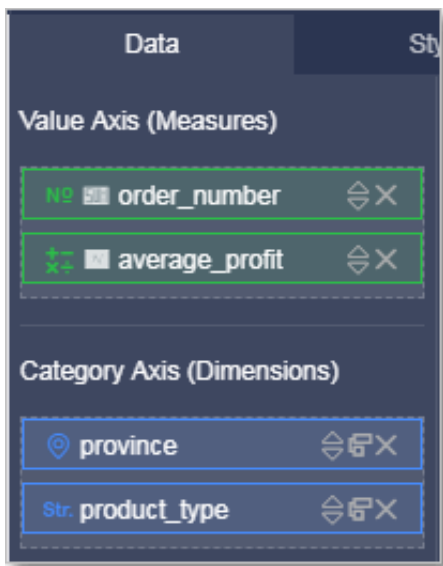
**Note:**

Data modeling may be required in this scenario. For more information about data modeling, see [Add a calculated field](#).

- a) On the Data tab page, select the target dimensions and measures.

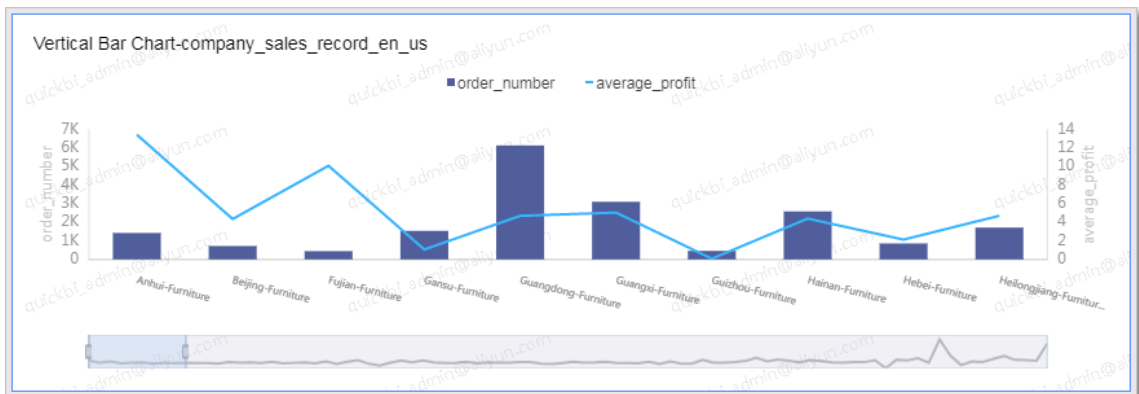
In the Dimensions list, find and add the province and product\_type dimensions to the Category Axis area. In the Measures list, find and add the order\_amt and average\_profit measures to the Value Axis area.

Figure 4-121: Select fields for the vertical bar chart



- b) Click Update and the system then updates the chart.
- c) On the Style tab page, select the Dual Y-Axis check box, as shown in the following figure.

Figure 4-122: The vertical bar chart



- d) Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.



e) Click OK to save the dashboard.

By default, the dashboard is saved to My Items on the Dashboards page.

#### 4.4.4.4 Horizontal bar charts

Similar to a vertical bar chart, a horizontal bar chart displays the differences between data in different categories.

##### Notes

You must specify at least one dimension to determine the category axis, such as a province or product type. You must specify at least one measure to determine the value axis, such as order amount or profits. You can only specify one dimension to determine the color legend.



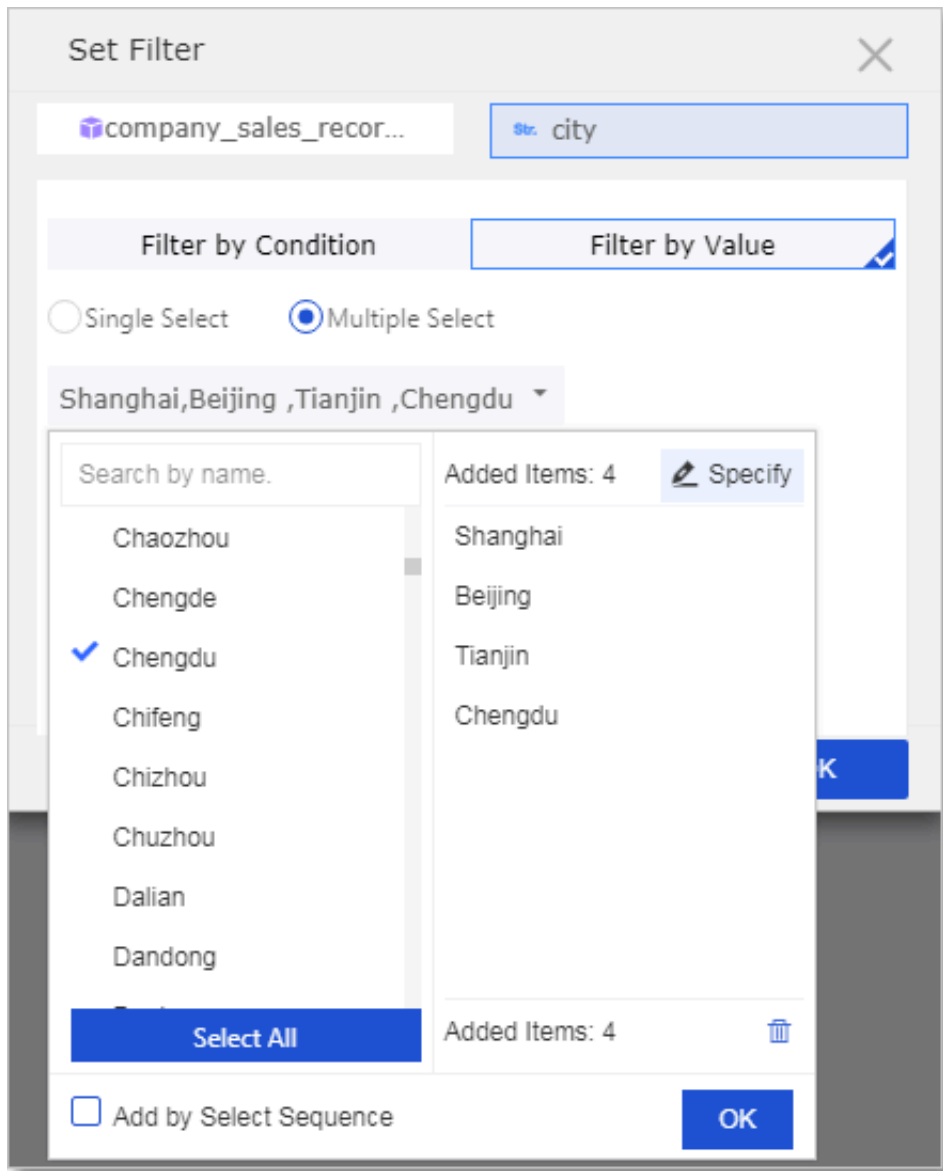
##### Note:

The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.

The following example uses the company\_sales\_record dataset to describe how to use a horizontal bar chart to compare the shipping costs in different regions.

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Horizontal Bar Chart icon. A horizontal bar chart is created in the display area of the dashboard.

5. Find and add the city dimension to the Filters area to filter out the four municipalities, as shown in the following figure.



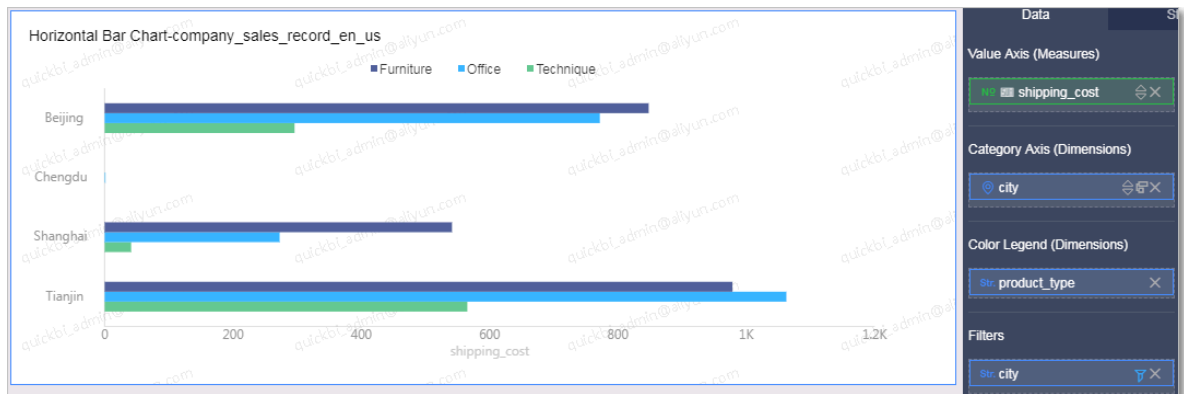
6. Drag the city dimension to the Category Axis area. Find and add the shipping\_cost measure to the Value Axis area. Find and add the product\_type dimension to the Color Legend area.



**Note:**

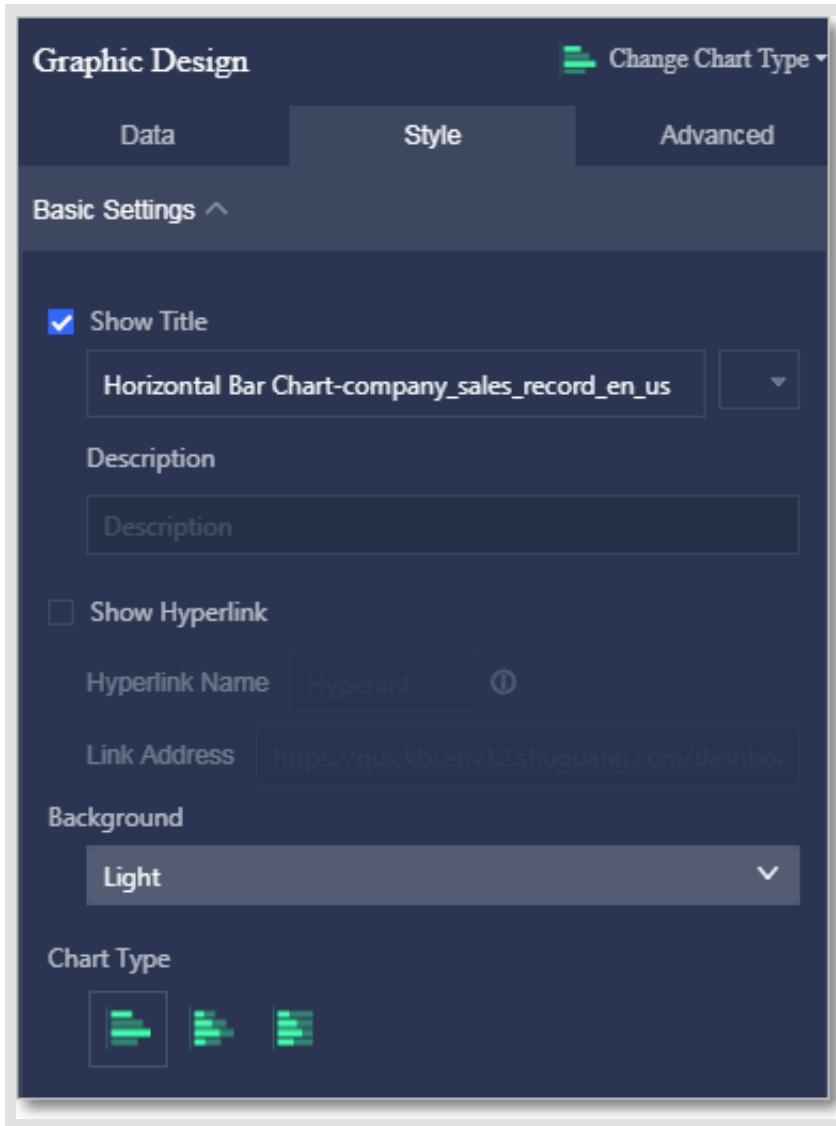
- Make sure that you have converted the city dimension from String to Geo.
- The color legend is available only when the value axis has one measure. Otherwise, you cannot use the color legend.

7. Click Update and the system then updates the chart, as shown in the following figure.



**Note:**

You can also switch the current chart to another bar chart type, such as a stacked horizontal bar chart and 100% stacked horizontal bar chart, as shown in the following figure.



8. Click Save in the upper-right corner to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.5 Progress bar charts

Similar to a gauge, a progress bar can be used to display the progress of a specific metric.

A progress bar consists of a pointer. The pointer is determined by measures, such as order amount.

#### Note

- You must specify at least one measure. You can specify up to five measures to determine the pointer.

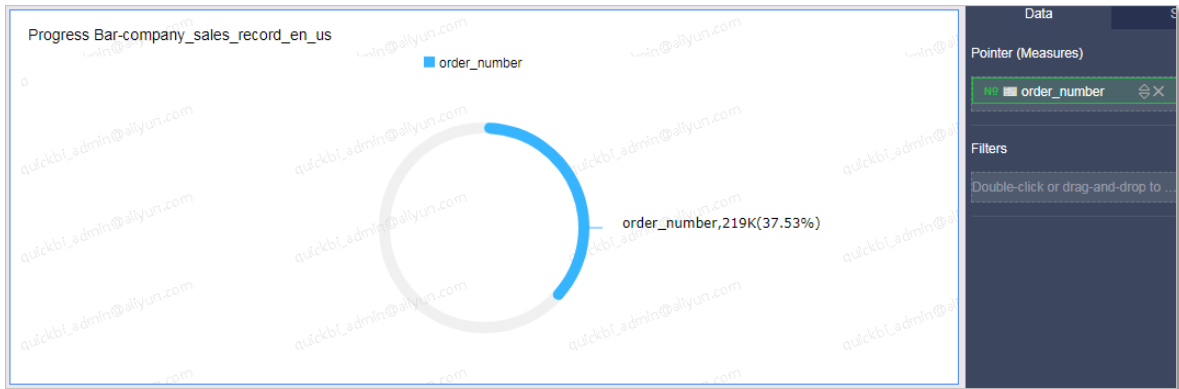
- To use a progress bar chart, you must choose **Style > Series Settings** to set the maximum and minimum values.

The following example uses the `company_sales_record` dataset to describe how to use a progress bar chart to show the order amount status.

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. On the **Datasets** page, find the `company_sales_record` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Progress Bar** chart. A progress bar chart is created in the display area of the dashboard.
5. On the **Data** tab page, select the target measures.

In the **Measures** list, find and add the `order_amt` measure to the **Pointer** area. On the **Style** tab, you can change the title and legend of the chart, set an alias for a measure, and set the maximum and minimum values.

6. Click **Update** and the system then updates the chart, as shown in the following figure.



7. Click **Save** in the upper-right corner to save the dashboard.

To delete the chart, click the **More** icon in the upper-right corner of the chart and select **Delete**.

#### 4.4.4.6 Pie charts

A pie chart shows data of different objects. Each object has a unique color or pattern. A pie chart can be used to show the ratio of each object to the total amount. For example, you can use a pie chart to show the ratio of the five social insurances (endowment, medical, unemployment, employment injury, and

maternity insurances) and housing fund to the total personal income, or the ratio of the sales volume of a car brand to the total car sales volume.

## Context

A pie chart consists of slices. Slice labels are determined by a dimension, such as province or product type. The central angle of each slice is determined by a measure, such as order amount or prices.

You can specify only one dimension to determine slice labels, such as area or product type. You can specify only one measure to determine the central angle, such as order amount or profits.

The following example uses the `company_sales_record` dataset to describe how to use a pie chart to compare the shipping costs in different regions.

## Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Pie Chart icon.
5. On the Data tab page, select the target dimension and measure.

In the Dimensions list, find and add the area dimension to the Labels area. In the Measures list, find and add the `shipping_cost` measure to the Central Angle area, as shown in [Figure 4-123: Select fields for the pie chart](#).



**Note:**

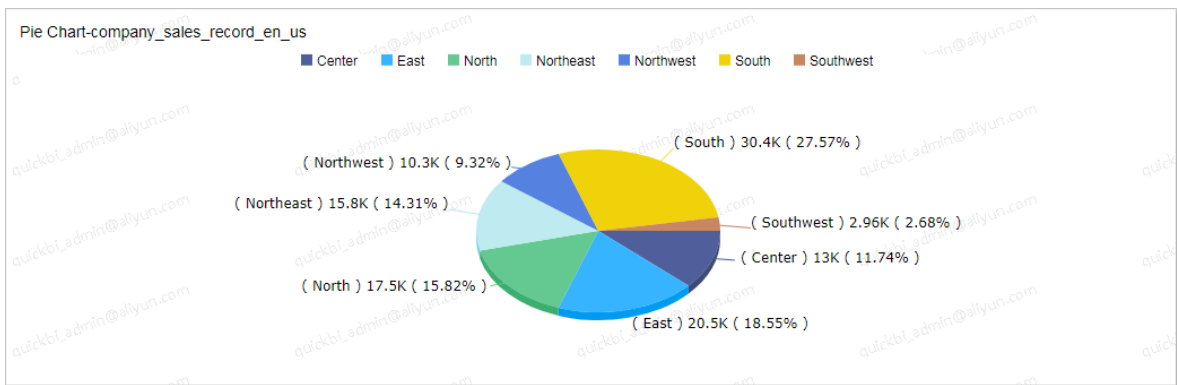
**Make sure that you have converted the area dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-123: Select fields for the pie chart



6. Click Update and the system then updates the chart.
7. On the Style tab page, select the 3D display mode and the Name, Value (Percentage) label style, as shown in [Figure 4-124: The pie chart](#).

Figure 4-124: The pie chart



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.7 Bubble maps

A bubble map uses a map profile as its background and shows data distribution with bubbles in different sizes. It directly displays data metrics and distribution

status in a country or region. For example, you can use a bubble map to display tourist arrivals of different destinations, or the average income in different regions.

## Context

A bubble map consists of geographic locations and the bubble size. Geographic locations are determined by a dimension, such as province. The bubble size is determined by measures, such as shipping costs and order amount.

You can specify only one dimension to determine geographic locations, such as area, province, or city. The type of the dimension must be Geo. You must specify at least one measure to determine the bubble size. You can specify up to five measures.

The following example uses the `company_sales_record` dataset to describe how to use a bubble map to compare the order amount and average profits in different provinces.

Data modeling may be required in this scenario.

For more information about data modeling, see [Add a calculated field](#).

## Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Bubble Map icon.
5. On the Data tab page, select the target dimension and measures.

In the Dimensions list, find and add the province dimension to the Geo Location area. In the Measures list, find and add the `order_amt` and `average_profit` measures to the Bubble Size area, as shown in [Figure 4-125: Select fields for the bubble map](#).

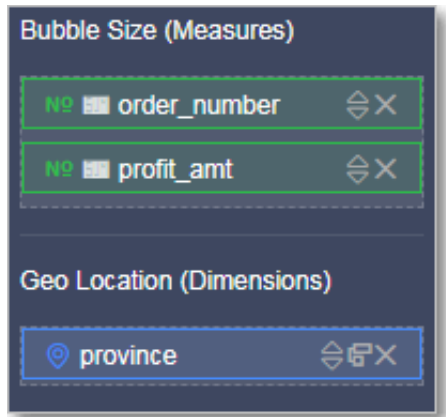


Note:



**Make sure that you have converted the province dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

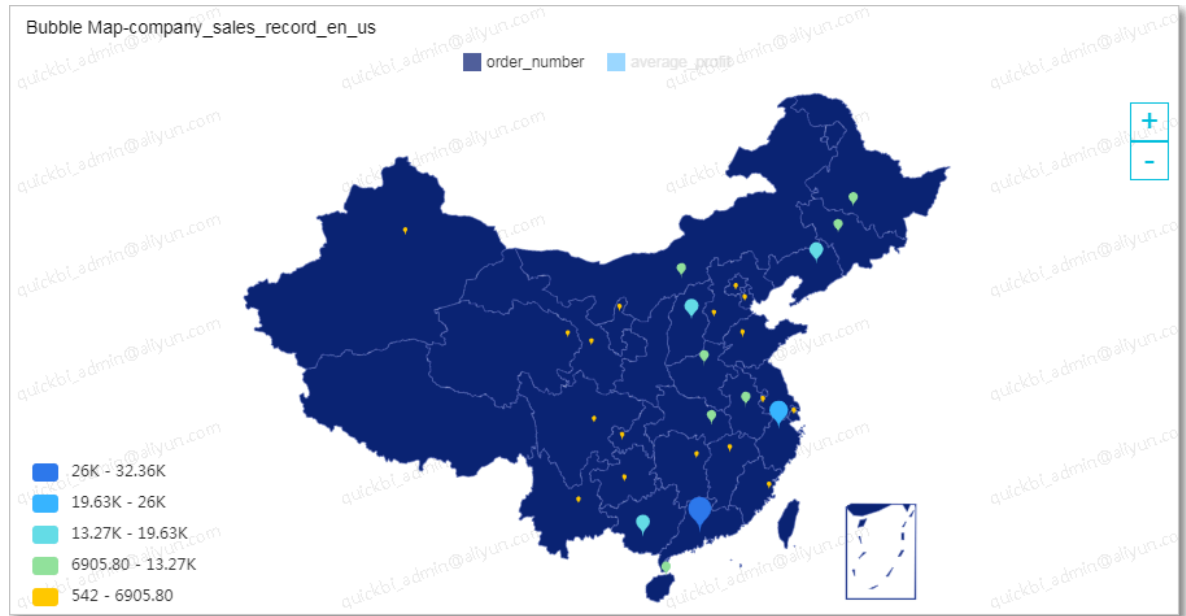
Figure 4-125: Select fields for the bubble map



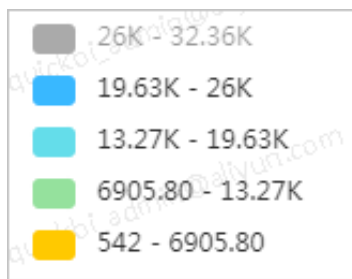
6. Click Update and the system then updates the map.

7. On the Style tab page, you can change the title and layout of the bubble map, as shown in [Figure 4-126: The bubble map](#).

Figure 4-126: The bubble map



- You can switch between `order_amt` and `average_profit` to show different data.
- You can click a color icon to hide the data that you do not want to show.



- You can click the plus (+) or minus (-) sign to zoom in or zoom out the map.
8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
  9. Click OK to save the dashboard.

To delete the map, click the More icon in the upper-right corner of the map and select Delete.

#### 4.4.4.8 Colored maps

Similar to [bubble maps](#), a colored map displays data in one hue with different saturation levels to demonstrate data distribution.

## Context

A colored map consists of geographic locations and the colorscale. Geographic locations are determined by a dimension, such as provinces. The colorscale is determined by measures, such as prices and profits.

You can specify only one dimension to determine geographic locations. The type of the dimension must be Geo. You must specify at least one measure to determine the colorscale. You can specify up to five measures.

The following example uses the `company_sales_record` dataset to describe how to use a colored map to compare the shipping costs, prices, and profits in different regions.

## Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Colored Map icon.
5. On the Data tab page, select the target dimension and measures.

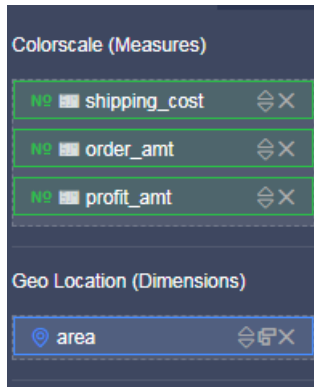
In the Dimensions list, find and add the area dimension to the Geo Location area. In the Measures list, find and add the `order_amt`, `profit_amt`, and `shipping_cost` measures to the Colorscale area, as shown in *Figure 4-127: Select fields for the colored map.*



**Note:**

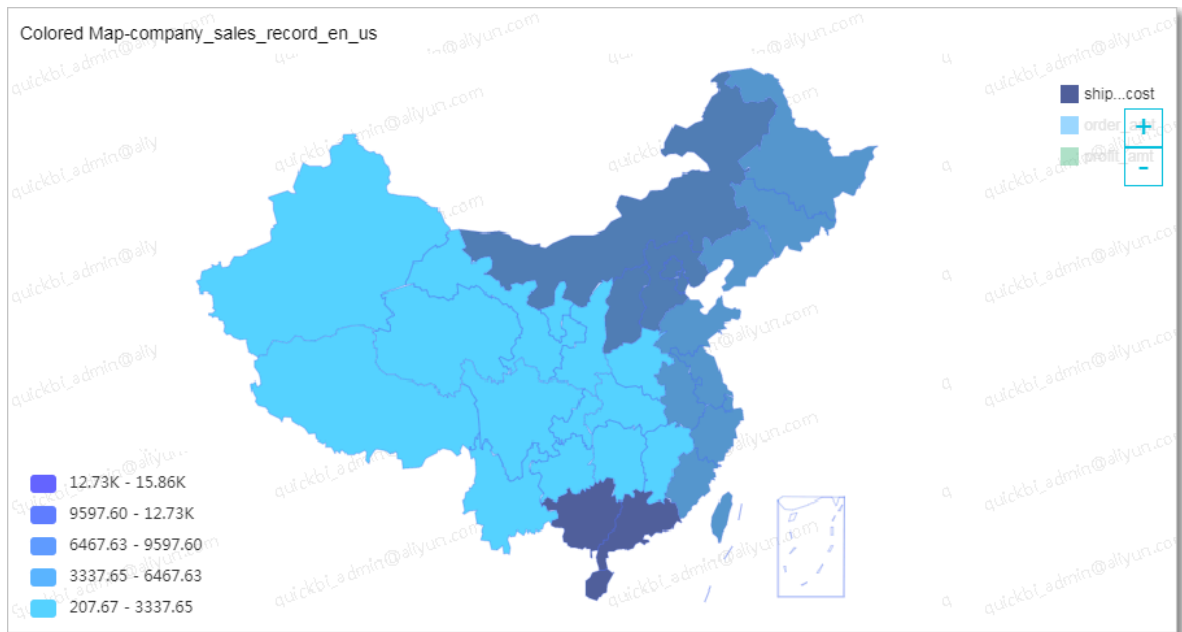
**Make sure that you have converted the area dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-127: Select fields for the colored map



6. Click Update and the system then updates the map.
7. On the Style tab page, select the Right icon in the Show Legend area, as shown in [Figure 4-128: The colored map](#).

Figure 4-128: The colored map



**You can change the title of the map, adjust the map position and size, and hide the data that you do not need to show. For more information, see [Colored maps](#).**

8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.

## 9. Click OK to save the dashboard.

To delete the map, click the More icon in the upper-right corner of the map and select Delete.

### 4.4.4.9 Geo bubble maps

A geo bubble map uses a map as its background and displays data values with bubbles. It clearly demonstrates data metrics and distribution in a country, region, or city. Compared with bubble maps, geo bubble maps provide more accurate geographic locations.

A geo bubble map displays data in one hue with different saturation levels to demonstrate data distribution. Geographic locations are determined by a dimension, such as province. The saturation level is determined by a measure, such as order amount. Only one dimension and one measure can be specified in a geo bubble map. The type of the dimension must be Geo, such as area, province, or city. The following example uses the `company_sales_record` dataset.

Sample scenario: Compare the order amount of different provinces in North China.

1. Log on to the Quick BI console.
2. In the left-side navigation pane, click Datasets.
3. Find the `company_sales_record` dataset and click the Create Dashboard icon in the Actions column.



**Note:**

If you are using Quick BI Enterprise Standard, select Standard or Full Screen. In the following example, Standard is selected.

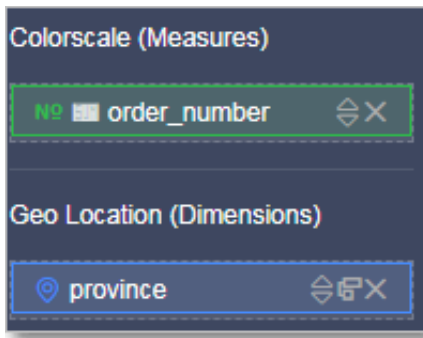
4. On the dashboard edit page, click the Geo Bubble Map icon.
5. On the Data tab page, select the target measure and dimension.

In the Dimensions list, find and add the province dimension to the Geo Location area. In the Measures list, find and add the `order_number` measure to the Colorscale area, as shown in the following figure.

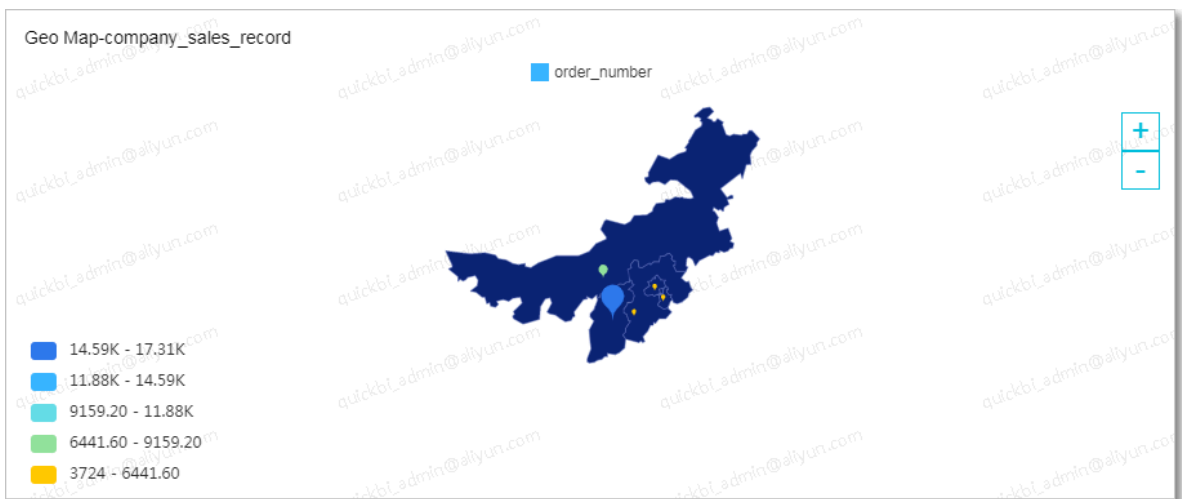


**Note:**

**Make sure that you have converted the province dimension from String to Geo.**



6. Click Update and the system then updates the map.
7. On the Style tab page, you can change the saturation level and value ranges, as shown in the following figure:



8. Click Save in the upper-right corner to save the dashboard.

To delete the current map, click the More icon in the upper-right corner of the map and select Delete.

#### 4.4.4.10 Geo maps

A geo map displays data in one hue with different saturation levels to demonstrate data distribution. Compared with colored maps, geo maps provide more accurate geographic locations.

A geo map consists of geographical locations displayed in one hue with different saturation levels. Geographical locations are determined by a dimension, such as province. The saturation level is determined by a measure, such as order amount. Only one dimension and one measure can be specified in a geo map. The type of the dimension must be Geo.

The following example uses the `company_sales_record` dataset.

**Sample scenario:** Compare the order amount of different provinces in South China.

1. Log on to the Quick BI console.
2. In the left-side navigation pane, click **Datasets**.
3. Find the `company_sales_record` dataset and click the **Create Dashboard** icon in the **Actions** column.



**Note:**

If you are using Quick BI Enterprise Standard, select **Standard** or **Full Screen**. In the following example, **Standard** is selected.

4. On the dashboard edit page, click the **Geo Map** icon.
5. On the **Data** tab page, select the target dimension and measure.

In the **Dimensions** list, find and add the province dimension to the **Geo Location** area. In the **Measures** list, find and add the `order_number` measure to the **Colorscale** area.




**Note:**

Make sure that you have converted the province dimension from **String** to **Geo**.

6. Click **Update** and the system then updates the map.

7. On the Style tab page, you can change the saturation level and value ranges.

**Graphic Design**  Change Chart Type ▾

Data **Style** Advanced

Display Scope

Regional Map ▾

Southeast ▾

Series Settings ^

order\_number ▾

Alias

order\_number

Data Display Format

☐ Automatic adaptation ☐ Custom format


☒ Manual input





EN

Set Value Ranges

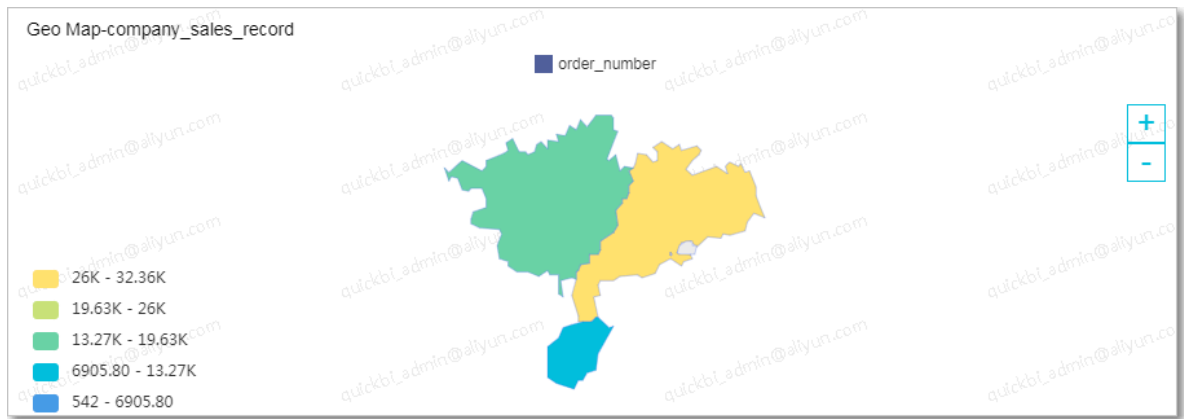
Value Ranges ?

5

Colors: 

542	6905.8	 ▾
6905.8	13269.6	 ▾
13269.6	19633.4	 ▾
19633.4	25997.2	 ▾





8. Click Save in the upper-right corner to save the dashboard.

To delete the current map, click the More icon in the upper-right corner of the map and select Delete.

#### 4.4.4.11 Cross tables

A cross table can be used to display the summary of a field and classify data items into different categories. Dimensions determine columns, and measures determine rows in the cross table. Cells support multiple aggregate functions, including sum, average, count, max, and min.

##### Context

A cross table consists of rows and columns. Rows are determined by dimensions, such as province and product type. Columns are determined by measures, such as order amount and profits.

You can specify an unlimited number of dimensions and measures to determine the rows and columns.

The following example uses the company\_sales\_record dataset to describe how to use a cross table to compare the packaging, shipping cost, order amount, and average profit of different products in multiple provinces.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Cross Table icon.

5. On the Data tab page, select the target dimensions and measures.

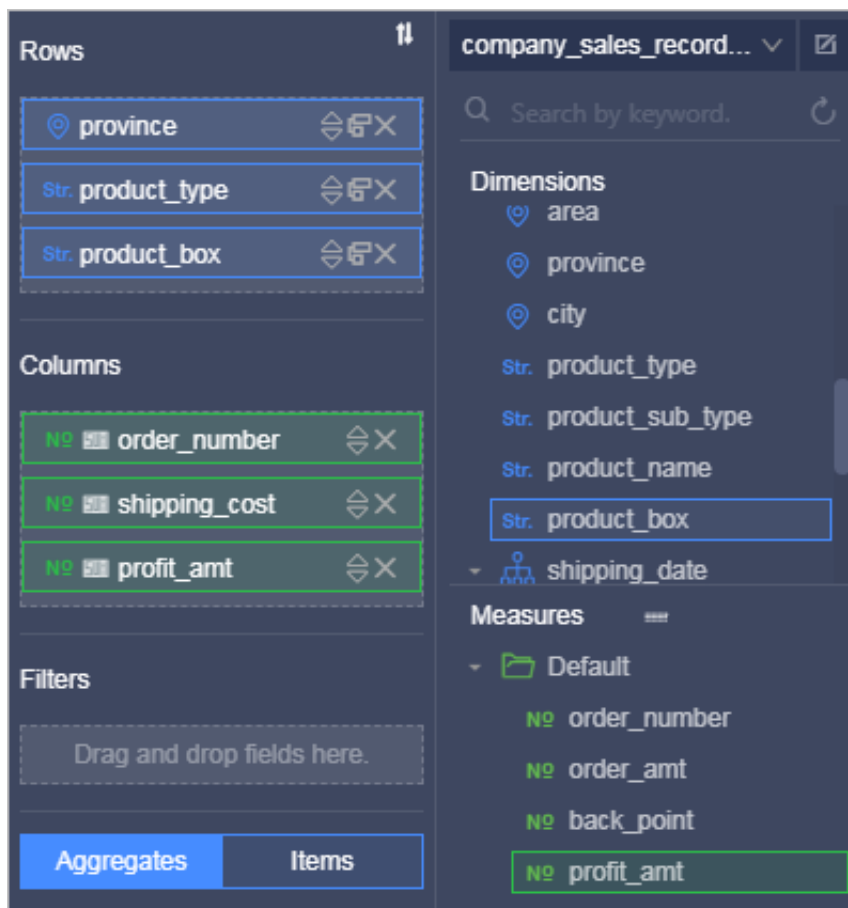
In the Dimensions list, find and add the province, product\_type, and product\_box dimensions to the Rows area. In the Measures list, find and add the order\_amt, shipping\_cost, and profit\_amt measures to the Columns area, as shown in *Figure 4-129: Select fields for the cross table*.



**Note:**

Make sure that you have converted the province dimension from String to Geo. For more information about converting a dimension to another type, see [Edit a dimension](#).

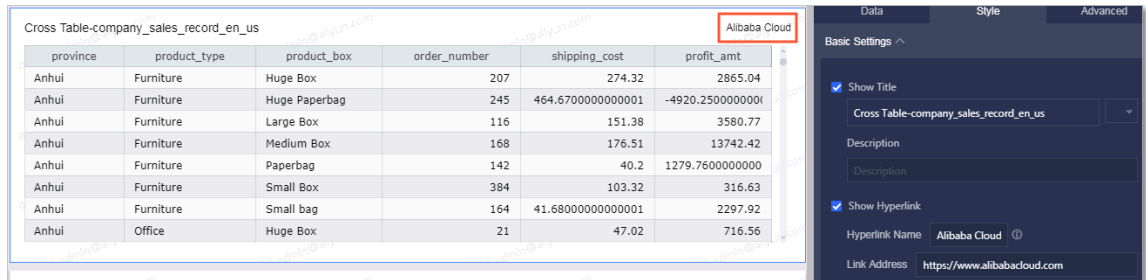
Figure 4-129: Select fields for the cross table



6. Click Update and the system then updates the chart.

**7. On the Style tab page, you can configure the following settings:**

- In the Basic Settings area, you can specify a title and hyperlink for the cross table, as shown in the following figure.



- In the Display Settings area, you can select a form topic, merge cells belong to the same category, freeze all columns, freeze specified columns, and select

the Auto (Table Head) mode to freeze columns. After you set the parameters, update the table, as shown in the following figure.

The screenshot shows a table titled 'Cross Table-company\_sales\_record\_en\_us' with 7 rows and 6 columns. The first row is highlighted. To the right of the table is a 'Display Settings' panel with tabs for 'Data', 'Style', and 'Advanced'. The 'Data' tab is active, showing options for 'Form topic' (Default, Short version), 'Show Row Numbers' (checked), 'Merge Same Cells' (unchecked), and 'Freeze' (checked). Under 'Freeze', 'Auto (Table Head)' is selected. Below this, there are fields for 'Columns', 'From First Column to', and 'From', with '0' entered in the 'From First Column to' field. At the bottom, there is a 'Display paging' section with a dropdown set to '20' and the text 'Article / Page'.

	province	product_type	product_box	order_number	shipping_cost	profit_amt
1	Anhui	Furniture	Huge Box	207	274.32	2865.04
2	Anhui	Furniture	Huge Paperbag	245	464.67000000000001	-4920.250000
3	Anhui	Furniture	Large Box	116	151.38	3580.77
4	Anhui	Furniture	Medium Box	168	176.51	13742.42
5	Anhui	Furniture	Paperbag	142	40.2	1279.760000
6	Anhui	Furniture	Small Box	384	103.32	316.63
7	Anhui	Furniture	Small bag	164	41.680000000000001	2297.92

- In the Functionality Settings area, you can set conditional formatting and sort columns.

### Conditional formatting

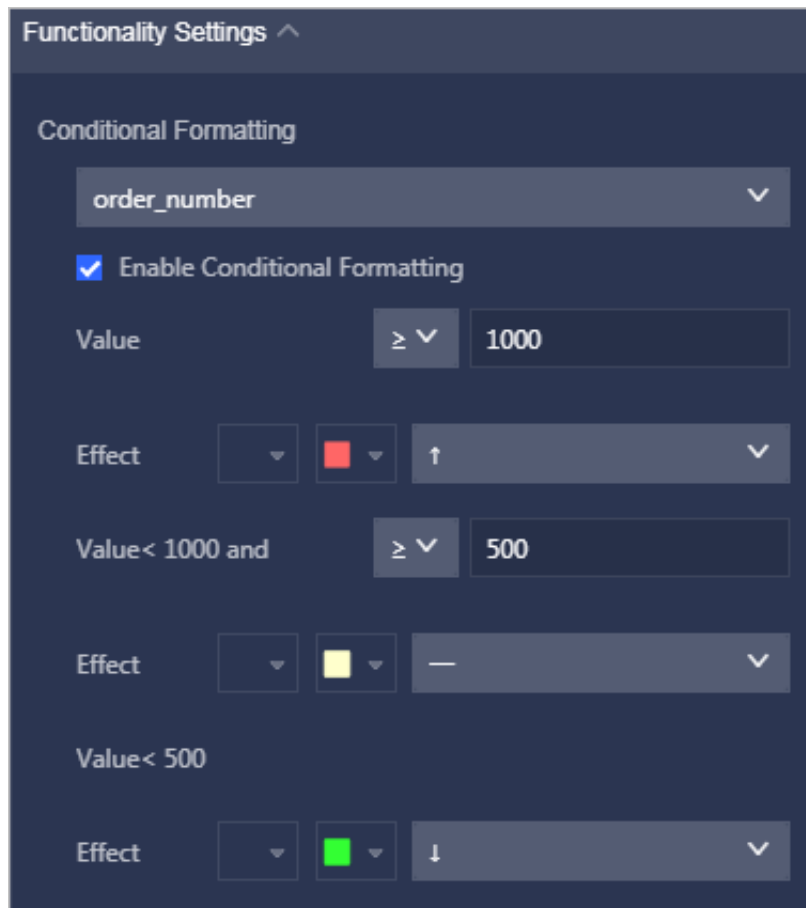
- Select the target field, and select the Enable Conditional Formatting check box to enable conditional formatting. To disable conditional formatting, clear the check box, as shown in the following figure.



- Click the drop-down icon and you can select another field, as shown in the following figure.



- Specify a value for the Value parameter, click the Color icons next to the Effect parameter, and select colors to mark the values, as shown in the following figure.



In the following example, the profit\_amt field is selected for conditional formatting. The condition is set to: values greater than 1,000, values between 500 and 1,000, and values lower than 500.

- Cells with values greater than 1,000 become red, and are marked with a green up arrow.
- Cells with values between 500 and 1,000 become grey, and are marked with an orange hyphen.
- Cells with values smaller than 500 become green, and are marked with a red down arrow.

Cross Table-company\_sales\_record\_en\_us

	province	product_type	product_box	order_number	shipping_cost	profit_amt
1	Anhui	Furniture	Huge Box	207	274.32	2865.04
2	Anhui	Furniture	Huge Paperbag	245	464.67000000000001	4920.25000
3	Anhui	Furniture	Large Box	116	151.38	3580.77
4	Anhui	Furniture	Medium Box	168	176.51	13742.42
5	Anhui	Furniture	Paperbag	142	40.2	1279.760000
6	Anhui	Furniture	Small Box	384	103.32	316.63
7	Anhui	Furniture	Small bag	164	41.680000000000001	2297.92

## Sort columns

This function allows you to sort columns into different groups. You must name the groups. Otherwise, only the column order is changed.

Sort Columns

- province
- product\_type
- product\_box
- Order information
- order\_number
- shipping\_cost
- profit\_amt

Cancel Save

Conditional Formatting

profit\_amt

☒ Enable Conditional Formatting

Value  $\geq$  1000

Effect  $\uparrow$

Value < 1000 and  $\geq$  500

Effect  $\downarrow$

Value < 500

Effect  $\downarrow$

Sort Columns

☒ Show Totals

☒ Show at Bottom

Cross Table-company\_sales\_record\_en\_us

	province	product_type	product_box	Order information		profit_amt
				order_number	shipping_cost	
1	Anhui	Furniture	Huge Box	207	274.32	2865.04
2	Anhui	Furniture	Huge Paperbag	245	464.67000000000001	4920.25000
3	Anhui	Furniture	Large Box	116	151.38	3580.77
4	Anhui	Furniture	Medium Box	168	176.51	13742.42
5	Anhui	Furniture	Paperbag	142	40.2	1279.760000
469	Total			218871	110332.98999999999	1549090.0

- You can select the Show Totals check box to obtain the sum of specific or all data items. You can also select an aggregate function, as shown in the following figure.

**Note:**

To obtain the sum of specific data items, you must select the Merge Same Cells check box in the Display Settings area.

Cross Table-company_sales_record_en_us						
	province	product_type	product_box	Order information		profit_amt
				order_number	shipping_cost	
1	Anhui	Furniture	Huge Box	207	274.32	2865.04
2			Huge Paperbag	245	464.6700000000001	-4920.25000
3			Large Box	116	151.38	3580.77
4			Medium Box	168	176.51	13742.41
5			Paperbag	142	40.2	1279.76000
6			Small Box	384	103.32	316.61
7			Small bag	164	41.68000000000001	2297.91
8			Subtotal	1426	1252.0800000000001	19162.2891
9		Office	Huge Box	21	47.02	716.51
10			Large Box	128	94.95999999999998	2271.31
11			Medium Box	373	92.80999999999997	3027.61
12			Paperbag	636	139.59	-2957.58000
13			Small Box	2689	1249.6599999999999	15793.30999
14			Small bag	405	107.5	7006.01
15			Subtotal	4252	1731.5399999999998	25857.2491
16			Huge Box	134	166.5	-2267.49999
17			Huge Paperbag	46	30.06	252.71
18			Large Box	73	50.97	4714.11
19			Medium Box	130	40.64	649.030000
588	Total		218871	110332.98999999999	1549090.01	

Data

Style

Advanced

Display Settings ^

Form topic

Default

Short version

Show Row Numbers

Merge Same Cells

Freeze

Auto (Table Head)

Columns

From First Column

to

0

and

From

Last Column

to

0

Display paging

20

Article / Page

Functionality Settings ^

Conditional Formatting

profit\_amt

Enable Conditional Formatting

Sort Columns

Previous Operation

Show Totals

Show at Bottom

Field Settings

Select an option.

Select...

Show Subtotals

- In the Series Settings area, you can rename a field, select an alignment mode, and select a format to display values.

8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.

9. Click OK to save the dashboard.

To delete the table, click the More icon in the upper-right corner of the table and select Delete.

#### 4.4.4.12 Pivot tables

A pivot table can be used to display the summary of a field and perform data drilling in a tree structure. Dimensions determine rows, and measures determine columns in the table. Cells support multiple aggregate functions, including sum, average, count, max, and min.

Similar to *Cross tables*, a pivot table consists of rows and columns. You can specify an unlimited number of dimensions and measures to determine the rows and columns. Columns correspond to dimensions, such as province and product type. Rows correspond to measures, such as order amount and profit.

The following example uses the company\_sales\_record dataset to describe how to use a pivot table to show the packaging, order amount, and price of different products.

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Pivot Table icon. A pivot table is created in the display area of the dashboard.
5. On the Data tab page, select the target dimensions and measures.

In the Dimensions list, find and add the province, product\_type and product\_box dimensions to the Rows area. In the Measures list, find and add the order\_amt and price measures to the Values area.

**Note:**

Make sure you have converted the province dimension from String to Geo.

6. Click Update and the system then updates the table, as shown in the following figure.

Pivot Table-company\_sales\_record\_en\_us

province	order_number	order_amt
	7502.0	550702.0390000003
	3724.0	308833.1645000002
	3456.0	236946.91600000008
	6704.0	423084.69999999998
	32361.0	2241383.039
	16197.0	1190224.7685
	1453.0	78768.74100000001
	12088.0	818755.0745000005
	4589.0	278922.24899999995
	7626.0	528938.352

7. Click Save to save the dashboard.

To delete the table, click the More icon in the upper-right corner of the table and select Delete.

#### 4.4.4.13 Gauges

Similar to a dashboard in a car, a gauge clearly shows the value range of a metric.

It allows you to learn about the progress of a task, or the status of data metrics.

You can check whether a metric is still within the valid value range. For example,



you can use a gauge to show the inventory status of a commodity and replenish the inventory accordingly.

## Context

A gauge consists of the pointer angle and tooltip. The tooltip and pointer angle are determined by a measure, such as profits or discounts.

For each gauge, one and only one measure must be specified to determine the pointer angle and tooltip.

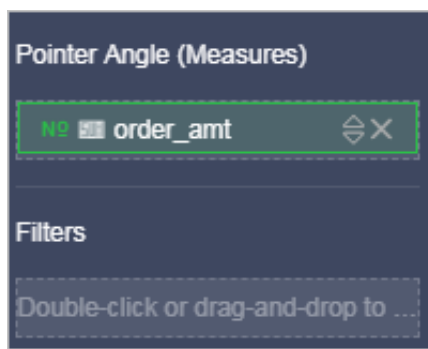
The following example uses the `company_sales_record` dataset to describe how to use a gauge to show the order price.

## Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Gauge icon.
5. On the Data tab page, select the target measure.

In the Measures list, find and add the price measure to the Pointer Angle area, as shown in *Figure 4-130: Select a field for the gauge.*

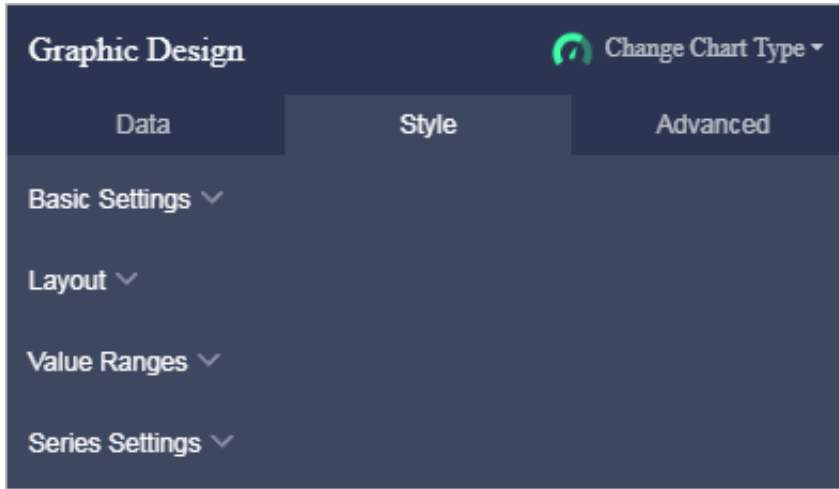
Figure 4-130: Select a field for the gauge



6. Click Update and the system then updates the gauge.

7. On the Style tab page, you can change the title, layout, and show or hide legend and tick marks, as shown in [Figure 4-131: Edit the gauge](#).

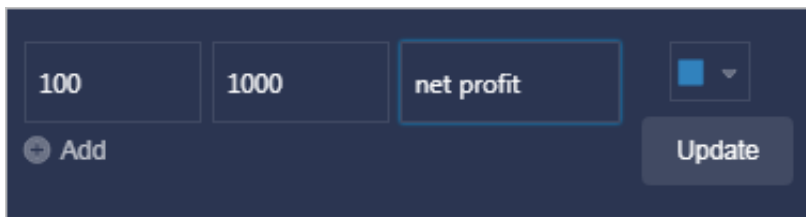
Figure 4-131: Edit the gauge



8. Click Add under the Value Ranges parameter, and specify the start value and end value.

For example, set the start value to 100, the end value to 1,000, and the range title to Net Profit, as shown in [Figure 4-132: Set value ranges](#).

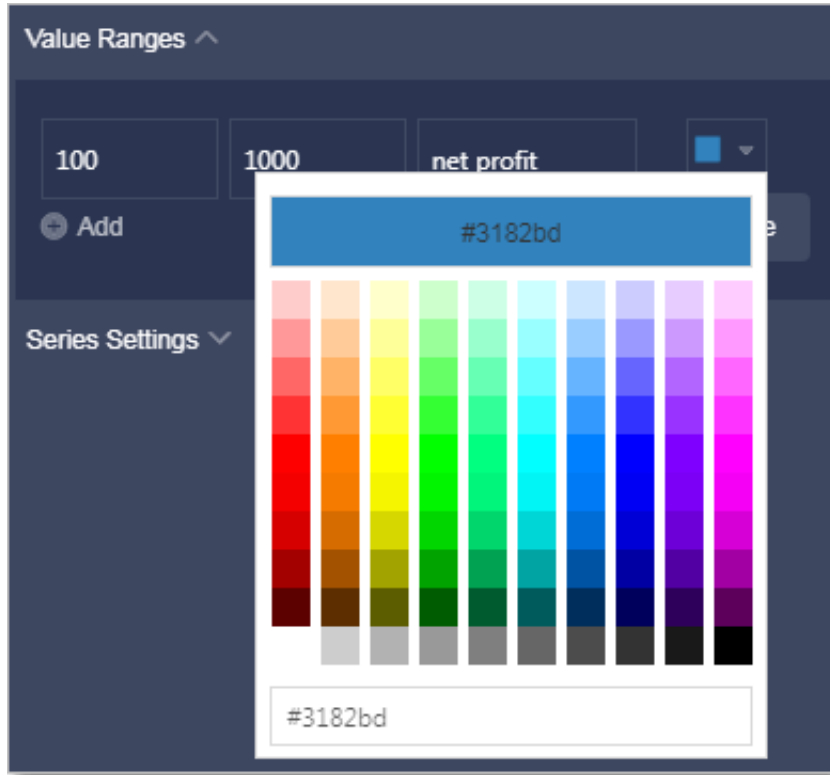
Figure 4-132: Set value ranges



9. Click the Color icon and select a color for the value range, as shown in [Figure 4-133](#):

*Change colors for value ranges.*

Figure 4-133: Change colors for value ranges



10. Click Update and the system then updates the gauge, as shown in [Figure 4-134](#): The gauge.

Figure 4-134: The gauge



11. Click **Save** in the upper-right corner and specify a name for the dashboard in the **Save Dashboard** dialog box that appears.

12. Click **OK** to save the dashboard.

To delete the gauge, click the **More** icon in the upper-right corner of the gauge and select **Delete**.

#### 4.4.4.14 Radar charts

A radar chart can be used to show numbers or ratios from analysis. It allows you to learn about the changes and trends of data metrics. For example, you can use a radar chart to show the sales volume in different regions.

##### Context

A radar chart consists of labels and label length. Labels are determined by dimensions, such as product type. The label length is determined by measures, such as shipping cost.

You must specify at least one dimension. You can specify up to two dimensions to determine labels. You must specify at least one measure to determine the label length.

The following example uses the `company_sales_record` dataset to describe how to use a radar chart to compare the order amount and price in different regions.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. On the **Datasets** page, find the `company_sales_record` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Radar Chart** icon.
5. On the **Data** tab page, select the target dimensions and measures.

In the **Dimensions** list, find and add the area dimension to the **Labels** area. In the **Measures** list, find and add the `order_amt` and `price` measures to the **Length** area, as shown in [Figure 4-135: Select fields for the radar chart](#).



**Note:**

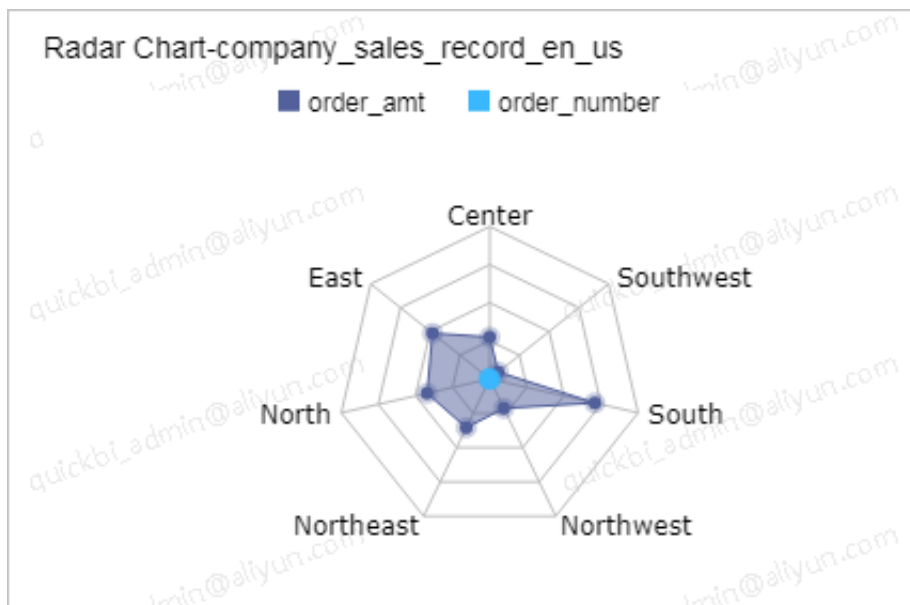
**Make sure that you have converted the area dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-135: Select fields for the radar chart



6. Click Update and the system then updates the chart.
7. On the Style tab page, you can change the title, layout, and legend mode of the radar chart, as shown in [Figure 4-136: The radar chart](#).

Figure 4-136: The radar chart



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.

**9. Click OK to save the dashboard.**

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.15 Scatter charts

A scatter chart can be used to show the distribution and aggregation status of data.

##### Context

A scatter chart consists of the X axis and Y axis. The color legend in a scatter chart is determined by a dimension, such as product type. The X axis and Y axis are determined by measures.

Only one dimension can be specified to determine the color legend. The maximum number of members in the dimension is 1,000.

You must specify at least one measure. You can specify up to three measures for the X axis.

One and only one measure must be specified for the Y axis.

The following example uses the company\_sales\_record dataset to describe how to use a scatter chart to compare the price and order amount of different products.

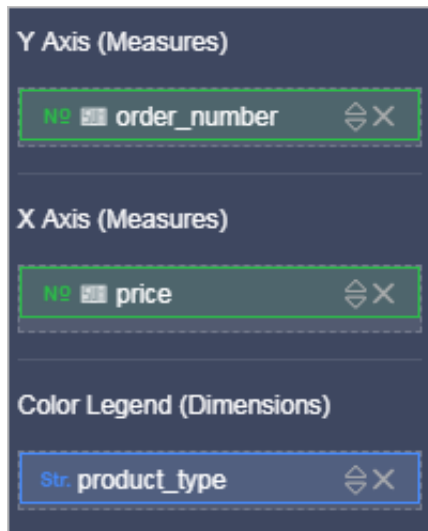
##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Scatter Chart icon.
5. On the Data tab page, select the target dimension and measures.

In the Dimensions list, find and add the product\_type dimension to the Color Legend area. In the Measures list, find and add the price and order\_amt measures

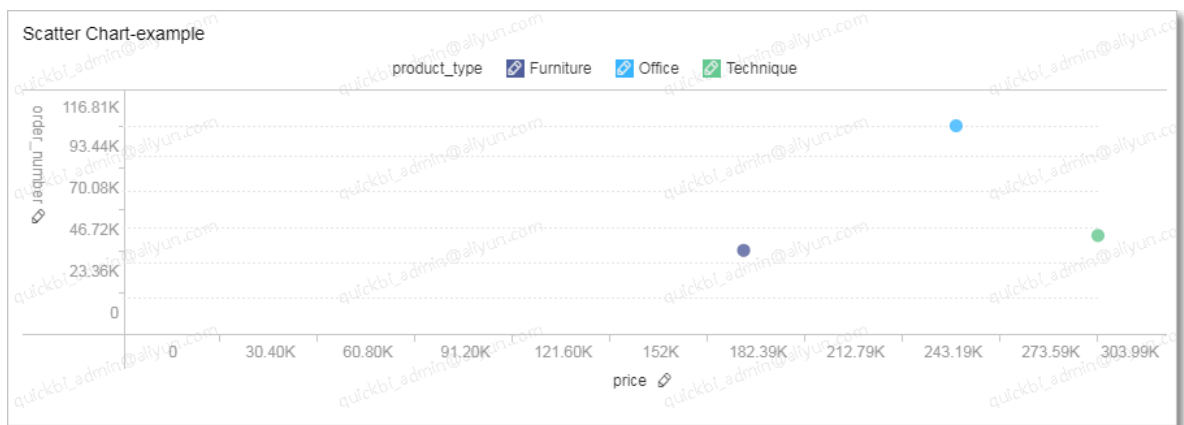
to the X Axis and Y Axis areas, respectively, as shown in [Figure 4-137: Select fields for the scatter chart](#).

Figure 4-137: Select fields for the scatter chart



6. Click Update and the system then updates the chart.
7. On the Style tab page, you can change the title, layout, and legend mode of the scatter chart, as shown in [Figure 4-138: The scatter chart](#).

Figure 4-138: The scatter chart



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.16 Bubble charts

A bubble chart shows data distribution status with locations and bubbles in different sizes.

##### Notes

A bubble chart consists of the X axis, Y axis, and bubbles in different sizes. You can specify only one dimension to determine the X axis, one measure to determine the Y axis, and one measure to determine the bubble size.

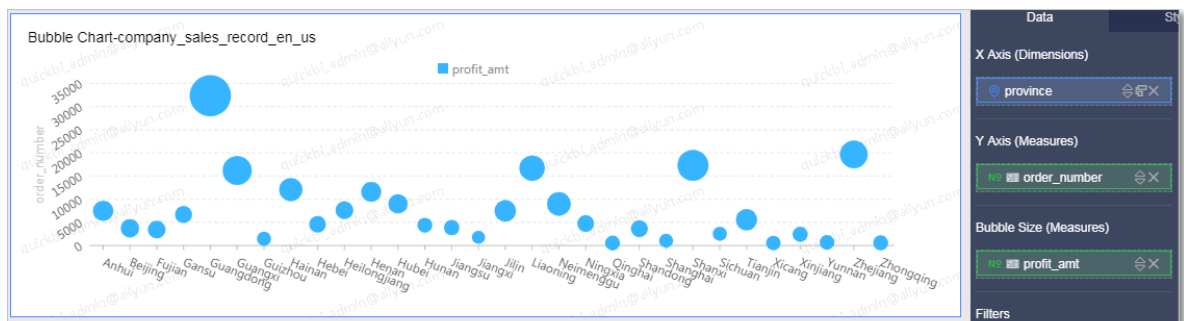
The following example uses the company\_sales\_record dataset to describe how to use a bubble map to compare the order amount and average profit in different provinces.

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the company\_sales\_record dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Bubble Map icon. A bubble map is created in the display area of the dashboard.
5. On the Data tab page, select the target dimension and measures.

In the Dimensions list, find and add the province dimension to the X Axis area.

In the Measures list, find and add the order\_amt to the Y Axis area, and the profit\_amt measure to the Bubble Size area. On the Style tab page, you can change the title, layout, and legend mode of the bubble map.

6. Click Update and the system then updates the map, as shown in the following figure.



7. Click Save in the upper-right corner to save the dashboard.

To delete the map, click the More icon in the upper-right corner of the map and select Delete.



#### 4.4.4.17 Funnel charts

A funnel chart can be used to analyze standard, long running, and multi-flow business procedures. By comparing business data from different flows, funnel charts allow you to explore and analyze problems. You can also use a funnel chart to display the conversion rate of each step. This enables you to perform flow analysis on business data that involves multiple flows. A funnel chart clearly displays the conversion rate, which represents the percentage of visitors who became paying customers of a website.

##### Context

A funnel chart consists of tier areas and tier labels. Tier labels are determined by a dimension, such as area. Tier areas are determined by a measure, such as order amount.

Only one dimension and one measure can be specified to determine tier labels and tier areas, respectively.

The following example uses the `company_sales_record` dataset to describe how to use a funnel chart to compare the order amount in different regions.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Funnel Chart icon.
5. On the Data tab page, select the target dimension and measure.

In the Dimensions list, find and add the area dimension to the Tier Labels area.

In the Measures list, find and add the `order_amt` measure to the Tier Area area, as shown in [Figure 4-139: Select fields for the funnel chart.](#)



**Note:**

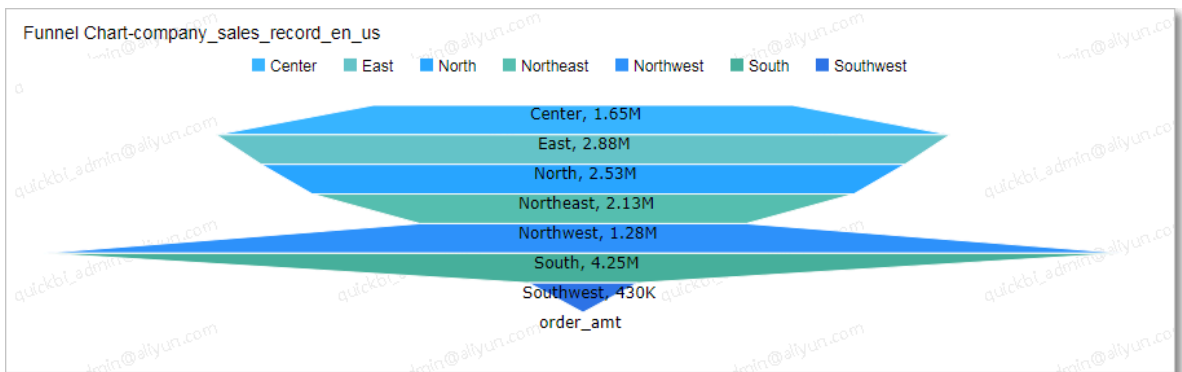
**Make sure that you have converted the area dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-139: Select fields for the funnel chart



6. Click Update and the system then updates the chart.
7. On the Style tab page, you can change the title, layout, and legend mode of the funnel chart, as shown in [Figure 4-140: The funnel chart](#).

Figure 4-140: The funnel chart



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.18 Kanban

A kanban directly displays data and helps you learn about the sales or operation status of your business. It helps you make solutions based on the data. Therefore, a kanban is an easy way to discover and fix problems.

##### Context

It consists of metrics and labels. Labels are determined by a data dimension, such as area. Metrics are determined by data measures, such as order amount and price.

For each kanban, one and only one dimension must be specified to determine the labels. You must specify at least one measure to determine the metrics. You can specify up to 10 measures.

The following example uses the `company_sales_record` dataset to describe how to use a kanban to compare the order amount, order price, shipping cost, and profits in different provinces.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. Find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Kanban icon.
5. On the Data tab page, select the target dimension and measures.

In the Dimensions list, find and add the province dimension to the Labels area.

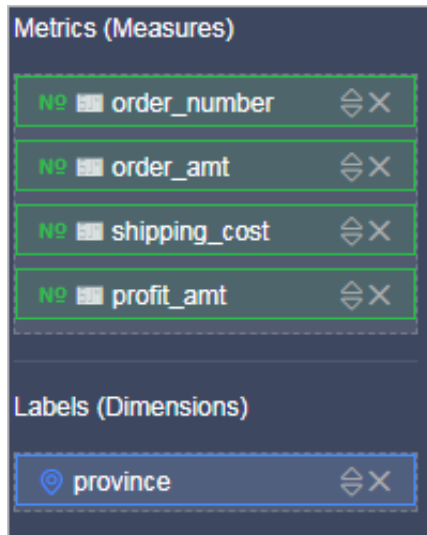
In the Measures list, find and add the `order_amt`, `price`, `shipping_cost`, and `profit_amt` measures to the Metrics area, as shown in [Figure 4-141: Select fields for the kanban](#).



**Note:**

**Make sure that you have converted the province dimension from String to Geo.**  
For more information about converting dimensions to another type, see [Edit a dimension](#).

Figure 4-141: Select fields for the kanban



6. Click Update and the system then updates the kanban.

7. On the Style tab page, set the Columns Allowed parameter to 3, as shown in [Figure 4-142: The kanban](#).

Figure 4-142: The kanban

Kanban-company_sales_record_en_us			
<b>Anhui</b> order number <b>7.5K</b> order_amt 551K shipping_cost 3.82K profit_amt 58.7K	<b>Beijing</b> order number <b>3.72K</b> order_amt 309K shipping_cost 1.92K profit_amt 42K	<b>Fujian</b> order number <b>3.46K</b> order_amt 237K shipping_cost 1.58K profit_amt 35.3K	<b>Gansu</b> order number <b>6.7K</b> order_amt 423K shipping_cost 3.72K profit_amt 30.2K
<b>Guangdong</b> order number <b>32.4K</b> order_amt 2.24M shipping_cost 15.9K profit_amt 247K	<b>Guangxi</b> order number <b>16.2K</b> order_amt 1.19M shipping_cost 8.45K profit_amt 137K	<b>Guizhou</b> order number <b>1.45K</b> order_amt 78.8K shipping_cost 711 profit_amt 2.94K	<b>Hainan</b> order number <b>12.1K</b> order_amt 819K shipping_cost 6.11K profit_amt 85.8K
<b>Hebei</b> order number <b>4.59K</b> order_amt 279K shipping_cost 2.41K profit_amt 26.3K	<b>Heilongjiang</b> order number <b>7.63K</b> order_amt 529K shipping_cost 3.87K profit_amt 36.5K	<b>Henan</b> order number <b>11.6K</b> order_amt 734K shipping_cost 6.11K profit_amt 57.2K	<b>Hubei</b> order number <b>9.01K</b> order_amt 674K shipping_cost 4.72K profit_amt 48.8K

8. Click **Save** in the upper-right corner and specify a name for the dashboard in the **Save Dashboard** dialog box that appears.
9. Click **OK** to save the dashboard.

To delete the kanban, click the **More** icon in the upper-right corner of the kanban and select **Delete**.

#### 4.4.4.19 Treemaps

A treemap can be used to compare the proportion of metrics of an object.

##### Context

It displays rectangle labels in different rectangle sizes based on the measure.

Rectangle labels are determined by data dimensions, such as packaging. The size of each rectangle label is determined by a data measure, such as shipping cost.

For each treemap, one and only one dimension must be specified to determine rectangle labels. One and only one measure can be specified to determine the rectangle size.

The following example uses the `company_sales_record` dataset to describe how to use a treemap to compare the order amount of different products.

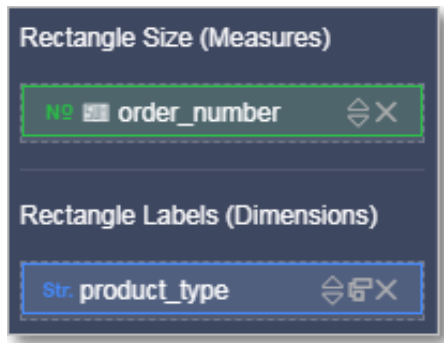
##### Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click **Datasets**.
3. Find the `company_sales_record` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Treemap** icon.

5. On the Data tab page, select the target dimension and measure.

In the Dimensions list, find and add the `product_type` dimension to the Rectangle Labels area. In the Measures list, find and add the `order_amt` measure to the Rectangle Size area, as shown in [Figure 4-143: Select fields for the treemap](#).

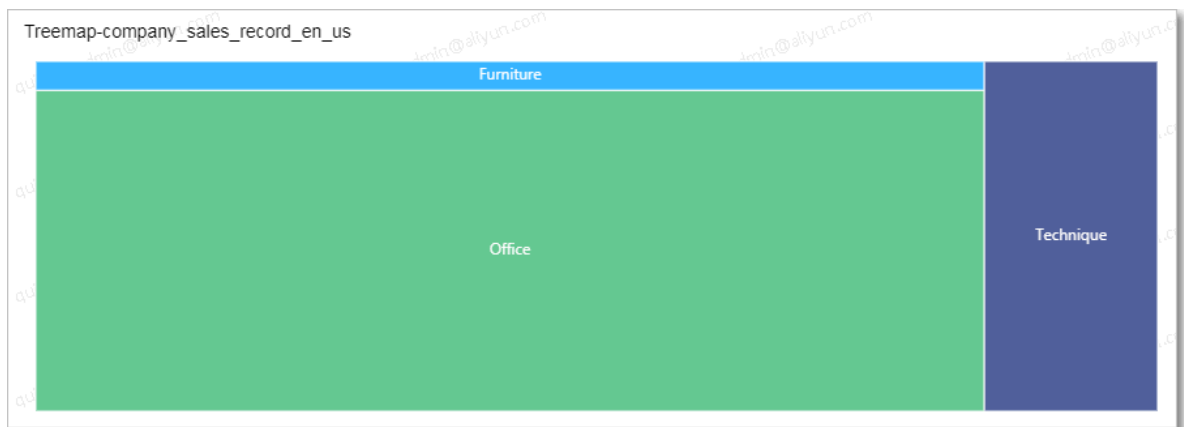
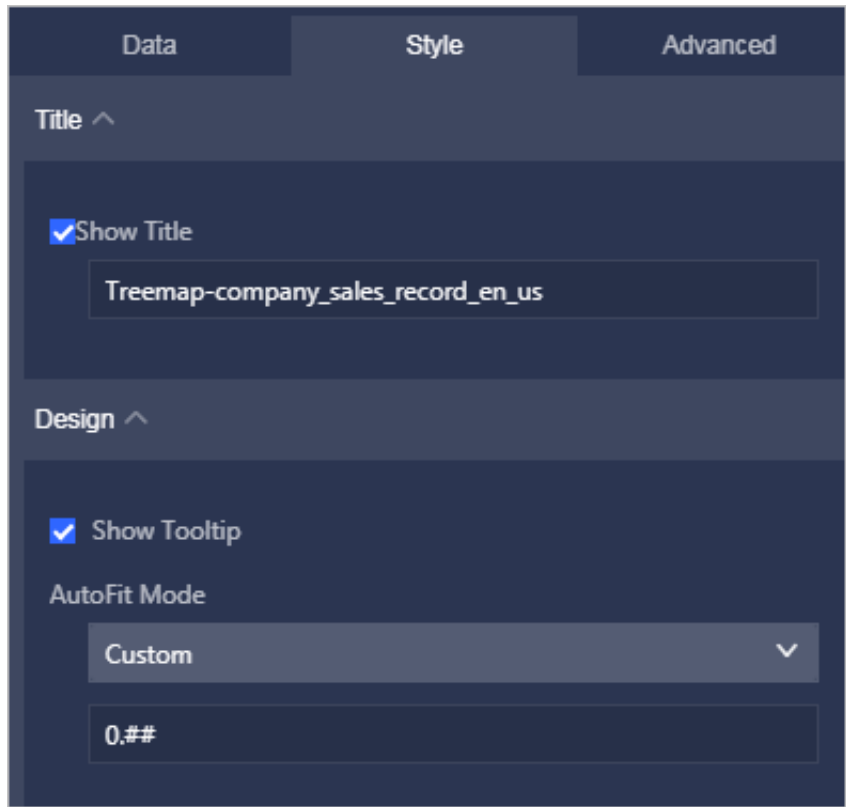
Figure 4-143: Select fields for the treemap



6. Click Update and the system then updates the treemap.

7. On the Style tab page, you can change the title and layout of the treemap, as shown in *Figure 4-144: The treemap*.

Figure 4-144: The treemap



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the treemap, click the More icon in the upper-right corner of the treemap and select Delete.

#### 4.4.4.20 Polar diagrams

A polar diagram can be used to display data changes over time or compare metric values. It is suitable for comparing data from different objects, for example, compare data across different regions.

##### Context

Similar to a [pie chart](#), a polar diagram consists of multiple slices. Slice labels are determined by data dimensions, such as area and product type. The arc radius of each slice is determined by data measures, such as order amount and order price.

For each polar diagram, one and only one dimension must be specified to determine slice labels. This dimension must contain 3 to 12 members. One and only one measure must be specified to determine the arc radius.

The following example uses the `company_sales_record` dataset to describe how to use a polar diagram to compare the order amount in different regions. The number of regions must be greater than 3 and less than or equal to 12.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Datasets.
3. Find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Polar Diagram icon.
5. On the Data tab page, select the target dimension and measure.

In the Dimensions list, find and add the area dimension to the Label area. In the Measures list, find and add the `order_amt` measure to the Arc Radius area, as shown in [Figure 4-145: Select fields for the polar diagram](#).

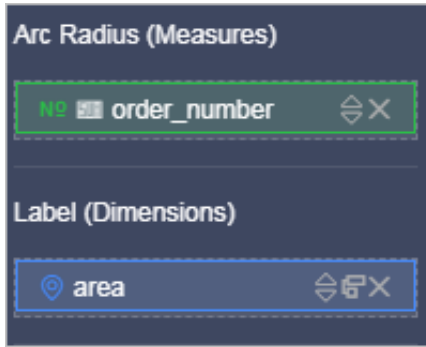


**Note:**



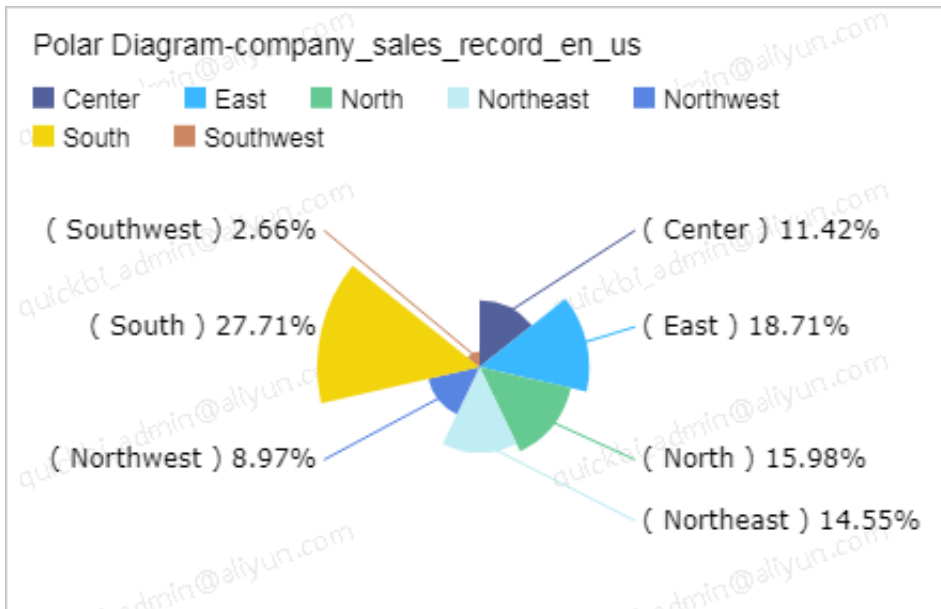
**Make sure that you have converted the area dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-145: Select fields for the polar diagram



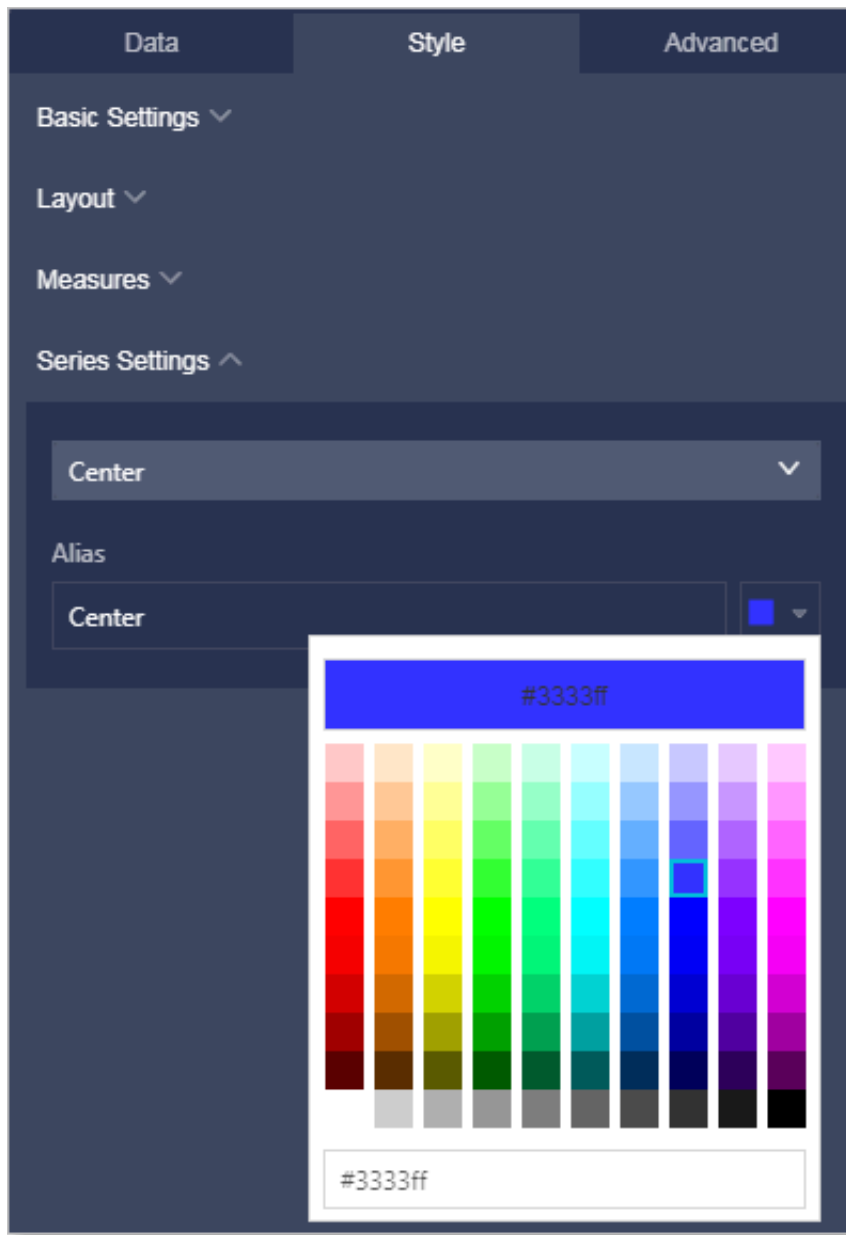
6. Click Update and the system then updates the polar diagram.
7. On the Style tab page, you can change the title and layout of the polar diagram, as shown in [Figure 4-146: The polar diagram](#).

Figure 4-146: The polar diagram



**8. To change legend colors, choose Style > Series Settings.**

Figure 4-147: Change legend colors



**9. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.**

**10. Click OK to save the dashboard.**

To delete the polar diagram, click the More icon in the upper-right corner of the polar diagram and select Delete.

#### 4.4.4.21 Word clouds

A word cloud displays the frequency of words that appear in a dataset. It is suitable for creating user personas and user tags.

##### Context

A word cloud displays words in different sizes based on the frequency of use. Words are determined by data dimensions, such as customer name and product type. The size of each word is determined by a data measure, such as profit or price.

For each word cloud, one and only one dimension and measure must be specified.

The following example uses the `company_sales_record` dataset to describe how to use a word cloud to compare the order amount in different provinces.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Datasets.
3. Find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Word Cloud icon.
5. On the Data tab page, select the target dimension and measure.

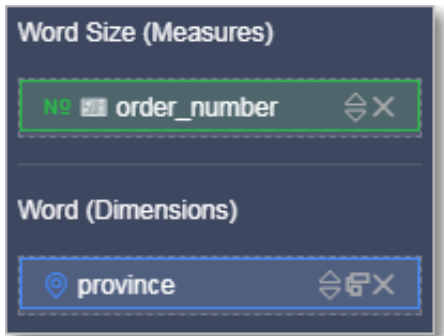
In the Dimensions list, find and add the province dimension to the Word area. In the Measures list, find and add the `order_amt` measure to the Word Size area, as shown in [Figure 4-148: Select field for the word cloud.](#)



**Note:**

**Make sure you have converted the province dimension from String to Geo.**  
**For more information about converting dimensions to another type, see [Edit a dimension](#).**

Figure 4-148: Select field for the word cloud



6. Click Update and the system then updates the word cloud.
7. On the Style tab page, you can change the title of the word cloud, as shown in [Figure 4-149: The word cloud](#).

Figure 4-149: The word cloud



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the word cloud, click the More icon in the upper-right corner of the word cloud and select Delete.

#### 4.4.4.22 Tornado-leaned funnel charts

A tornado-leaned funnel chart is the combination of a tornado chart and a funnel chart. Tornado-leaned funnel charts can be used to compare different metrics

between two objects, for example, the income and education levels between residents in two cities. Funnel charts can be used to show the conversion rates between stages of the business process and are suitable for business process analysis. Funnel charts allow you to learn about the percentage of visitors who became paying customers.

## Context

A tornado-leaned funnel chart combines the features of tornado charts and funnel charts. For example, when you compare the percentage of the migrant population, employment rate, and commercial housing transactions in Beijing and Shanghai, if a conversion relation exists between the items being compared, the tornado-leaned funnel chart can show the difference between metrics, and also display the conversion rates between the items.

If no conversion relationship exists, the tornado-leaned funnel chart functions the same as a tornado chart. If a conversion relation exists between the items being compared and only one metric is defined, the chart functions the same as a funnel chart.

A tornado-leaned funnel chart consists of items and metrics to be compared. Items are determined by a data dimension, such as area or product type. Metrics are determined by data measures, such as order quantity and order amount.

For each tornado-leaned funnel chart, one and only one dimension must be specified to determine the items to be compared. At least one measure must be specified to determine metrics.

The following example uses the `company_sales_record` dataset to describe how to use a tornado-leaned funnel chart to compare the order quantity, profits, and average profit of different products.

## Procedure

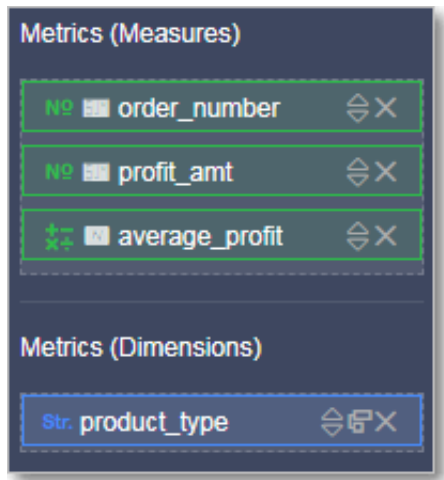
1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Datasets.
3. On the Datasets page, find the `company_sales_record` dataset, click the Create Dashboard icon in the Actions column, and select Standard.
4. On the dashboard edit page, click the Tornado-Leaned Funnel Chart icon.

5. On the Data tab page, select the target dimensions and measures.

In the Dimensions list, find and add the `product_type` dimension to the Metrics (Dimensions) area. In the Measures list, find and add the `order_amt`, `profit_amt`, and `average_profit` measures to the Metrics (Measures) area, as shown in [Figure 4-150: Select fields for the tornado-leaned funnel chart](#).

*4-150: Select fields for the tornado-leaned funnel chart.*

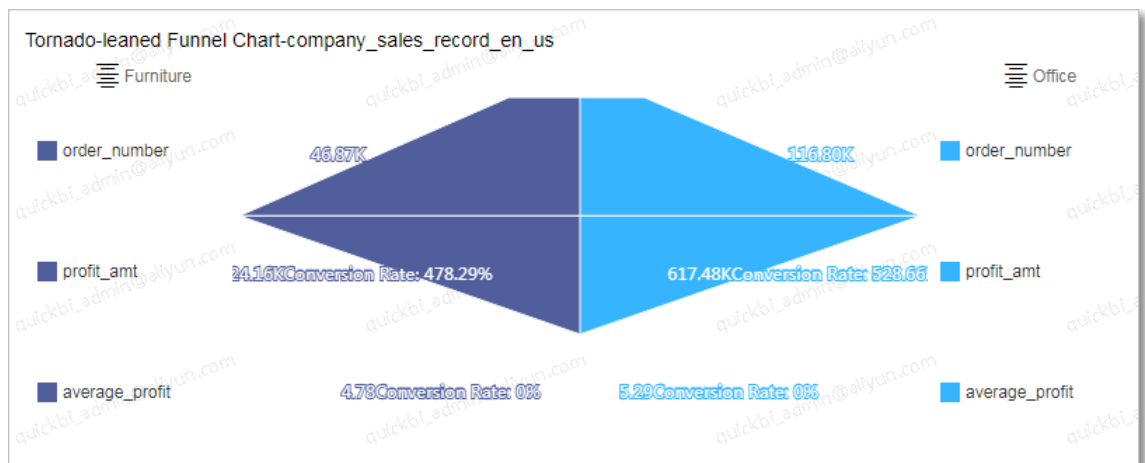
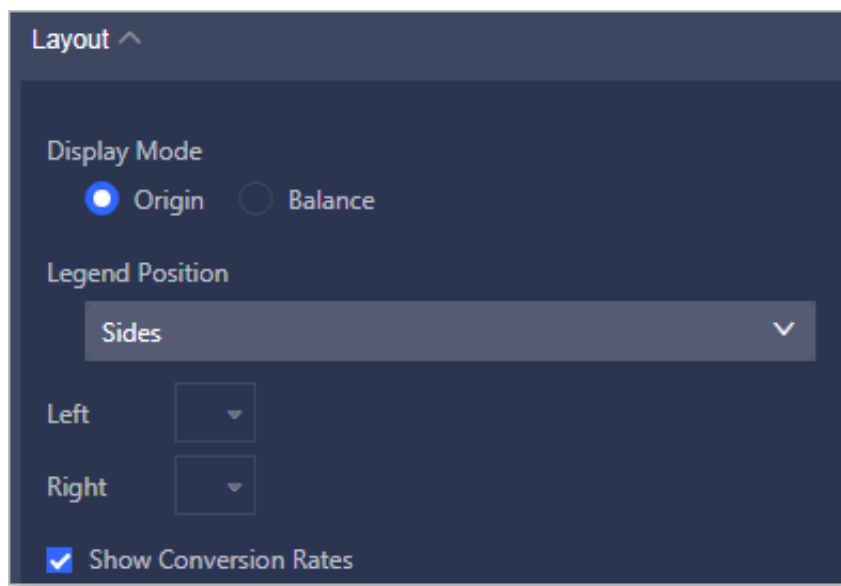
Figure 4-150: Select fields for the tornado-leaned funnel chart



6. Click Update and the system then updates the chart.

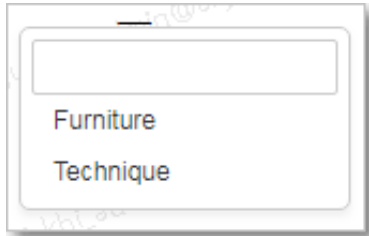
7. On the Style tab page, you can change the title, layout, legend position, background color, and hide or show the conversion rate.
  - a. Quick BI provides two types of layouts for tornado-leaned funnel charts. Select the layout based on your actual needs.
  - b. In the Layout area of the Style tab, you can change the legend position, background color, and hide or show the conversion rate, as shown in the following figure.

Figure 4-151: The tornado-learned funnel chart



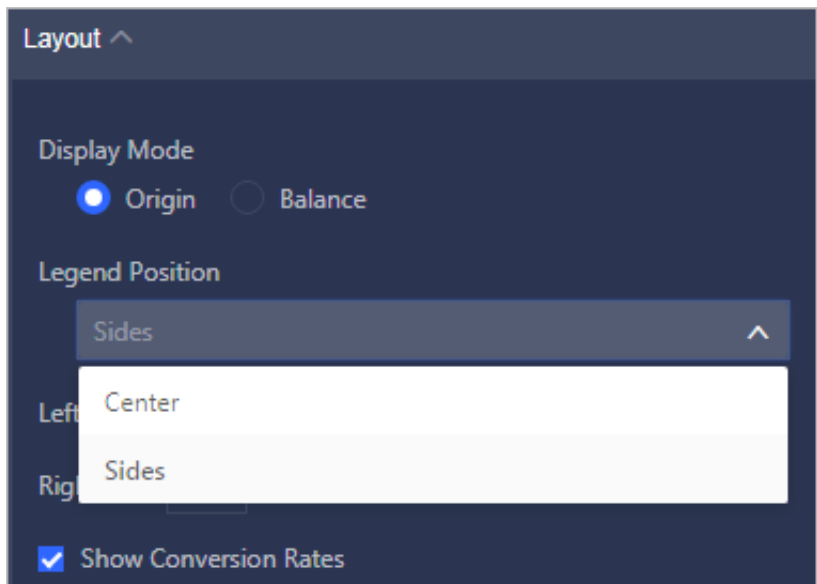
- You can hover over the product type field to switch to another product, as shown in the following figure.

Figure 4-152: Switch products



- You can change the legend position, as shown in the following figure.

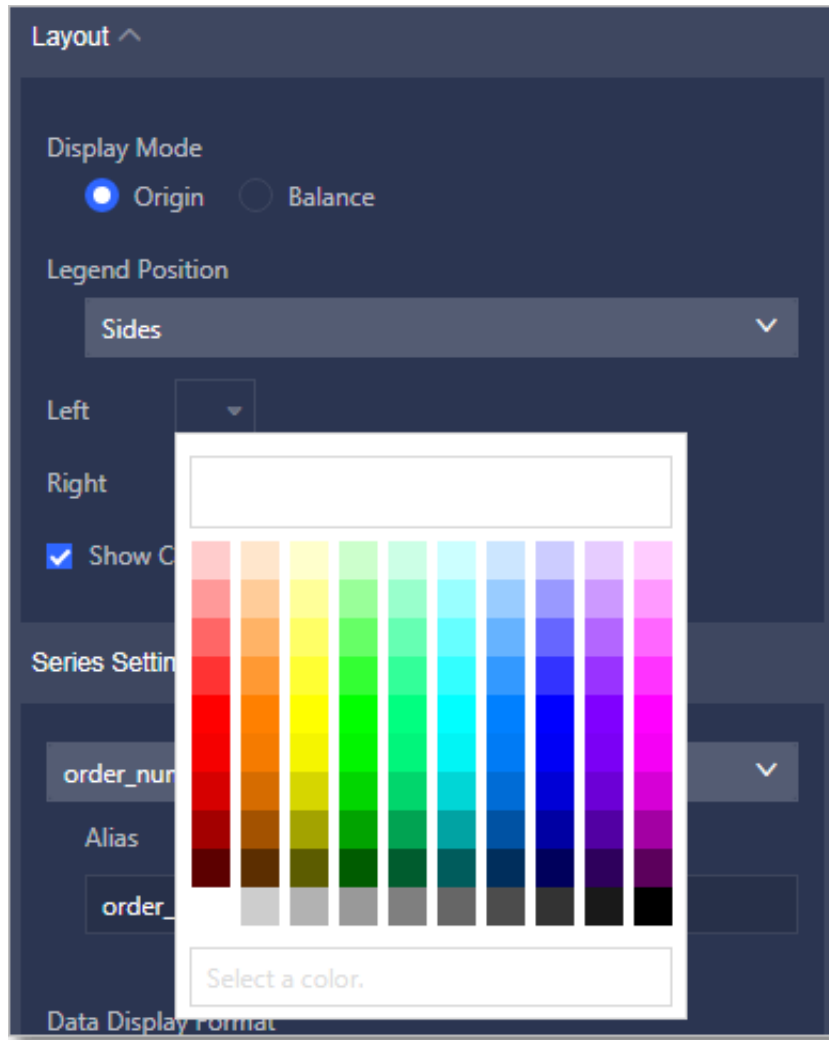
Figure 4-153: Change the legend position





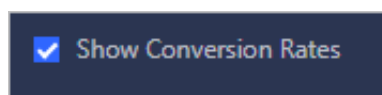
- You can click the Color icon next to Left or Right and select a color from the drop-down list, as shown in the following figure.

Figure 4-154: Change colors



- You can hide or show the conversion rate, as shown in the following figure.

Figure 4-155: Hide or show the conversion rate



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.23 Hierarchy charts

A hierarchy chart uses the tree structure to organize and display hierarchical data. It is an implementation of the enumeration method. For example, when you review revenues of the cities in a province, the relationships between the province and cities can be displayed in a hierarchical structure. Hierarchy charts are used to analyze data related to organizational structures, for example, the staff structure of a company or department structure of a hospital.

##### Context

A hierarchy chart consists of node metrics and node labels. Node labels are determined by data dimensions, such as area and product type. Node metrics are determined by data measures, such as order quantity and order amount.

This topic uses examples based on the following scenarios to describe how to use a hierarchy chart, including the usage of the filter:

- **Scenario 1:** Compare the order quantity of different products in provinces in different regions.
- **Scenario 2:** View the average profit of different products in different municipalities.

In a hierarchy chart, you must select at least two dimensions as the node labels. Data will be displayed more clearly if these dimensions have a hierarchical relationship. You must select one measure as the node metric.

**Scenario 1: Compare the order quantity of different products in provinces in different regions**

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. Find the `company_sales_record` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Hierarchy Chart** icon.
5. On the **Data** tab page, select the target dimensions and measures.

In the **Dimensions** list, find and add the `area`, `province`, and `product_type` dimensions to the **Node Labels** area. The sequence of the dimensions determines the hierarchical relationship in the chart. In the **Measurement** list, find and

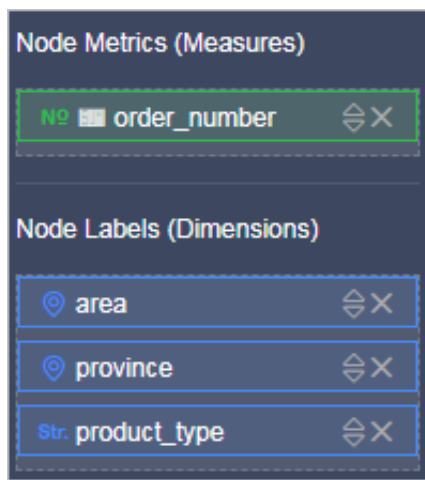
add the `order_amt` measure to the Node Metrics area, as shown in the following figure.



**Note:**

Make sure that the type of the area and province dimensions has been converted from String to Geo. For more information about converting a dimension to another type, see [Edit a dimension](#).

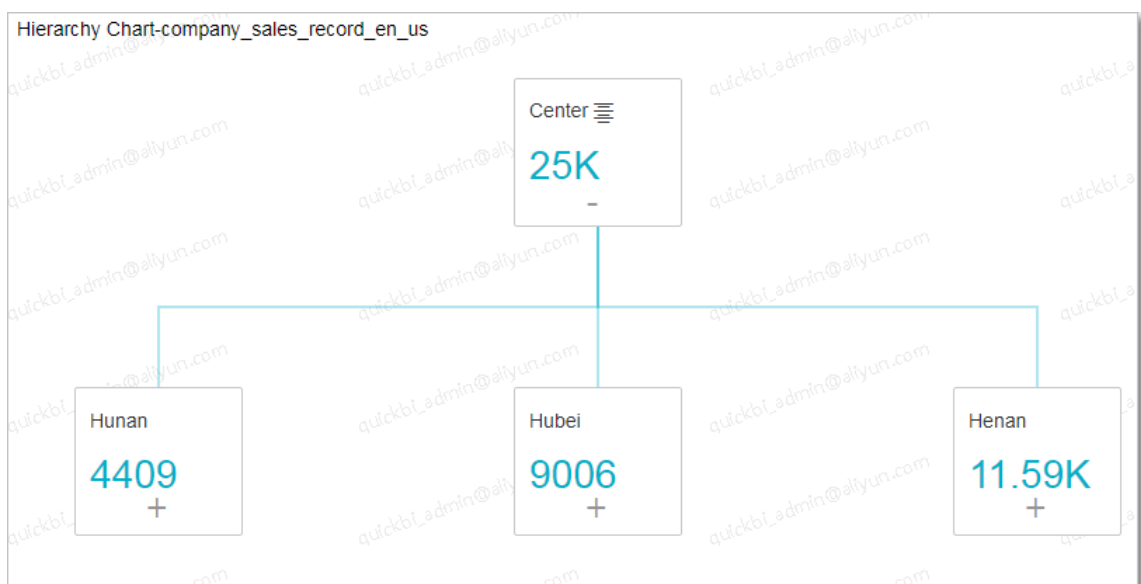
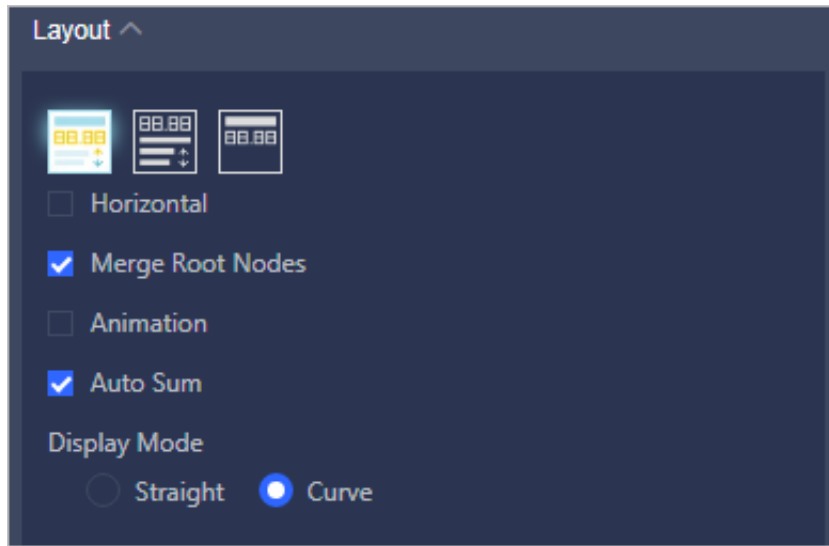
Figure 4-156: Select fields for the hierarchy chart



6. Click Update and the system then updates the chart.
7. On the Style tab page, you can change the title, layout, and design of the chart.
  - a. Quick BI provides three types of layouts for hierarchy charts. You can select a structure and mode to display the chart based on your needs. The Merge Root

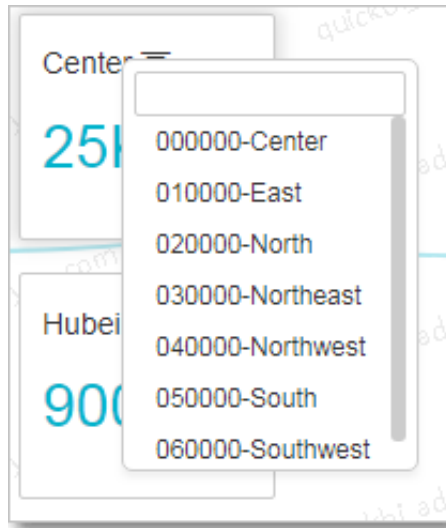
**Nodes check box is selected by default. In the following example, the Straight mode is selected.**

Figure 4-157: Chart layouts



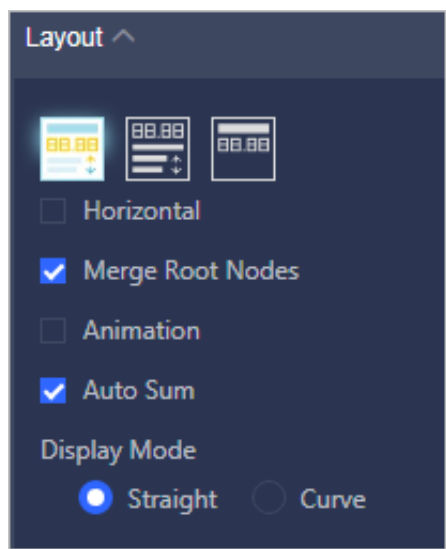
- You can hover over the region field and switch to another region from the drop-down list that appears, as shown in the following figure.

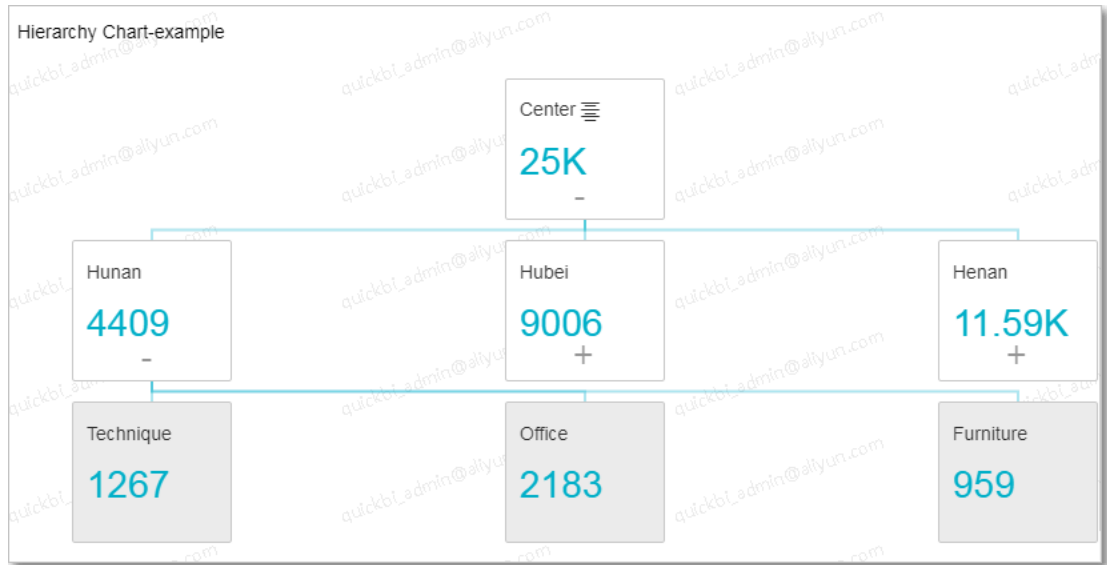
Figure 4-158: Switch regions



- You can click the minus sign (-) or plus sign (+) to hide or show the child nodes.
- In the Layout area, if you select Auto Sum, the chart automatically displays the value of the total amount in the parent node, as shown in the following figure.

Figure 4-159: Auto Sum



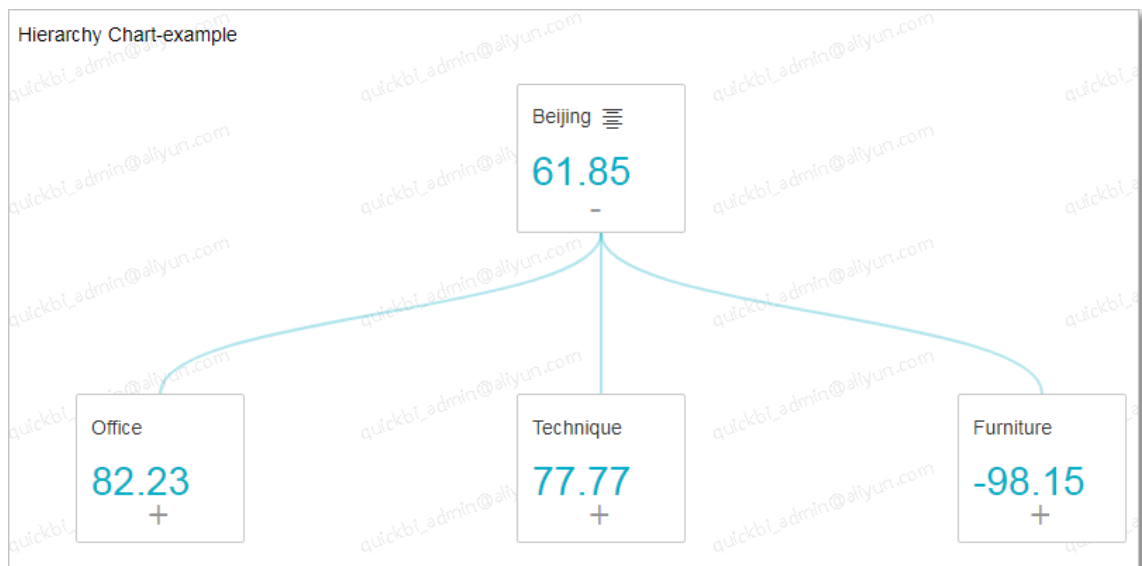
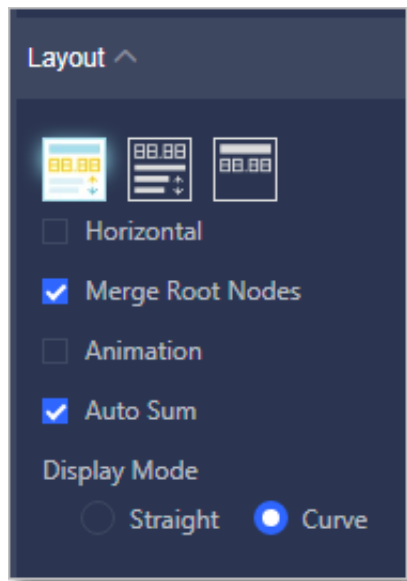


- b. You can edit the levels of the hierarchy by selecting or clearing the All Items check box, or specifying a value for the Levels parameter. You can select a field to be the primary path. The primary path is displayed in the chart in a different color. You can also select the Show Filter Bar check box to add a**

toolbar to the chart. This allows you to edit the chart in the preview mode or in the dashboard.

In the following example, the Primary Path parameter is set to order\_amt, the Sort parameter is set to Ascend, the Show Filter Bar check box is selected, and the Curve mode is selected, as shown in the following figure.

Figure 4-160: The hierarchy chart



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard Dialog box that appears.
9. Click OK to save the dashboard.

By default, the dashboard is saved to My Items on the Dashboards page.

## Scenario 2: Display the average profit of different products in different municipalities

### Context

Data modeling may be required in this scenario. For more information about data modeling, see [Add a calculated field](#).

### Procedure

1. In the Dimensions list, find and drag the province dimension to the Filters area.

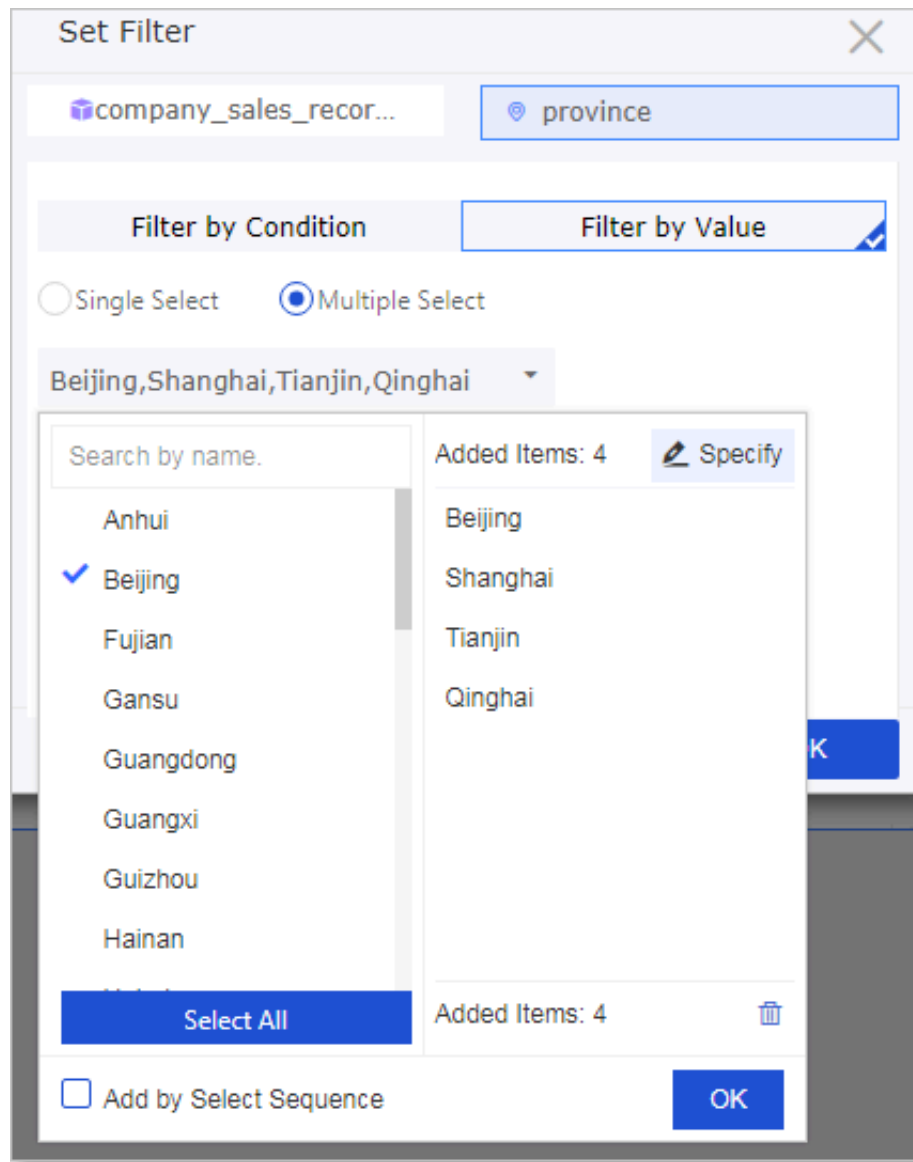
This allows you to filter provinces.



2. Click the Filter icon and select Filter By Value, as shown in the following figure.

The system automatically lists all available options of the province field in the drop-down list.

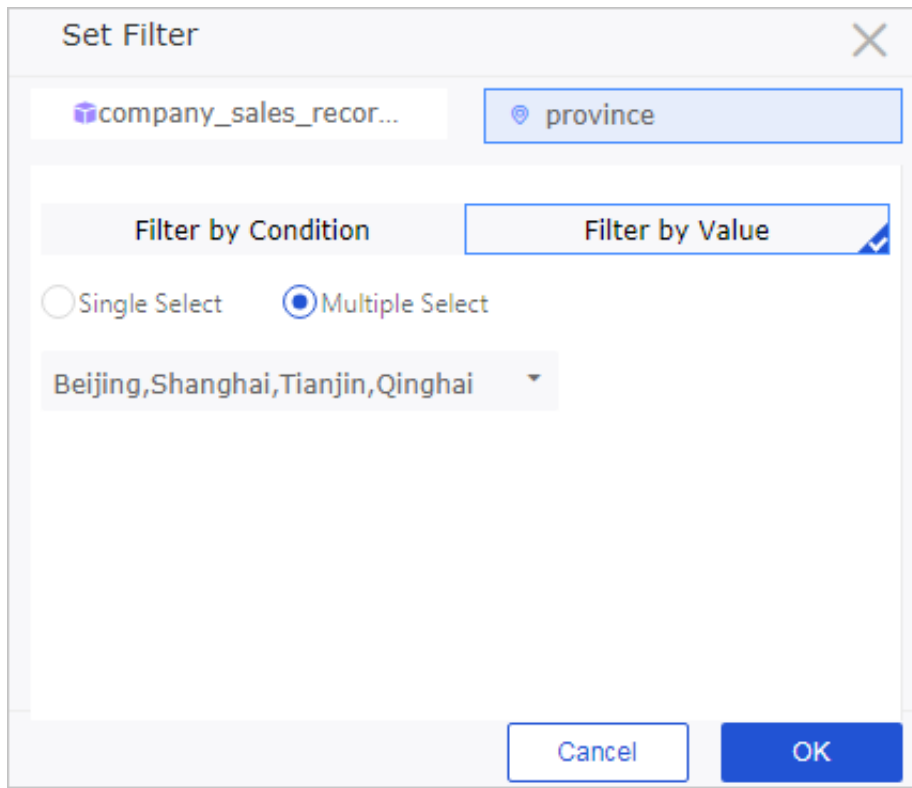
Figure 4-161: Filter by value



3. Select the target municipalities or manually enter the municipality names.

4. Click OK to set the filter condition, as shown in the following figure.

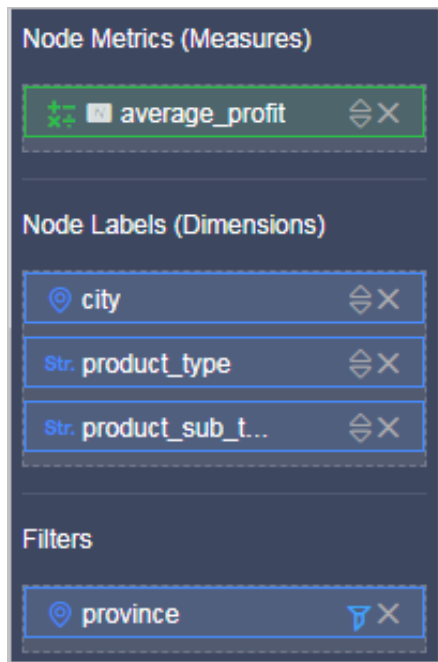
Figure 4-162: Set the filter condition



5. In the Dimensions list, find and add the city, product\_type, and product\_sub\_type dimensions to the Node Labels area, as shown in the following figure.

The sequence of these dimensions determines the hierarchical relationship displayed in the chart. In the Measurement list, find and add the average\_profit measure to the Node Metrics area.

Figure 4-163: Select fields for the hierarchy chart



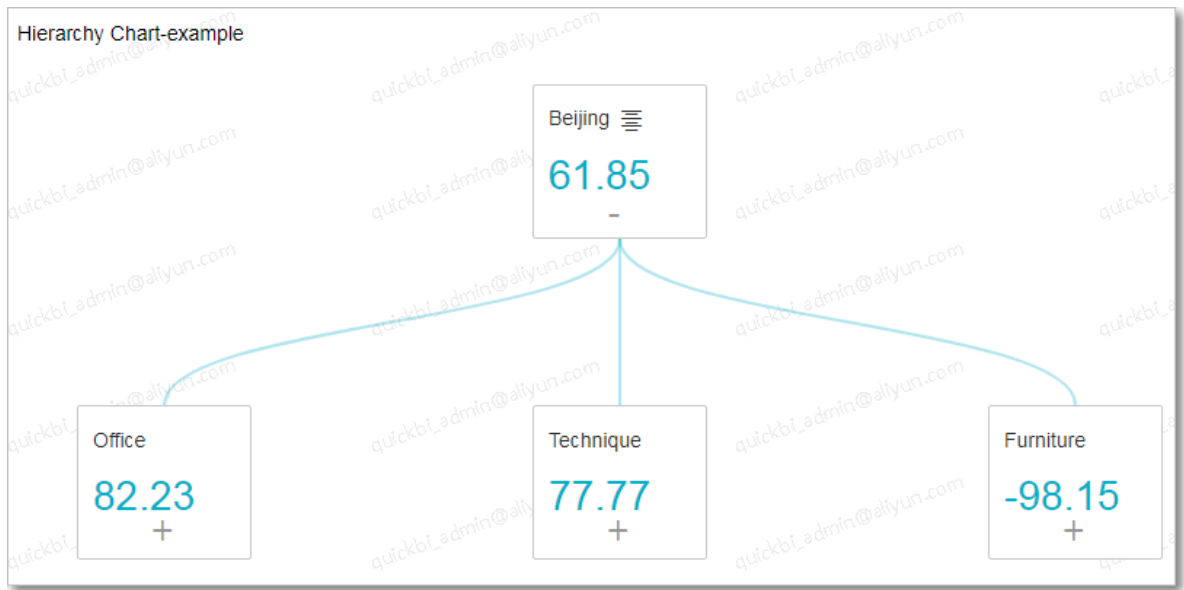
6. Click Update and the system then updates the chart.

7. On the Style tab page, you can edit the title, layout, and design of the chart, as shown in the following figure.

Figure 4-164: The hierarchy chart

The screenshot shows the 'Style' tab of a configuration interface for a hierarchy chart. The interface is divided into three main sections: Title, Layout, and Design.

- Title:** Includes a 'Show Title' checkbox (checked) and a text input field containing 'Hierarchy Chart-example'.
- Layout:** Includes three icons for different chart styles. Below the icons are four checkboxes: 'Horizontal' (unchecked), 'Merge Root Nodes' (checked), 'Animation' (unchecked), and 'Auto Sum' (checked). There is also a 'Display Mode' section with two radio buttons: 'Straight' (unchecked) and 'Curve' (checked).
- Design:** Includes a 'Levels' section with a checkbox 'All Items' (unchecked) and a text input field containing '2'. Below this is a 'Primary Path' dropdown menu set to 'None'. There is also a 'Sort' dropdown menu. At the bottom are three checkboxes: 'Highlight Primary Path' (unchecked), 'Highlight Bounce Path' (checked), and 'Show Filter Bar' (unchecked).



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.4.24 Flow analysis charts

A flow analysis chart uses metrics such as page visits, page views (PV), and unique visitors (UV) to calculate the conversion rate of your website. This helps you understand the overall performance of marketing campaigns and measure the sales volume of certain products. Flow analysis charts are suitable for analyzing digital marketing campaigns and e-commerce websites. For example, you can use flow analysis charts to find out which products are in great demand and what are the peak hours of your business.

##### Context

Currently, flow analysis charts support the following dimensions: previous page, current page, and next page, and the following measures: PV, UV, conversion rate, and bounce rate. You need to specify the PV or UV for the previous, current, and next pages.

For each flow analysis chart, one and only one dimension must be specified for each of the three pages. The dimensions must have a hierarchical relationship. The sequence of the dimensions determines the hierarchical relationship between pages in the chart. One and only one measure must be specified for each of the

PV and UV of the previous, current, and next pages, the conversion rate, and the bounce rate.

The three dimensions, the conversion rate, and the bounce rate are required fields . You can choose to specify only the PV or UV for the previous, current, and next pages. The system prompts an error message if you add the wrong dimension or measure.

The following example uses the `page_source_target_day_stat` dataset to describe how to use a flow analysis chart to demonstrate the conversion rate and bounce rate between pages based on PV.

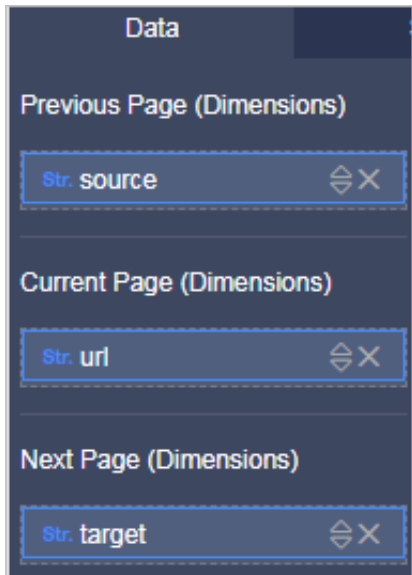
### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Datasets**.
3. Find the `page_source_target_day_stat` dataset, click the **Create Dashboard** icon in the **Actions** column, and select **Standard**.
4. On the dashboard edit page, click the **Flow Analysis** icon.
5. On the **Data** tab page, select the target dimensions and measures.

In the **Dimensions** list, find the target dimensions and add them to the **Previous Page**, **Current Page**, and **Next Page** areas, respectively. The sequence of these dimensions determines the hierarchical relationship in the chart. In the **Measures** list, find and add the target measures to the **Conversion Rate**, **Bounce**

**Rate, Previous Page PV, Current Page PV, Next Page PV, Previous Page UV, Current Page UV, and Next Page UV areas, respectively.**

Figure 4-165: Select fields for the flow analysis chart



Data

Previous Page (Dimensions)

Str. source

Current Page (Dimensions)

Str. url

Next Page (Dimensions)

Str. target

Previous Page PV (Measures)

N9 source\_pv

Previous Page UV (Measures)

Double-click or drag-and-drop t...

Current Page PV (Measures)

N9 url\_pv

Current Page UV (Measures)

Double-click or drag-and-drop t...

Next Page PV (Measures)

N9 target\_pv

Next Page UV (Measures)

Double-click or drag-and-drop t...

Conversion Rate (Measures)

N9 transfer\_rate1

Bounce Rate (Dimensions)

N9 transfer\_rate2

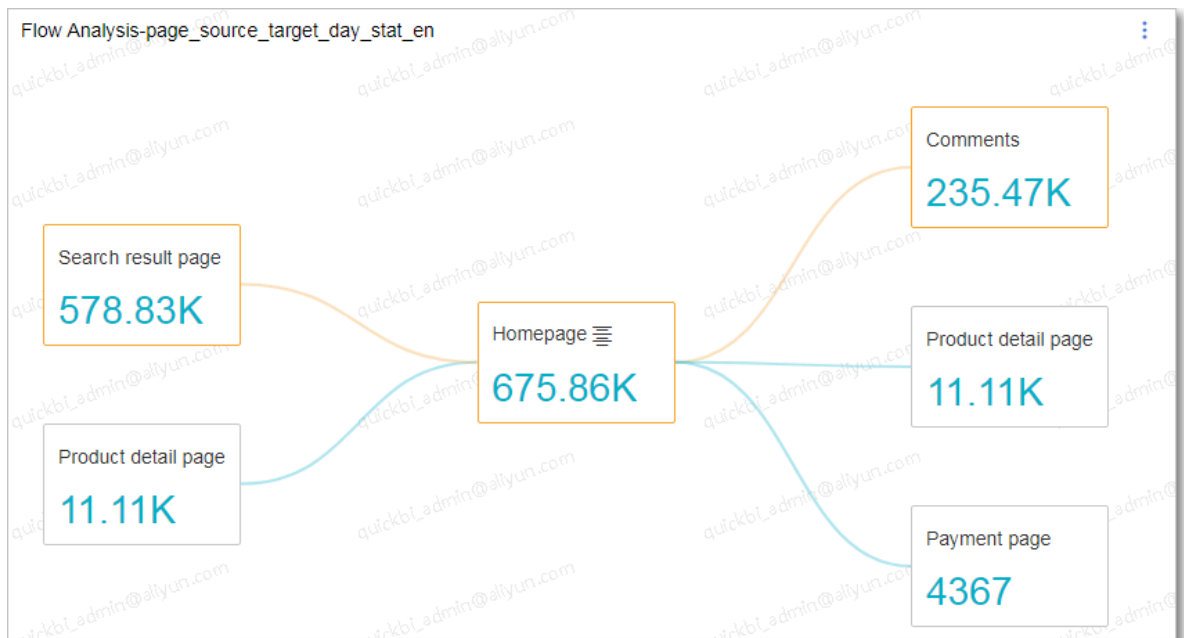
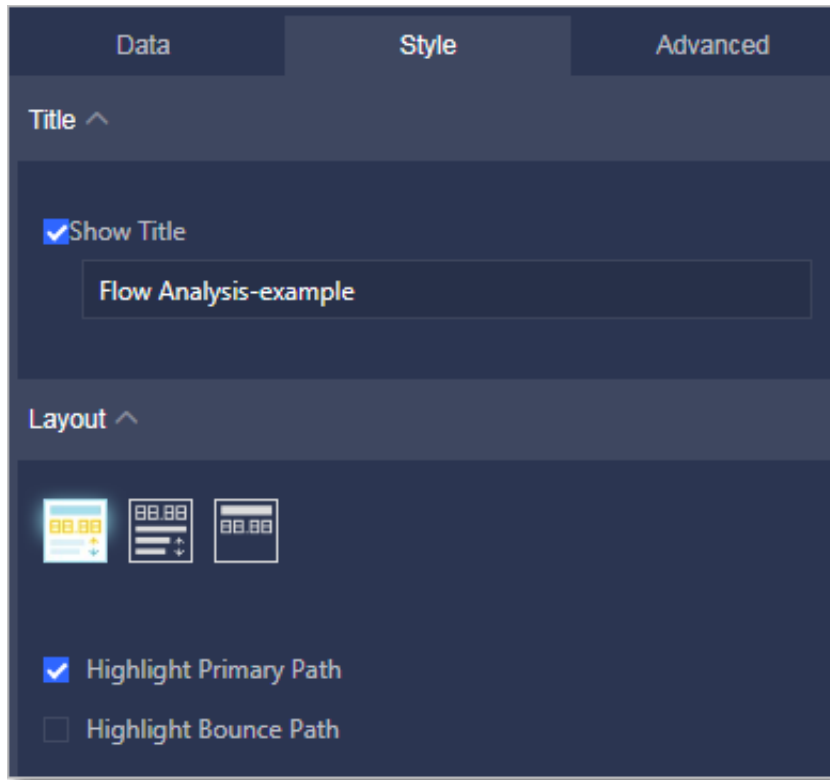


6. Click Update and the system then updates the flow analysis chart.
7. On the Style tab page, you can change the title and layout of the chart.

Quick BI provides three types of layouts for flow analysis charts. You can also select Highlight Primary Path or Highlight Bounces as required. In the following

example, **Highlight Primary Path** is selected, the primary path is displayed in a different color in the chart, as shown in [Figure 4-166: The flow analysis chart](#).

Figure 4-166: The flow analysis chart



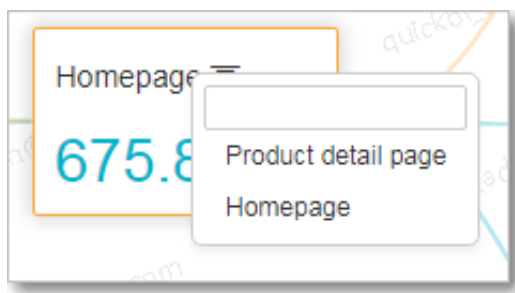
- Click the View icon to view the flow analysis of the page. If the View icon is not displayed, it indicates that no flow analysis can be performed on this page, as shown in the following figure.

Figure 4-167: Flow analysis



- You can hover over the Switch icon to switch to another page for flow analysis, as shown in the following figure.

Figure 4-168: Switch pages



8. Click Save in the upper-right corner and specify a name for the dashboard in the Save Dashboard dialog box that appears.
9. Click OK to save the dashboard.

To delete the chart, click the More icon in the upper-right corner of the chart and select Delete.

#### 4.4.5 Full Screen mode

This topic describes features of the Full Screen mode.

The Full Screen mode allows you to perform the following operations in the display area of a dashboard.

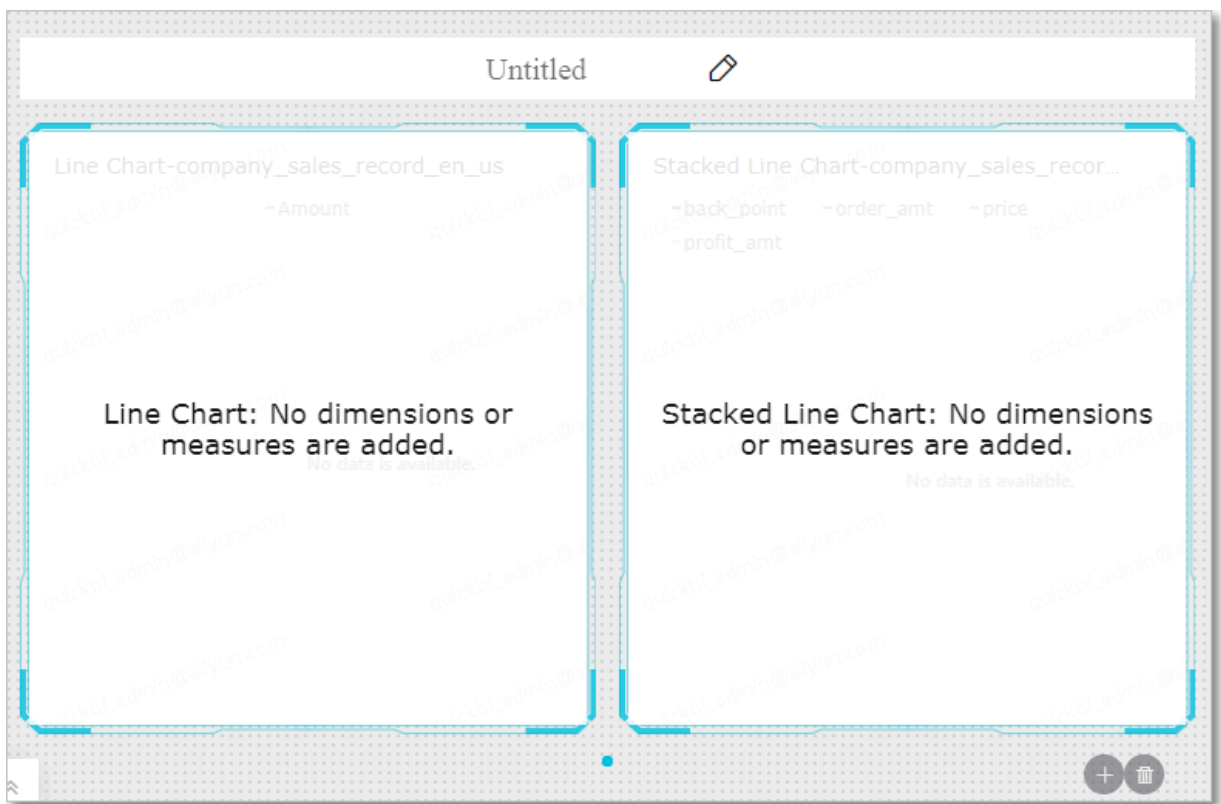
- Adjust chart positions
- Add a subscreen
- View chart data
- Delete a chart

- Change the chart type
- Configure page settings

Adjust chart positions

If you create a dashboard by using the Full Screen mode and only one chart is created, the chart covers the entire display area. If you have created multiple charts, you can click the Move icon, an arrow cross symbol, and drag the chart to the target position, as shown in [Figure 4-169: Adjust chart locations](#).

Figure 4-169: Adjust chart locations

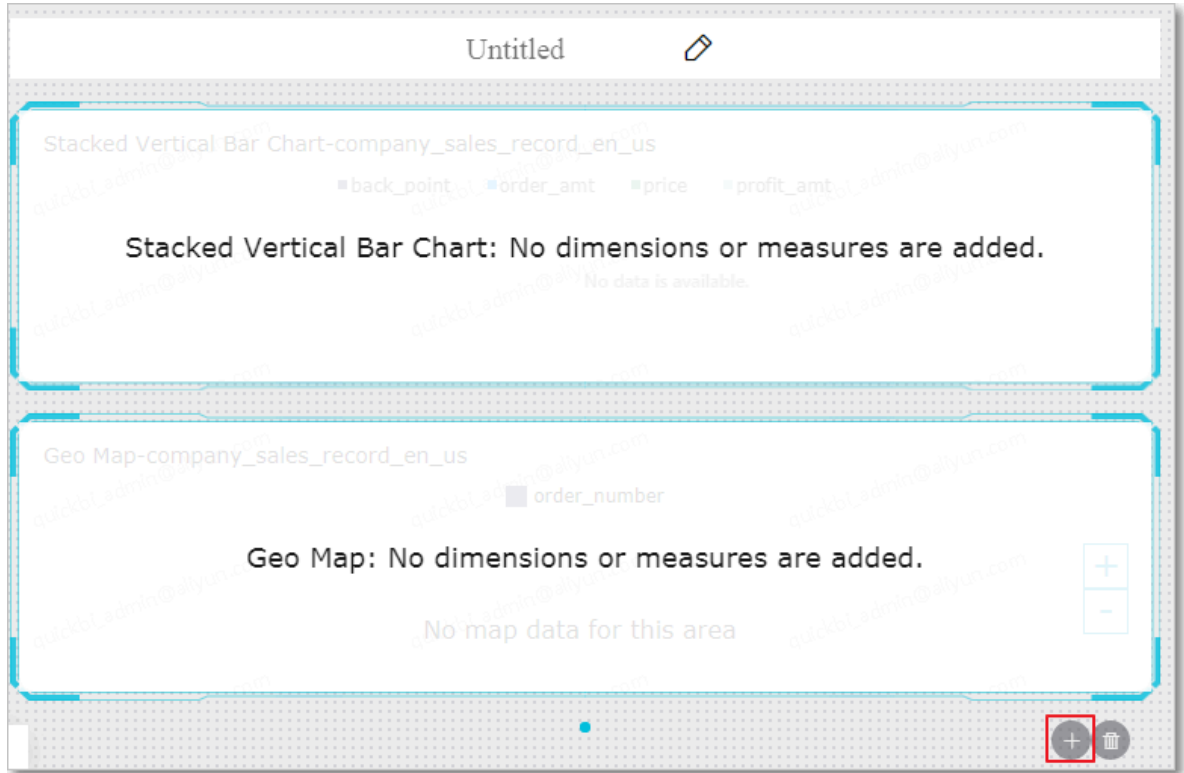


Add a subscreen

1. To add a subscreen, click the plus sign (+) in the lower-right corner, as shown in

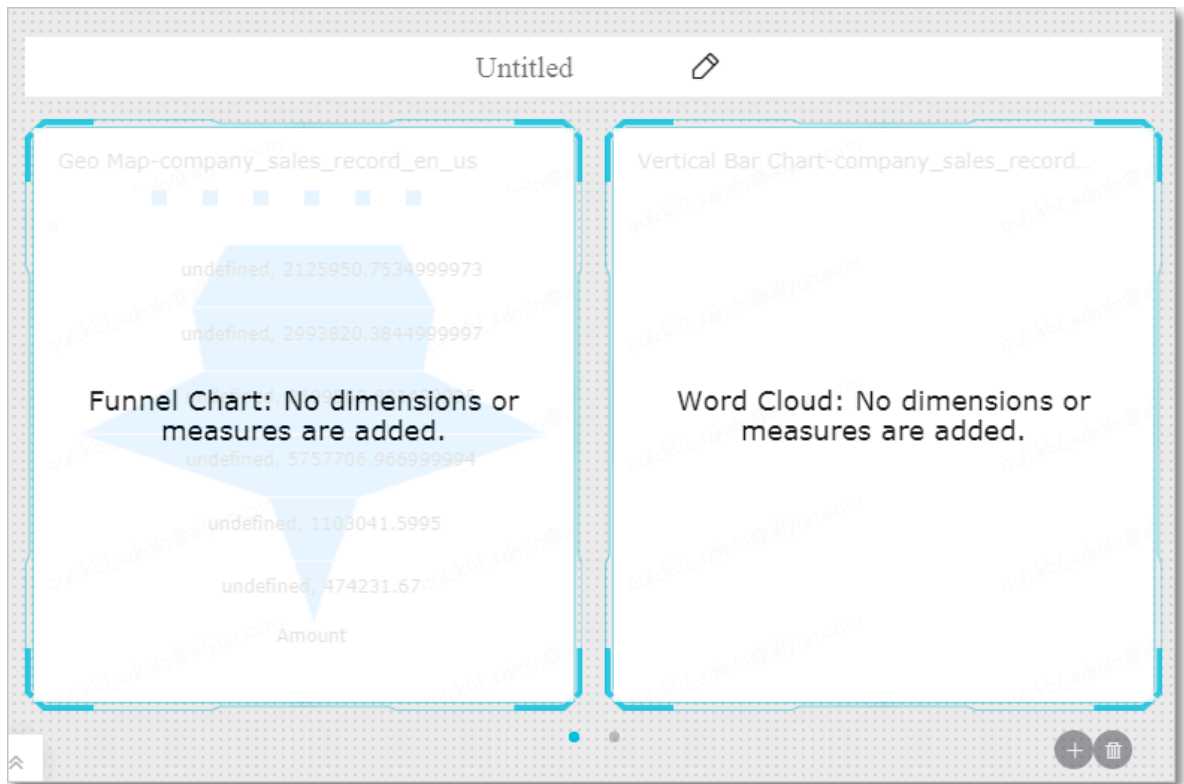
*Figure 4-170: Click the plus sign (+).*

Figure 4-170: Click the plus sign (+)



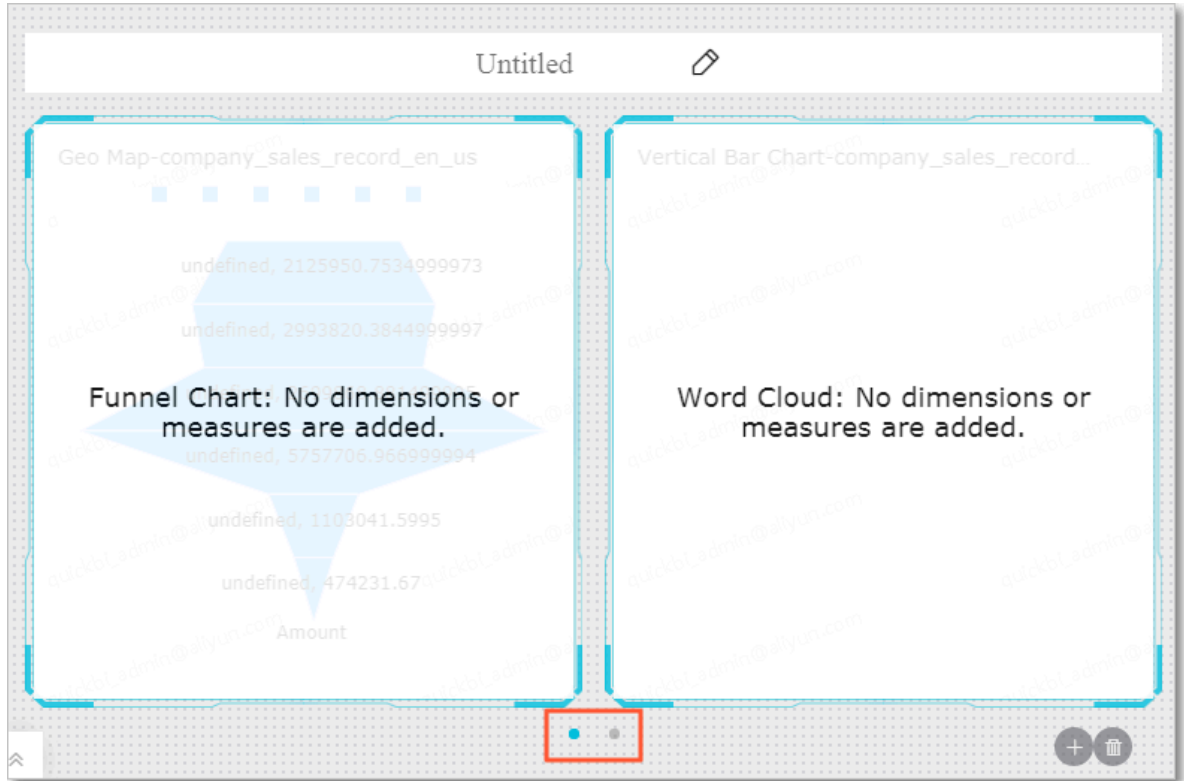
2. You can then add a chart on the subscreen, as shown in [Figure 4-171: Add a chart](#).

Figure 4-171: Add a chart



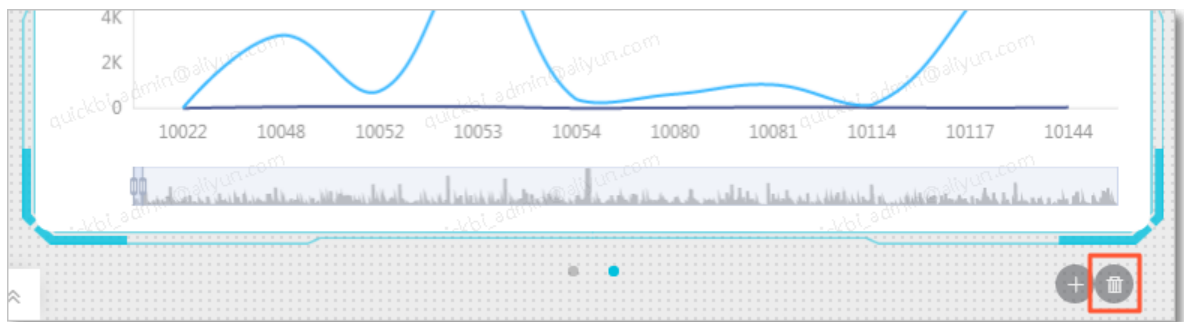
3. To switch the current subscreen to another subscreen, click the Switch Screen icon, as shown in [Figure 4-172: Switch to another subscreen](#).

Figure 4-172: Switch to another subscreen



4. To delete a subscreen, click the Delete icon in the lower-right corner, as shown in [Figure 4-173: Delete a subscreen](#).

Figure 4-173: Delete a subscreen



View data, export and view SQL statements, and delete a chart

1. In the target chart, click the More icon in the upper-right corner.
2. Select View Data to view data items in the chart.
3. Select Export to export the chart data to a local device.

4. **Select View SQL Statements to view the SQL statements.**
5. **Select Delete to delete the chart.**

Change chart types

1. **Select the target chart.**
2. **In the Graphic Design area, click Change Chart Type.**
3. **Click the target type of chart.**



**Note:**

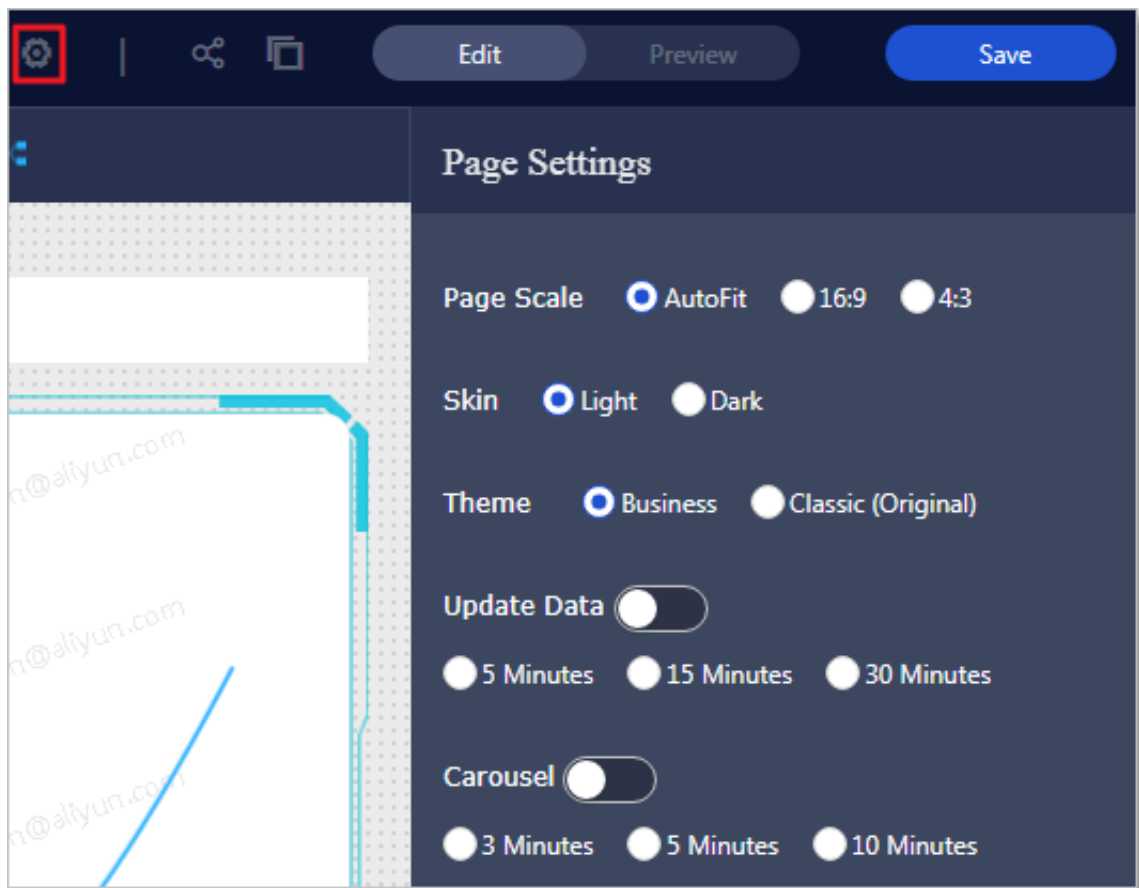
If you fail to change the chart type, it indicates that the elements of the current chart do not match those of the target chart. You need to manually adjust the elements before you change the chart type. The system provides instructions to help you adjust the elements based on the current and target chart types. You can follow the instructions to adjust the dimensions and measures to change the chart type.



## Page settings

To change the page scale, skin color, theme, and time interval of data updates or data carousels, click the Page Settings icon on the top of the page, as shown in [Figure 4-174: Page settings](#).

Figure 4-174: Page settings



### 4.4.6 Search for a dashboard

This topic describes how to search for a specific dashboard.

#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Dashboards.
3. Enter a keyword in the search box to search for the target dashboard.

### 4.4.7 Create a dashboard folder

This topic describes how to create a dashboard folder.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Dashboards.
3. Click **Create Folder** in the upper-right corner.
4. In the Create Folder dialog box that appears, specify a name for the folder and click OK.

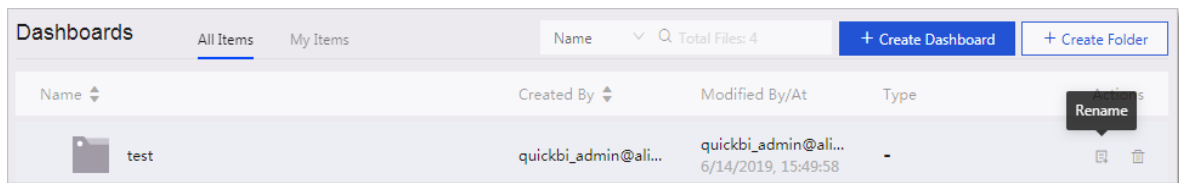
#### 4.4.8 Rename a dashboard folder

This topic describes how to rename a dashboard folder.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Dashboards.
3. On the Dashboards page, find the target dashboard folder.
4. Click the Edit Properties icon to rename the folder, as shown in [Figure 4-175: Rename a dashboard folder.](#)

Figure 4-175: Rename a dashboard folder



5. Change the folder name and click Save.

#### 4.4.9 Share a dashboard

This topic describes how to share a dashboard with other users.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Dashboards.
3. On the Dashboards page, find and right-click the target dashboard.
4. Select Share.

5. Specify the user that you want to share the dashboard with, and specify the expiration date, as shown in [Figure 4-176: Share a dashboard](#).

Figure 4-176: Share a dashboard



**Note:**

You can set the Scope parameter to All Users, User Groups, or Users based on your actual needs.

6. Click Save to share the dashboard.

#### 4.4.10 Make a dashboard public

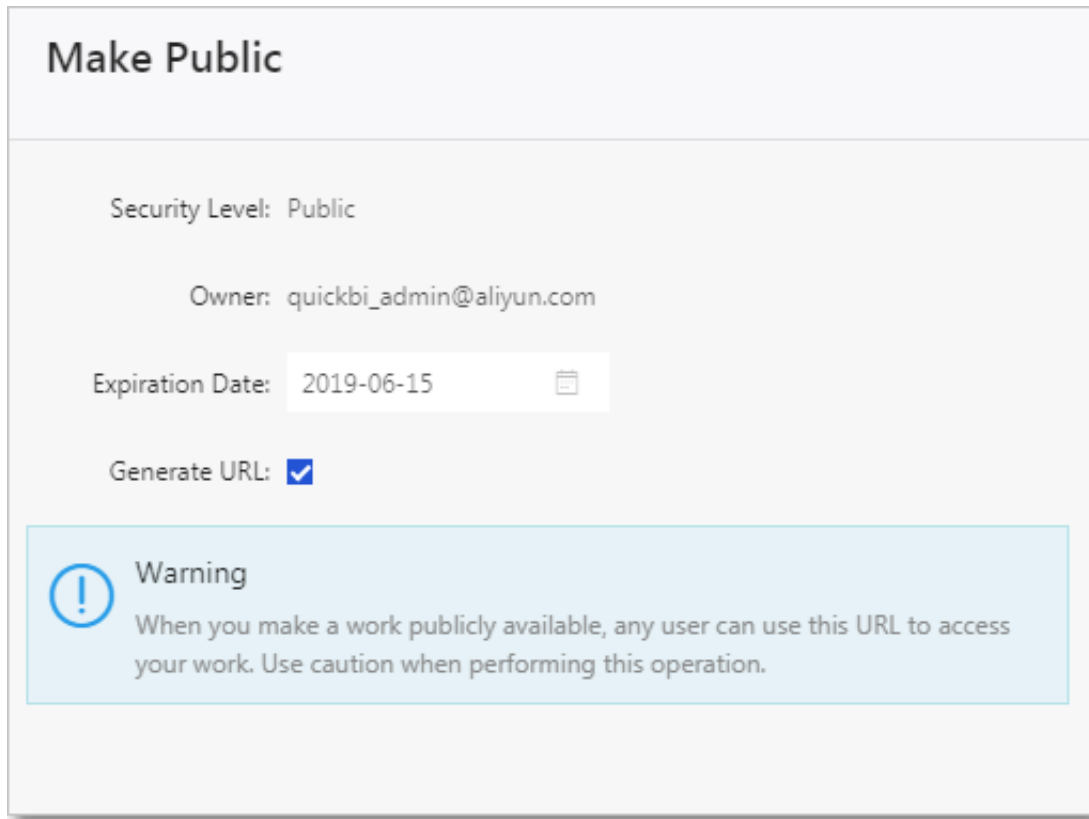
After you set a dashboard to Public, Internet users can access the dashboard through a share link.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Dashboards.
3. On the Dashboards page, find the target dashboard and click the More icon in the Actions column.
4. Select Make Public.

5. Set the expiration date and click **Make Public**, as shown in [Figure 4-177: Make a dashboard public](#).


Figure 4-177: Make a dashboard public




**Make Public**

Security Level: Public

Owner: quickbi\_admin@aliyun.com

Expiration Date: 2019-06-15 

Generate URL: ☒

 **Warning**  
When you make a work publicly available, any user can use this URL to access your work. Use caution when performing this operation.

## 4.5 Workbooks

### 4.5.1 Overview

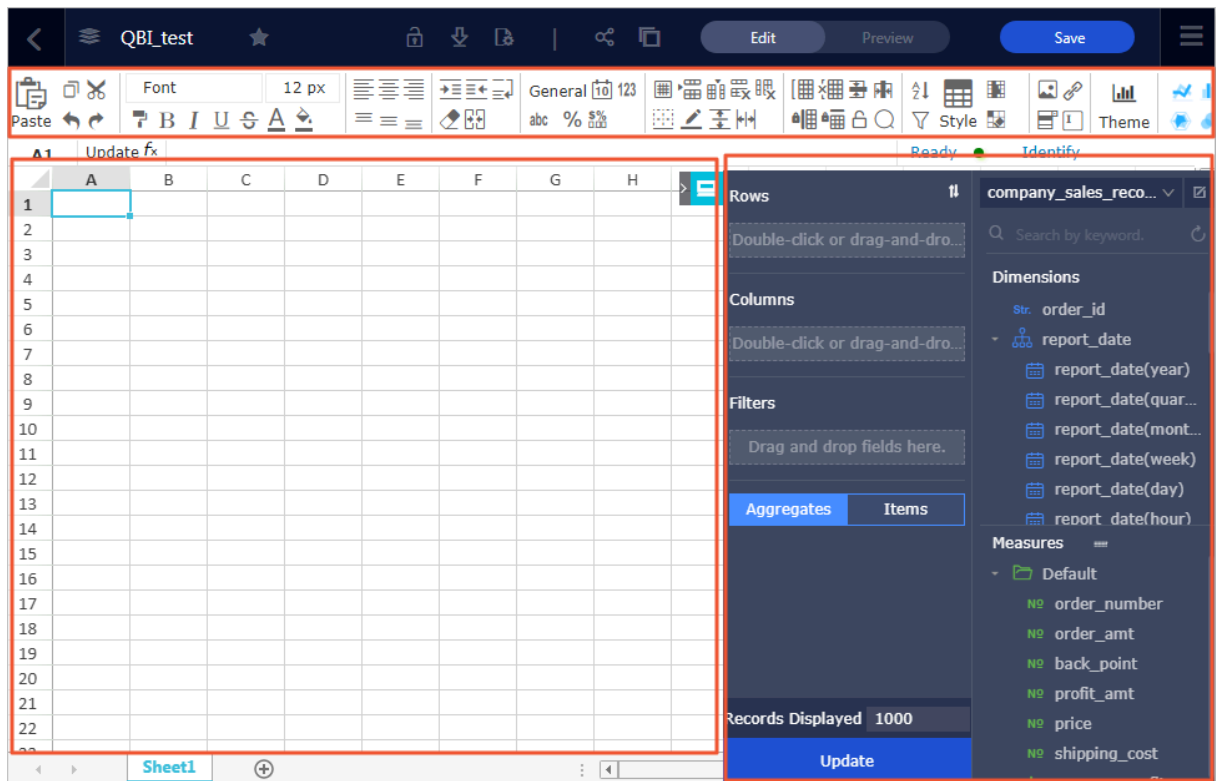
On the workbook edit page, you can filter and query data in a dataset. You can visualize data by using different types of charts.

The workbook editing page consists of three areas, as shown in [Figure 4-178: Workbook edit page](#).

- Dataset selection area
- Workbook configuration area

- **Workbook display area**

Figure 4-178: Workbook edit page



- **Dataset selection area:** In this area, you can switch the current dataset to another dataset. The fields of each dataset are respectively displayed in the Dimensions and Measures areas based on the data types preset in the system. You can select dimensions and measures based on data elements required by the chart.
- **Workbook configuration area:** In this area, you can select the target chart type, and set the color, font, and data format of cells as needed.
- **Workbook display area:** In this area, you can reprocess data based on the displayed data in cells and reference data.

## 4.5.2 Create a workbook

This topic describes how to create a workbook.

### Context

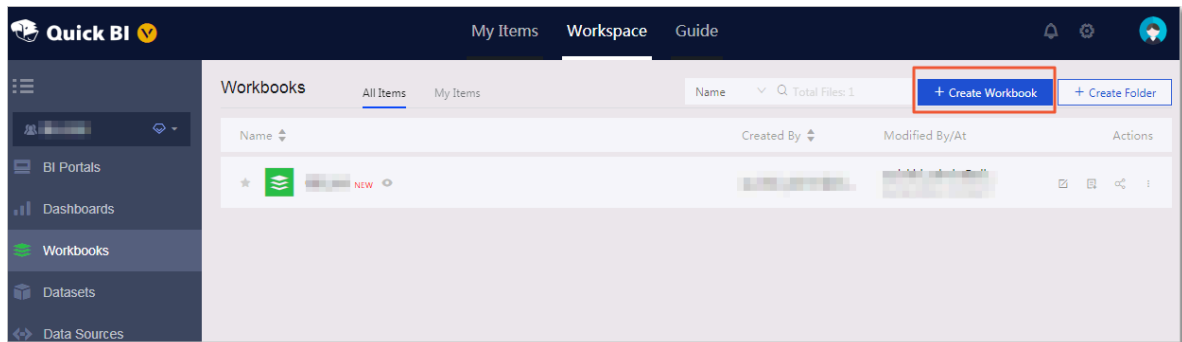
You can create workbooks in workspaces only. The personal workspace does not support workbooks.

### Procedure

1. [Log on to the Quick BI console.](#)

2. In the left-side navigation pane, click **Workbooks**.
3. Click **Create Workbook** in the upper-right corner, and you are redirected to the workbook edit page, as shown in [Figure 4-179: Create a workbook](#).

Figure 4-179: Create a workbook



4. Click **Save** and a **Save Workbook** dialog box appears. Specify a name for the workbook and set the location where you want to store the workbook, and click **OK**.

### 4.5.3 Switch datasets

This topic describes how to switch the current dataset to another dataset.

#### Context

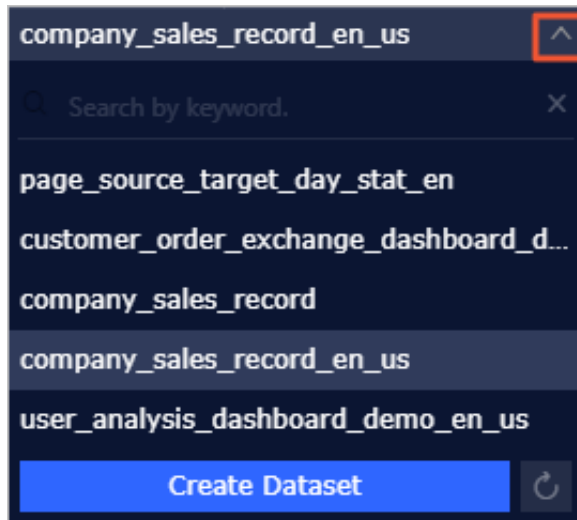
If you cannot find the target dataset, navigate to the **Datasets** page and check whether the dataset exists. For more information, see [Create datasets](#).

#### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. On the **Workbooks** page, click the target workbook to go to the edit page.

4. On the edit page, click the Drop-down List icon. In the drop-down list, select or search for the target dataset, as shown in [Figure 4-180: Switch datasets](#).

Figure 4-180: Switch datasets



#### 4.5.4 Search for a dimension or measure

This topic describes how to search for a specific dimension or measure.

##### Context

After you select a dataset, the system automatically lists the dimensions and measures in the Dimensions and Measures areas, respectively.

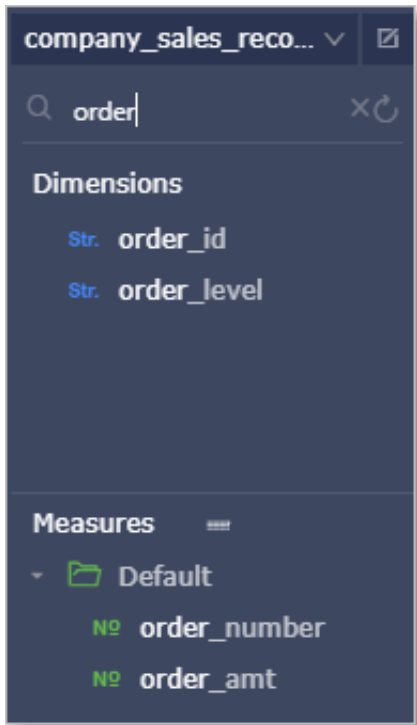
For more information about editing dimensions and measures, see [Edit a dimension](#) and [Edit a measure](#).

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Workbooks.
3. Enter a keyword of the target field in the search box.

#### 4. Click the Search icon to search for the field.

Figure 4-181: Search for a field



### 4.5.5 Fonts

You can set a font for the specified text, including the font size, font color, font style, and background color.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Workbooks**.
3. On the **Workbooks** page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

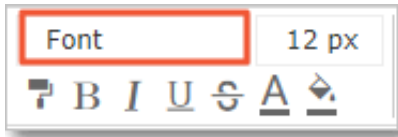


4. Click the target icon to set the font size and style.

**Set the font style**

- Click the font area.
- In the drop-down list that appears, select the target font, as shown in [Figure 4-182: Select a font](#).

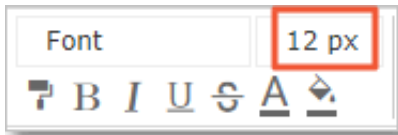
Figure 4-182: Select a font



**Set the font size**

- Click the font size area.
- In the drop-down list that appears, select the target font size, as shown in [Figure 4-183: Select a font size](#).

Figure 4-183: Select a font size



## 4.5.6 Alignment modes

You can set an alignment mode to adjust the layout of the text.

**Procedure**

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. On the **Workbooks** page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

4. Click the target alignment mode icon to adjust the layout of the text, as shown in

*Figure 4-184: Alignment modes.*

Figure 4-184: Alignment modes



## 4.5.7 Text and number formats

You can set the format to display texts and numbers in a workbook.

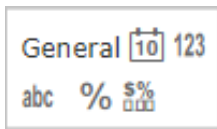
### Procedure

1. *Log on to the Quick BI console.*
2. In the left-side navigation pane, click Workbooks.
3. On the Workbooks page, click the target workbook to go to the edit page.

For more information about creating workbooks, see *Create a workbook*.

4. Click the target icon to set the format, as shown in *Figure 4-185: Display formats.*

Figure 4-185: Display formats



Parameter	Description
General	The General format directly displays numbers the way that you type them.
Date	The Date format displays data in the YYYY-MM-DD format.
Number	The Number format aligns numbers along the right side. It rounds numbers to two decimal places without thousands separators. You can double-click the cell or adjust the column width to show the complete number .
String	The String format aligns strings along the left side. You can double-click the cell or adjust the column width to show the complete string.

Parameter	Description
Percentage	The Percentage format aligns numbers along the right side. It rounds numbers to two decimal places without thousands separators. You can double-click the cell or adjust the column width to show the complete number .

### 4.5.8 Style, cell, and pane settings

You can change style, cell, and pane settings to adjust the gridlines, row heights, and border styles.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Workbooks**.
3. On the **Workbooks** page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

4. On the edit page, you can click the corresponding icons to adjust the layout of a table, as shown in [Figure 4-186: Style, cell, and pane settings](#).

Figure 4-186: Style, cell, and pane settings



Parameter	Description
Gridlines	A table shows the gridlines by default. You can click the Gridlines icon to hide gridlines.
Borders	You can click the Borders icon to add a top border , bottom border, left border, right border, outside borders, or all borders, or remove all borders.
Border Color	You can specify a color for borders.
Insert and Delete	You can insert rows and columns into a workbook, and delete rows and columns from a workbook. You can also insert and delete a workbook.
AutoFit Row Height	Double-click the AutoFit Row Height icon and the system automatically adjusts the row height.

Parameter	Description
AutoFit Column Width	Double-click the AutoFit Column Width icon and the system automatically adjusts the column width.

## 4.5.9 Insert images, hyperlinks, and drop-down boxes

This topic describes how to insert images, hyperlinks, and drop-down boxes into a workbook.

### Procedure

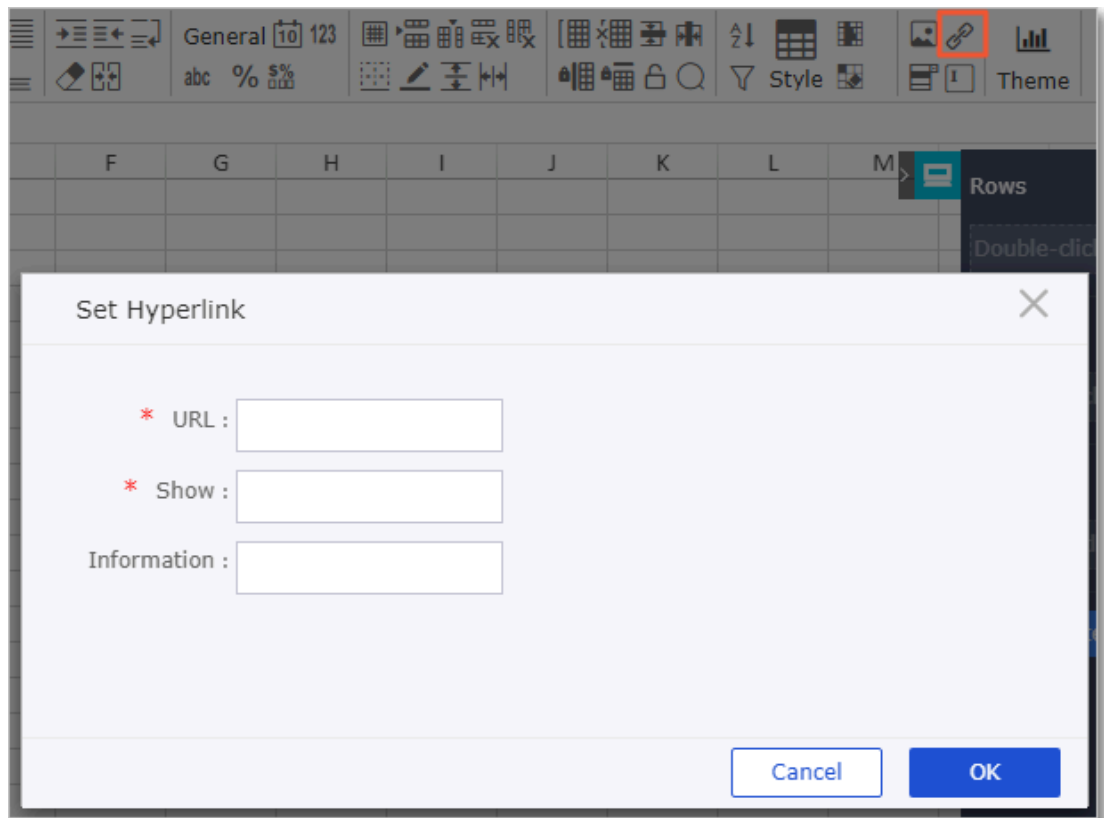
1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click **Workbooks**.

3. On the Workbooks page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

- Insert an image
  - Click the Upload Image icon.
  - In the Upload Image dialog box that appears, click Select File and select the target image.
  - Click OK to insert the image.
- Insert a hyperlink
  - Click the Hyperlink icon.
  - In the Set Hyperlink dialog box that appears, enter the target hyperlink and specify the jump text to represent the hyperlink, as shown in [Figure 4-187](#): [Insert a hyperlink](#).

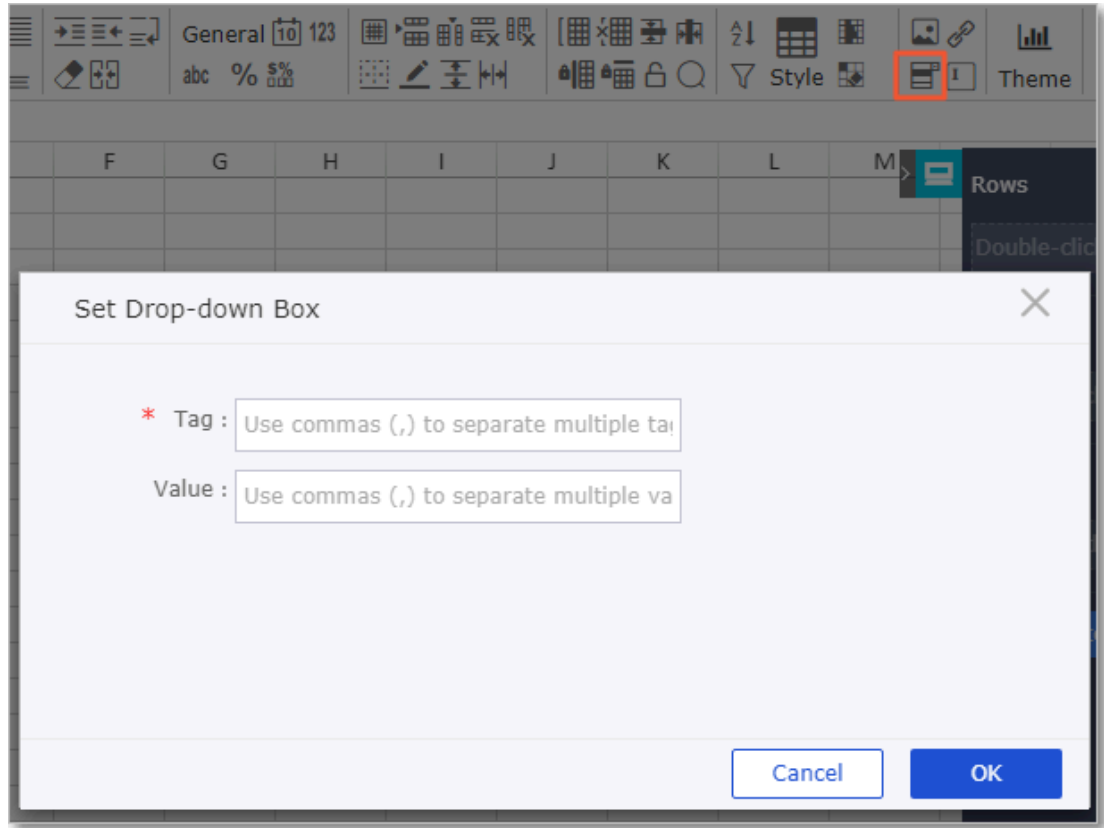
Figure 4-187: Insert a hyperlink



- Click OK to insert the hyperlink.
- Insert a drop-down box
  - Click the Drop Down icon.

- In the Set Drop-Down Box dialog box that appears, specify tags and values, as shown in [Figure 4-188: Set a drop-down box](#).

Figure 4-188: Set a drop-down box



- Click OK to insert the drop-down box.

#### 4.5.10 Set a table style

This topic describes how to set a style for a workbook.

##### Procedure

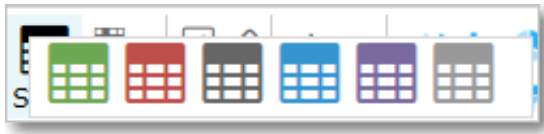
1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. On the Workbooks page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

4. Click the **Style** icon.

5. In the Style list that appears, select a style, as shown in [Figure 4-189: Styles](#).

Figure 4-189: Styles



#### 4.5.11 Set conditional formatting

This topic describes how to set conditional formatting, for example, highlight specific numbers or add an up arrow or down arrow to indicate an ascending or descending order.

##### Procedure

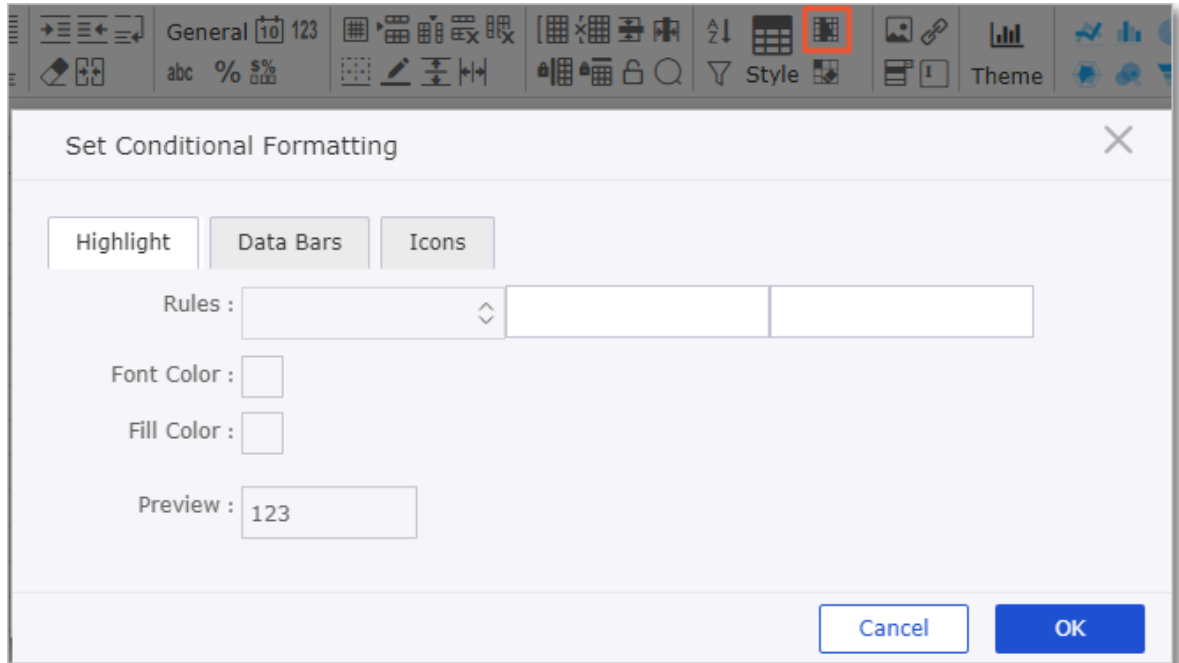
1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Workbooks.
3. On the Workbooks page, click the target workbook to go to the edit page.

For more information about creating workbooks, see [Create a workbook](#).

4. Click the Set Conditional Formatting icon to set conditions.

5. In the Set Conditional Formatting dialog box that appears, click the **Highlight** tab, as shown in *Figure 4-190: Highlight*.

Figure 4-190: Highlight

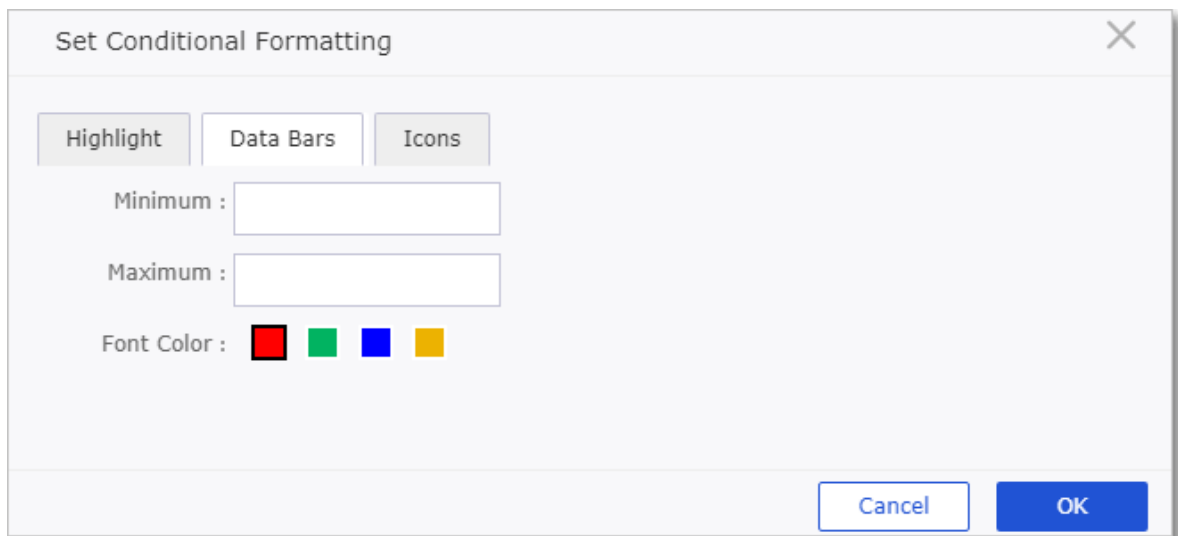


Parameter	Description
Rules	Click the drop-down icon to select a highlighting rule from the drop-down list, and specify the value or value range in the input boxes.
Font Color	Click the Color icon and select a color.
Fill Color	Click the Color icon and select a color.
Preview	Displays the highlight effect after you set the parameters.



6. Click the Data Bars tab, as shown in [Figure 4-191: Data Bars](#).

Figure 4-191: Data Bars

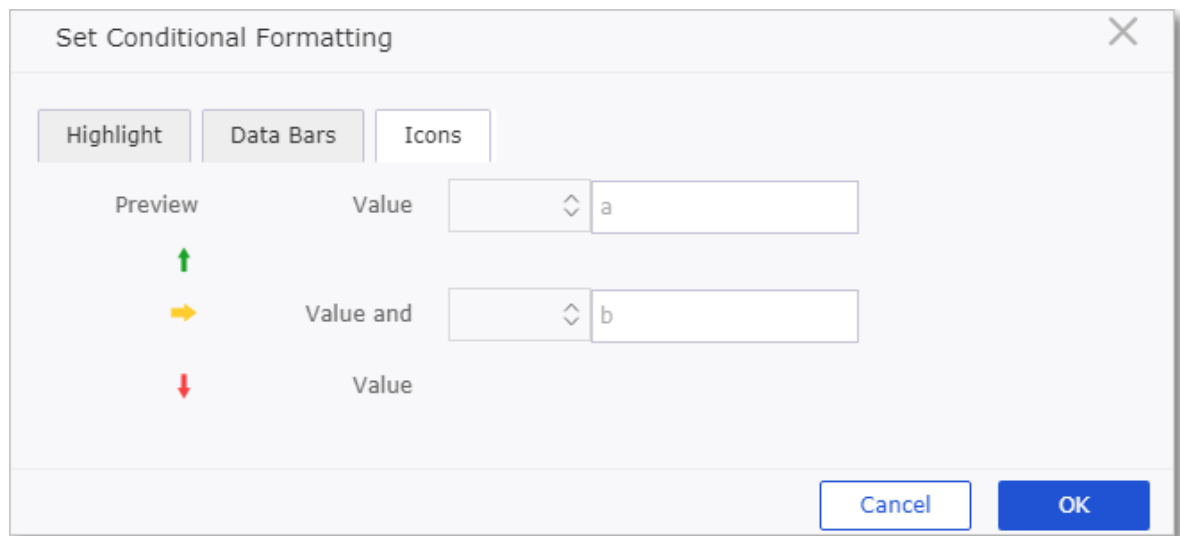


The image shows a 'Set Conditional Formatting' dialog box with the 'Data Bars' tab selected. The dialog has three tabs: 'Highlight', 'Data Bars', and 'Icons'. Under the 'Data Bars' tab, there are two input fields labeled 'Minimum' and 'Maximum'. Below these fields is a 'Font Color' section with four color swatches: red, green, blue, and yellow. At the bottom right of the dialog are 'Cancel' and 'OK' buttons.

Parameter	Description
Minimum	Enter a value in the input box.
Maximum	Enter a value in the input box.
Font Color	Click the Color icon and select a color.

7. Click the Icons tab, as shown in [Figure 4-192: Icons](#).

Figure 4-192: Icons



Click the drop-down icon, select a mathematical notation form the drop-down list, and enter a value in the input box. Next to the values that fit the specified value range, a green, yellow, or red arrow appears, accordingly.

8. After you set the parameters, click OK.

## 4.5.12 Search for a workbook

This topic describes how to search for a specific workbook.

### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. Enter a keyword in the search box.
4. Click the Search icon to search for the workbook.

## 4.5.13 Create a workbook folder

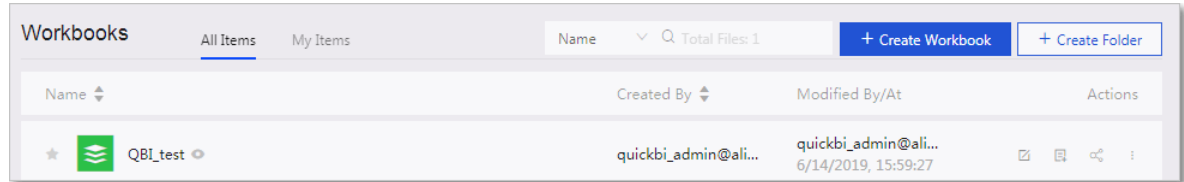
This topic describes how to create a workbook folder. Workbook holders help you manage workbooks.

### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.

3. Click **Create Folder** in the upper-right corner, as shown in [Figure 4-193: Create a holder](#).

Figure 4-193: Create a holder



4. In the **Create Folder** dialog box that appears, specify a name for the folder and click **OK**.

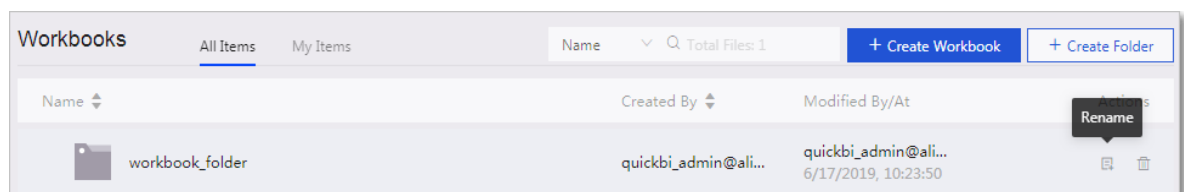
#### 4.5.14 Rename a workbook folder

This topic describes how to rename a workbook folder.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. Find the target workbook folder and click the **Edit Properties**, as shown in [Figure 4-194: Edit Properties](#).

Figure 4-194: Edit Properties



4. On the **Edit Properties** page, change the folder name and click **Save**.

#### 4.5.15 Share a workbook

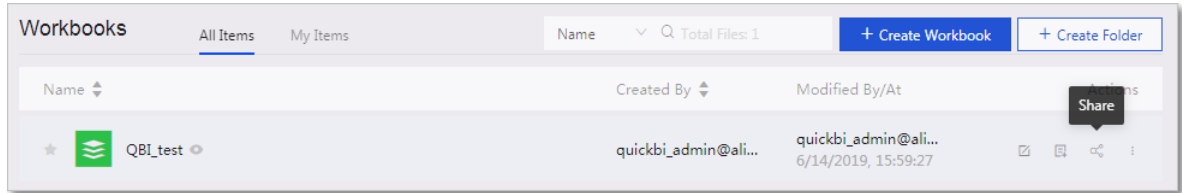
This topic describes how to share a workbook with other users.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click **Workbooks**.
3. On the **Workbooks** page, find the target workbook.

4. Click the Share icon in the Actions column, as shown in [Figure 4-195: Share a workbook](#).

Figure 4-195: Share a workbook



5. On the Share page that appears, specify the users that you want to share the workbook with.
6. Specify the expiration date.
7. Click Save to share the dashboard.

#### 4.5.16 Make a workbook public

After you create a workbook, you can make it public to allow other users to access the workbook.

##### Procedure

1. [Log on to the Quick BI console](#).
2. In the left-side navigation pane, click Workbooks.
3. On the Workbooks page, find the target workbook and click the More icon in the Actions column.
4. Select Make Public.
5. On the Make Public page that appears, specify the expiration date.
6. Select the Generate URL and click Make Public.

## 4.6 BI portals

### 4.6.1 Overview

A BI portal is a collection of dashboards, workbooks, and external links organized with menus. You can create a BI portal to perform complex thematic analysis with navigation panes.

### 4.6.2 Create a BI portal

This topic describes how to create a BI portal.

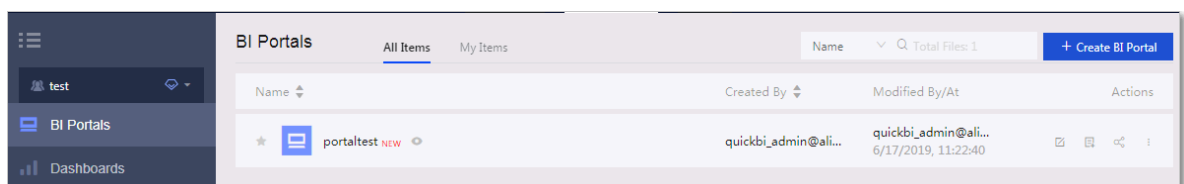
#### Context

You can create BI portals in workspaces only. The personal workspace does not support BI portals.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click BI Portals.
3. On the BI Portals page, click Create BI Portal in the upper-right corner, as shown in [Figure 4-196: Create a BI portal.](#)

Figure 4-196: Create a BI portal



4. On the Page Settings page, set the parameters and click Save in the upper-right corner.

### 4.6.3 Page settings

This topic describes how to edit a BI portal page, including the title, layout, logo, and footer.

#### Procedure

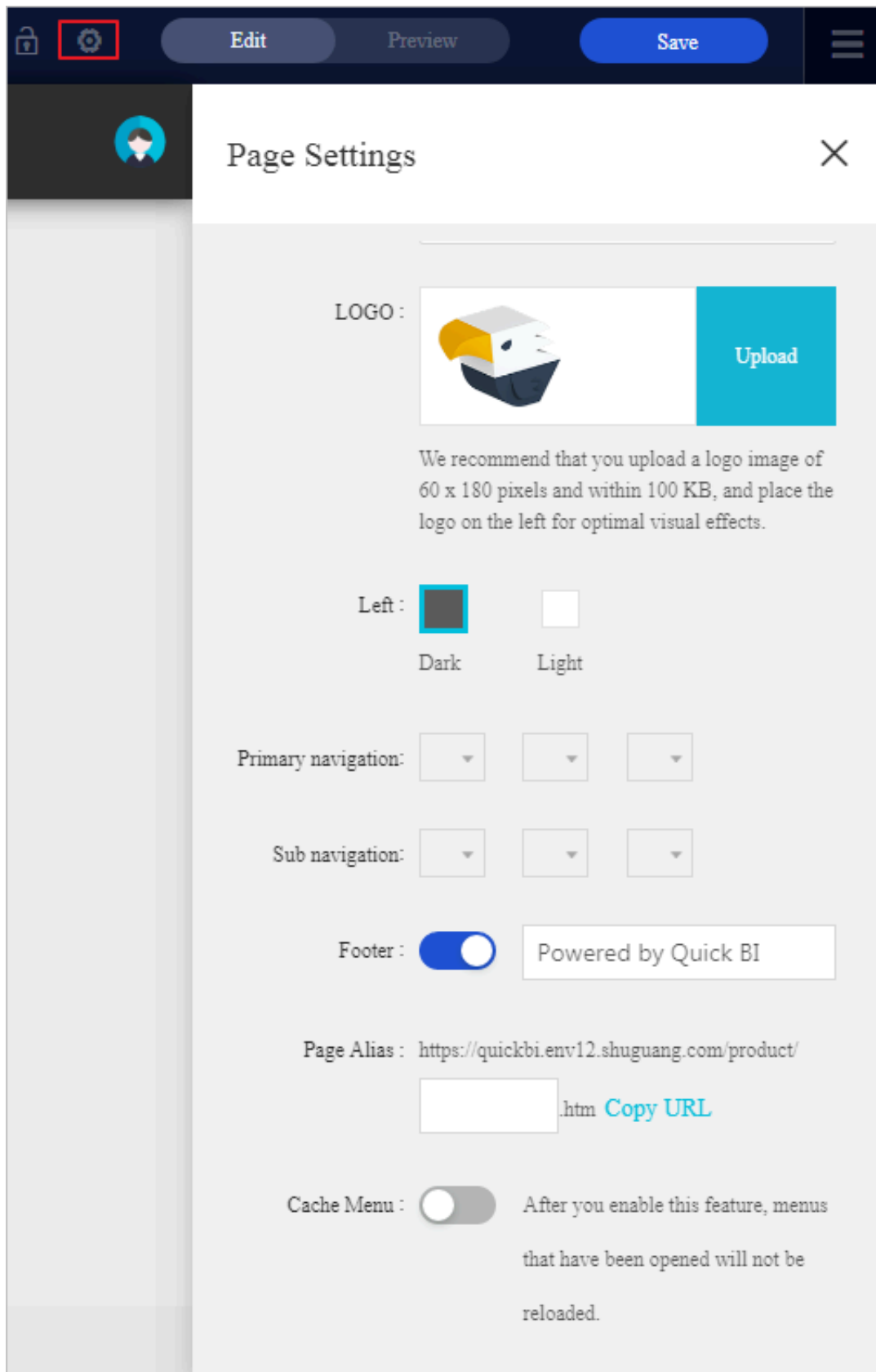
1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click BI Portals.

**3. On the BI Portals page, click the target BI portal.**

**For more information about creating BI portals, see [Create a BI portal](#).**

4. Click the Settings icon to edit the BI portal page, as shown in *Figure 4-197: Example*.

Figure 4-197: Example



5. Click Save.

## 4.6.4 Menu settings

This topic describes how to edit menu content and menu settings, including menu titles and URLs.

### Procedure

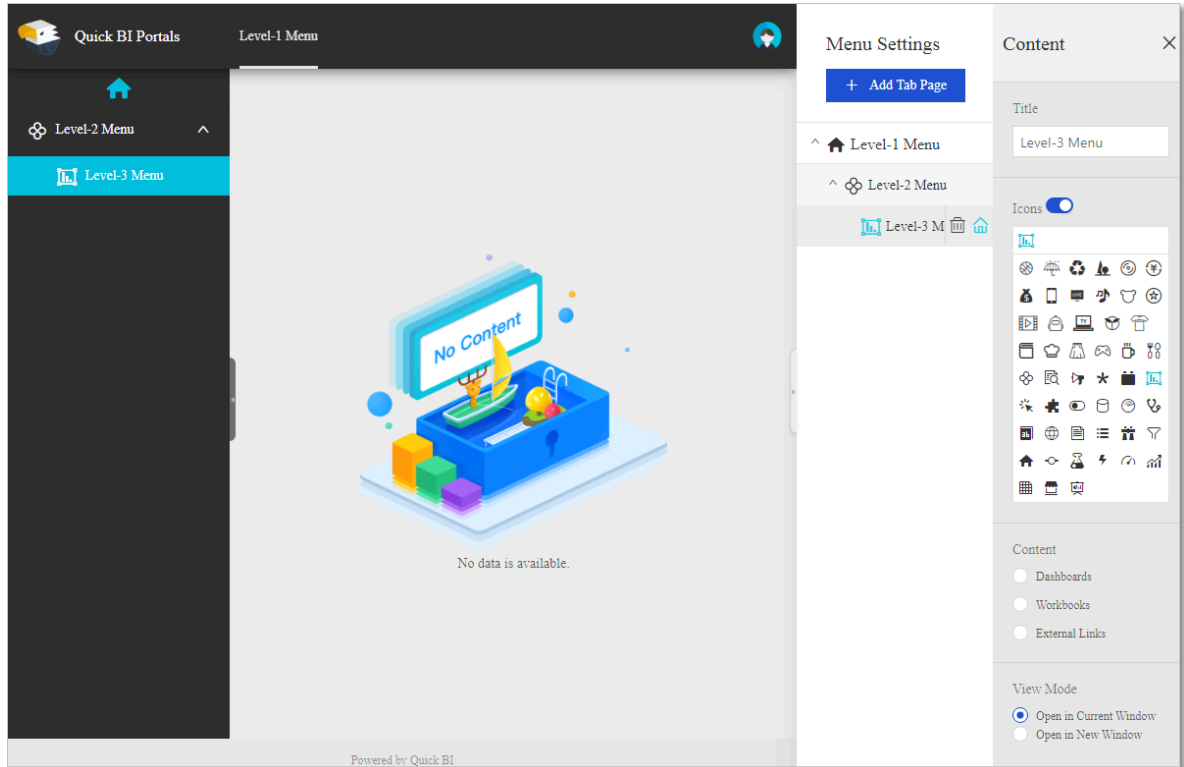
1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click BI Portals.
3. On the BI Portals page, click the target BI portal.

For more information about creating BI portals, see [Create a BI portal.](#)



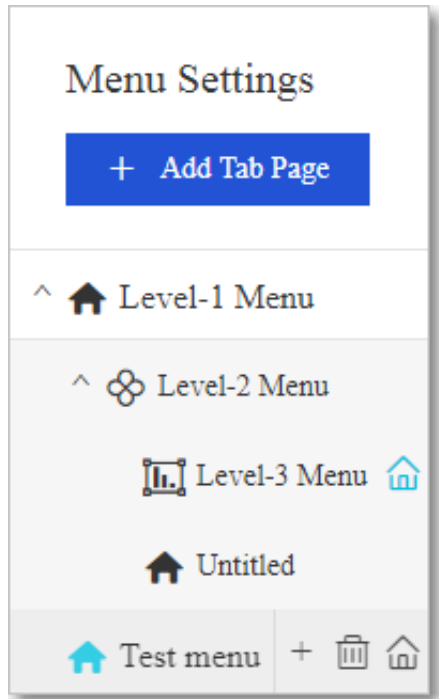
4. In the left-side navigation pane, click the target menu, and edit the menu on the right-side Content tab page, as shown in *Figure 4-198: Edit the menu content*.

Figure 4-198: Edit the menu content



- On the Menu Settings tab page, you can edit menu settings, as shown in the following figure.

Figure 4-199: Edit the menu structure



- You can add a dashboard or workbook as a menu.

5. Click Save.

## 4.7 Organization

### 4.7.1 Overview

An organization typically refers to a small or medium enterprise, public institution, college department, or a department of a large enterprise.

If your organization has a large number of members, requires multiple members to collaborate on data analysis, and has high requirements on data security, Quick BI provides the following features to meet your requirements:

- Different departments have access to different reports.
- Members with different roles have access to different data.

Members in an organization are classified into two types: administrators and common members.

## 4.7.2 Create an organization

This topic describes how to create an organization.

### Context

Before you create an organization, you must create an Apsara Stack tenant account in the Apsara Stack console. Each Apsara Stack tenant account can create or join one organization only. Make sure that you have not created or joined an organization before.

### Procedure

1. [Log on to the Quick BI console.](#)
2. Click the Settings icon, as shown in [Figure 4-200: The Settings icon.](#)

Figure 4-200: The Settings icon



3. In the left-side navigation pane, click Organization.
4. Select the Agree check box and click Create Organization.
5. In the Create Organization dialog box, specify a name for the organization.

## 4.7.3 Modify organization information

Quick BI allows you to modify the information about an organization.

### Context

Administrators of an organization can modify the information about the organization, as shown in *Figure 4-201: Modify organization information*.

Figure 4-201: Modify organization information

The screenshot shows the 'Organization' settings interface. At the top, there are tabs: 'Basics' (selected), 'Members', 'User Groups', and 'AccessKey'. Below the tabs is the title 'Organization Information'. The main content area contains four fields: 'Name' (with a placeholder and a validation message: 'The name must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (( )), and square brackets ([ ]).'), 'Description' (containing 'QuickBI testubg'), 'Created At' (containing '3/14/2019, 10:26:47'), and 'Owner' (with a placeholder). At the bottom, there are two buttons: 'Save' and 'Leave Organization'.

Administrators of an organization are responsible for adding members to the organization to collaborate on tasks.

Administrators of workspaces are responsible for adding members to their workspaces based on the roles and responsibilities of the members. Workspaces represent actual departments of an organization. Administrators of the organization can create workspaces based on actual business requirements. For example, if the organization has a sales department and an HR department, the administrators can create a sales workspace and an HR workspace accordingly. The administrators can then add employees in the sales department to the sales workspace, and employees in the HR department to the HR workspace.

Only administrators of an organization have the permission to manage members in the organization. By default, the creator of an organization is one of the administrator of the organization.

Members in an organization are classified into two types: administrators and common members.

## Procedure

1. [Log on to the Quick BI console.](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Basics tab.
4. Change the organization information and click Save.

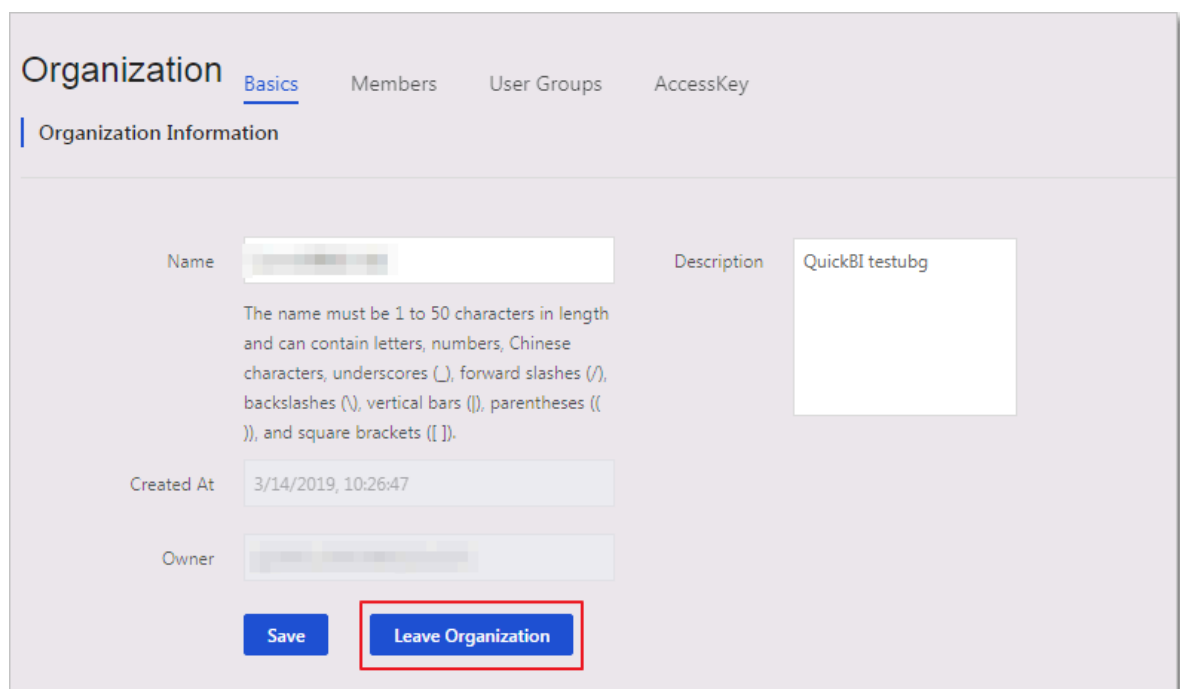
## 4.7.4 Leave an organization

Quick BI allows you to leave an organization.

### Procedure

1. [Log on to the Quick BI console.](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Basics tab and then click Leave Organization, as shown in [Figure 4-202: Leave the organization.](#)

Figure 4-202: Leave the organization



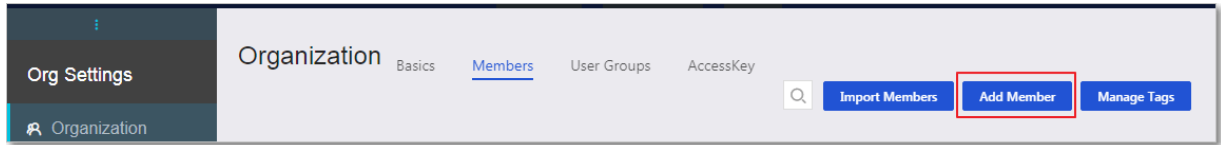
## 4.7.5 Add a member

This topic describes how to add a member to an organization.

### Context

**Quick BI allows you to add one member at a time or multiple members at the same time to an organization.**

Figure 4-203: Add a member to an organization



**When you add one member at a time, you can add an Apsara Stack tenant account or RAM user account.**

Figure 4-204: Add one member at a time

**Add Member**

Tenant Account RAM User

\* Account Enter a valid Apsara Stack tenant account.  
The account name cannot contain colons (:).

\* Alias Enter a unique alias.  
The alias must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (( )), and square brackets ([ ]).

☐ Set as Admin

Cancel OK

**Add Member**

Tenant Account RAM User

\* Account Enter a valid Apsara Stack tenant account.  
The account name cannot contain colons (:).

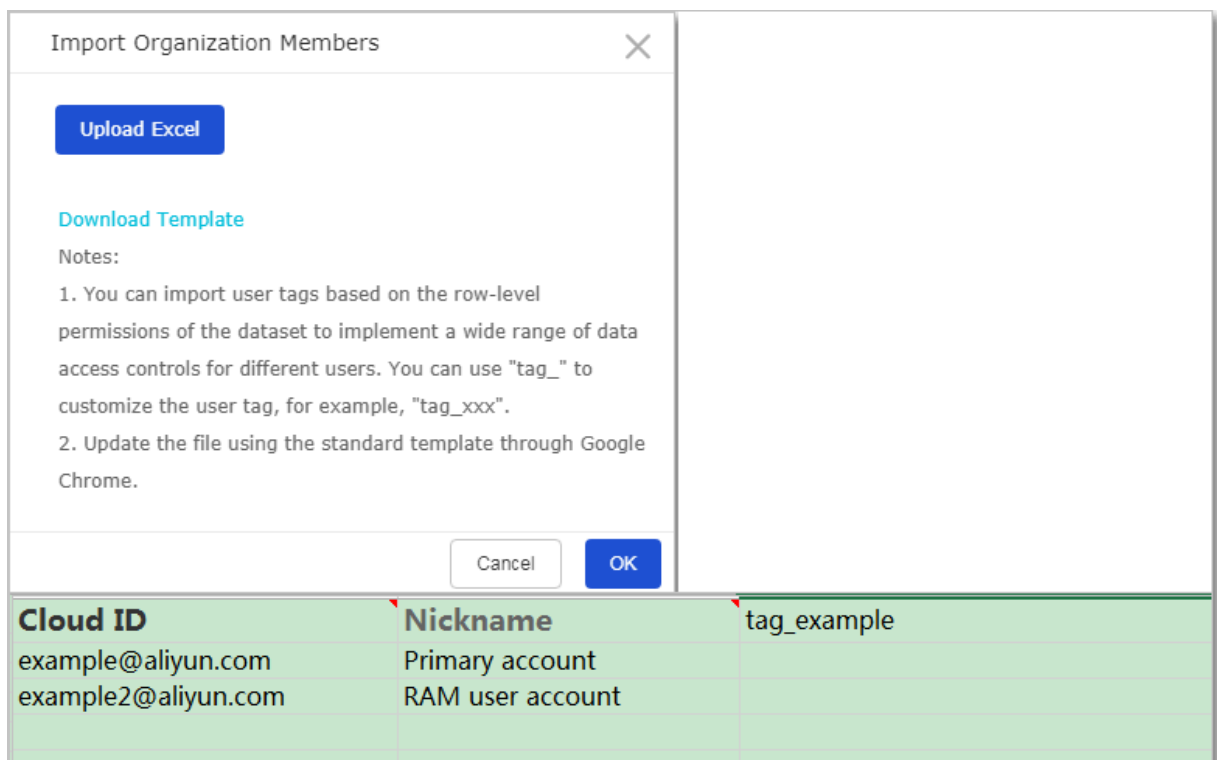
\* RAM User Enter a valid RAM user.  
The account name cannot contain colons (:).

\* Alias Enter a unique alias.  
The alias must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses (( )), and square brackets ([ ]).

☐ Set as Admin

To add multiple members at the same time, you need to download a template and enter the Apsara Stack tenant accounts and aliases of the target members into the template. Apsara Stack tenant accounts and RAM user accounts are added in different formats. When you add a RAM user, the format of the Apsara Stack tenant account is Apsara Stack tenant account: RAM user account, as shown in [Figure 4-205: Specify the target users in the template](#).

Figure 4-205: Specify the target users in the template



Import Organization Members

Upload Excel

Download Template

Notes:

1. You can import user tags based on the row-level permissions of the dataset to implement a wide range of data access controls for different users. You can use "tag\_" to customize the user tag, for example, "tag\_xxx".
2. Update the file using the standard template through Google Chrome.

Cancel OK

Cloud ID	Nickname	tag_example
example@aliyun.com	Primary account	
example2@aliyun.com	RAM user account	

## Obtain the Apsara Stack tenant account

### Context

Each department created on Apsara Stack corresponds to an Apsara Stack tenant account. To obtain the Apsara Stack tenant account, follow these steps:

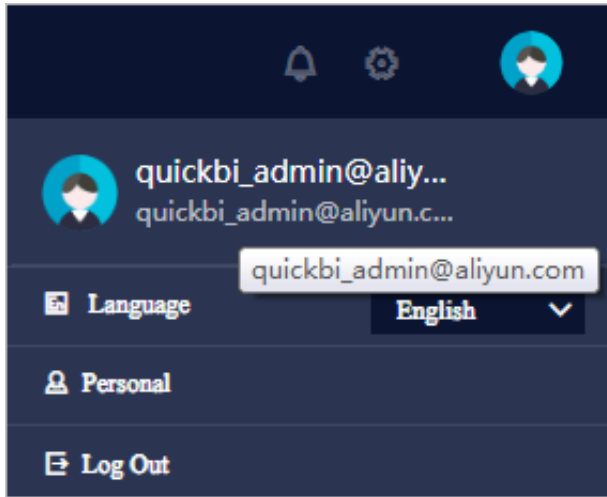
### Procedure

1. Log on to the Apsara Stack console as an administrator.
2. Choose Big Data > Quick BI, select the target department, and click Quick BI.



3. On the homepage of the Quick BI console, hover over your avatar. The Apsara Stack tenant account is displayed.

Figure 4-206: Obtain the Apsara Stack tenant account



## Obtain the Apsara Stack RAM user account

### Context

Users created on the User Management page in the Apsara Stack console are RAM user accounts. To add a RAM user account to the organization, you must specify both the Apsara Stack tenant account and RAM user account. The RAM user account is displayed on the User Management page in the Apsara Stack console. To obtain the Apsara Stack tenant account, see the preceding section.

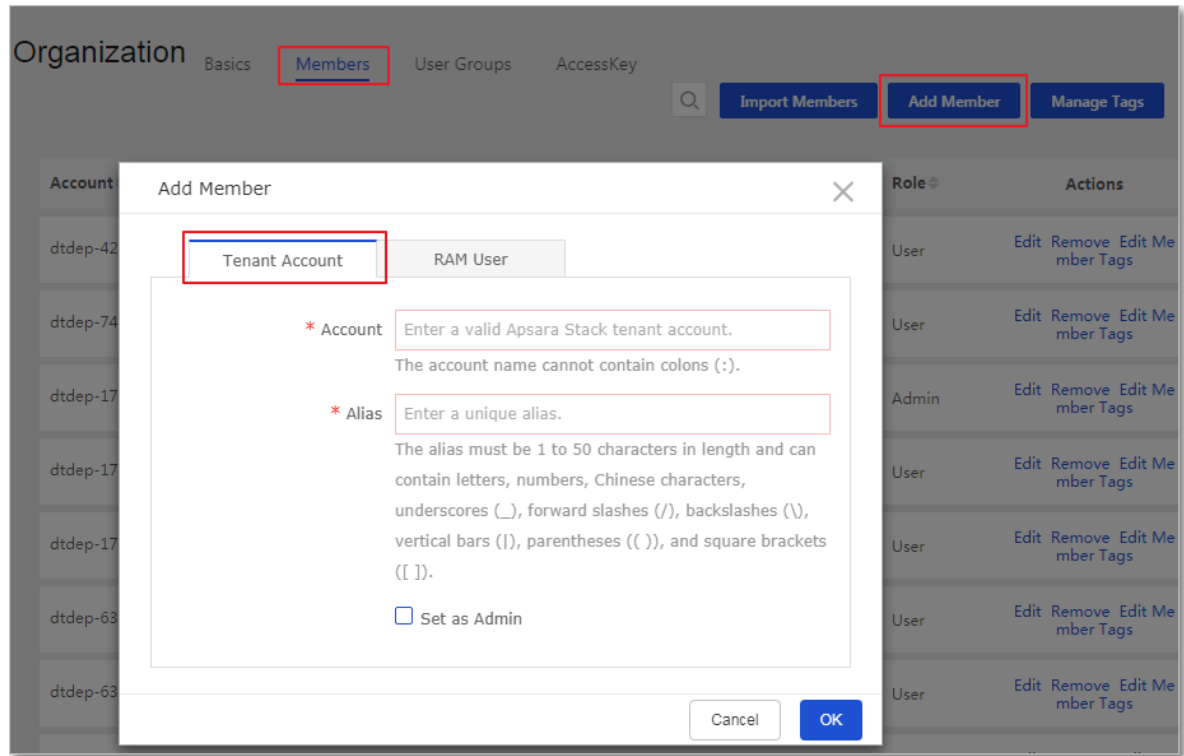
## Add an Apsara Stack tenant account

### Procedure

1. *Log on to the Quick BI console.*
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Members tab.
4. Click Add Member in the upper-right corner.
5. In the Add Member dialog box that appears, select Tenant Account.

6. Enter the Apsara Stack tenant account and alias, and select the Set as Admin check box as needed, as shown in *Figure 4-207: Add a member*.

Figure 4-207: Add a member



7. Click OK to add the member.

## Add a RAM user account

### Procedure

1. On the Organization page, click the Members tab.
2. Click Add Member in the upper-right corner.
3. In the Add Member dialog box that appears, select RAM User.

4. Enter the Apsara Stack tenant account, RAM user account, and alias, and select the Set as Admin check box as needed, as shown in [Figure 4-208: Add a member](#).

Figure 4-208: Add a member

**Add Member**

Tenant Account | **RAM User**

\* **Account** Enter a valid Apsara Stack tenant account.  
The account name cannot contain colons (:).

\* **RAM User** Enter a valid RAM user.  
The account name cannot contain colons (:).

\* **Alias** Enter a unique alias.  
The alias must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (\_), forward slashes (/), backslashes (\), vertical bars (|), parentheses ( ( ) ), and square brackets ( [ ] ).

☐ Set as Admin

Cancel OK

5. Click OK to add the member.

If the user has been already added to another organization, the system prompts an error.

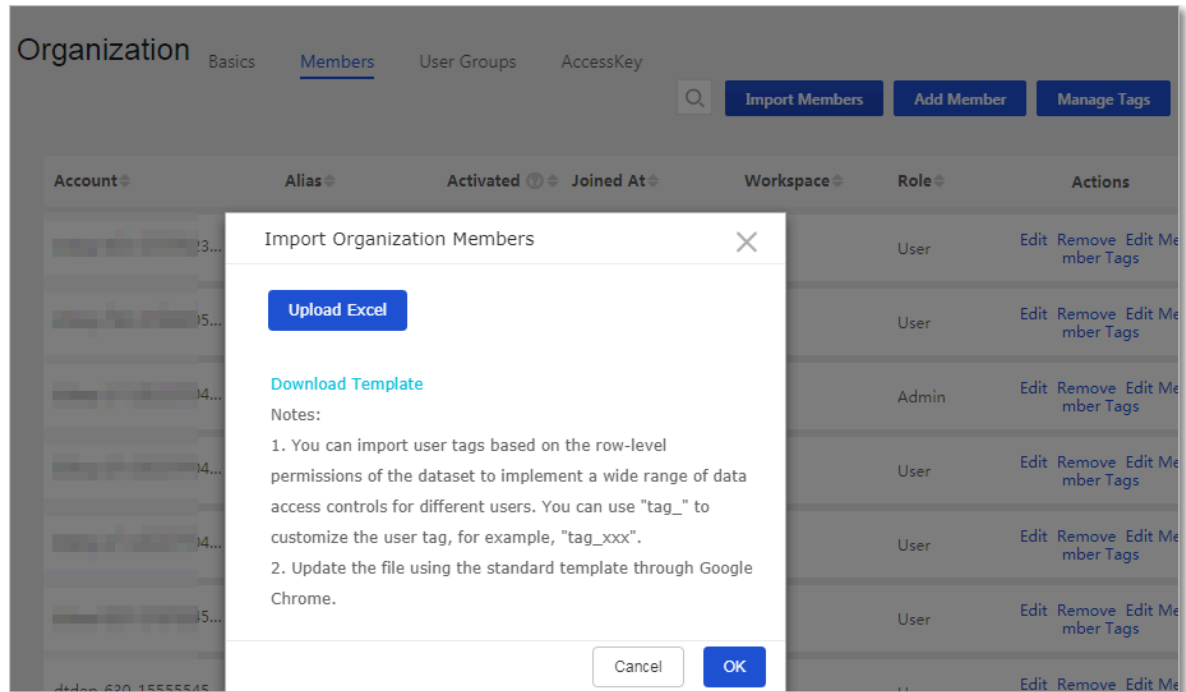
## Add multiple members at the same time

### Procedure

1. On the Organization page, click Import Members.

2. In the Import Organization Members dialog box that appears, click Upload Excel and select the target local file, as shown in [Figure 4-209: Upload an Excel file](#).

Figure 4-209: Upload an Excel file



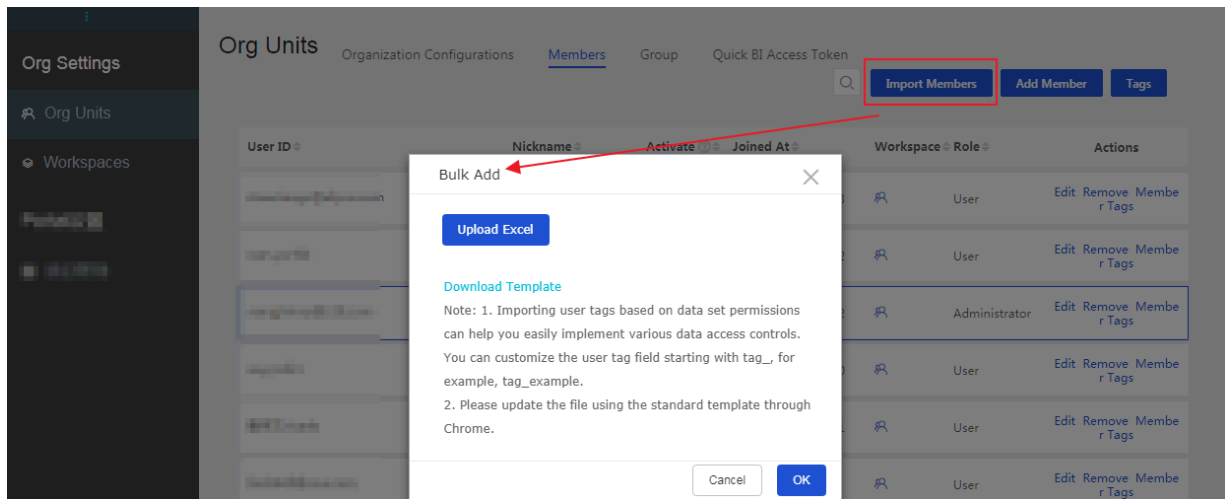
3. Click OK to add the members.

#### 4.7.6 Manage member tags

Member tags are used to set dataset row-level permissions. This topic describes how to manage member tags. For more information about how to configure row-level permissions, see [Set row-level permissions](#).

Add a member tag

With member tags, you can batch import members, as shown in the following figure.



You can click Download Template in the Import Organization Members dialog box to obtain a template. The following figure shows the member information in this example.

Account	Nickname	tag_tagArea	tag_tagProvince
example1@aliyun.com	example1	East	Anhui
example2@aliyun.com	example2	East	Anhui

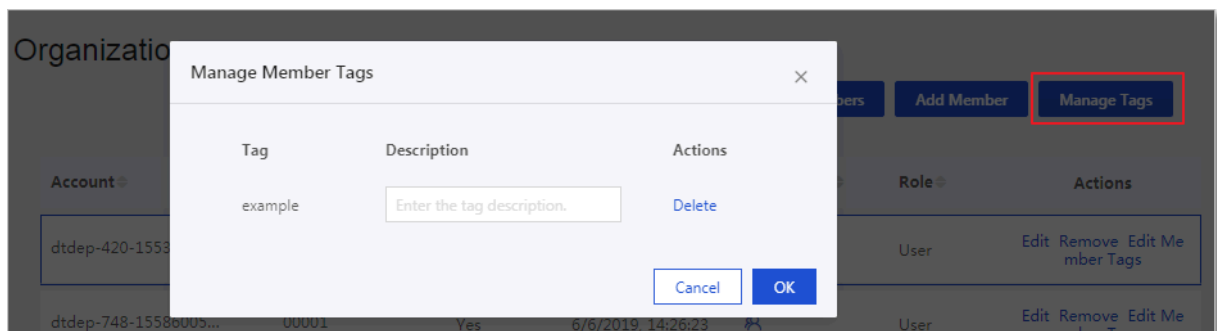


#### Note:

If you do not need to set row-level permissions for a member, set the member tag to `$ALL_MEMBERS$`.

Manage a member tag

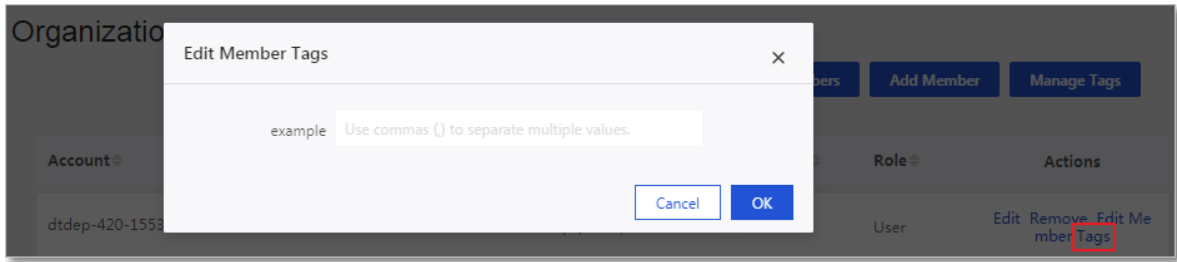
After a member tag is added, you can click Manage Tags to manage the tag, as shown in the following figure.



Edit a member tag

1. Select the target member and click Edit Member Tag in the Actions column.

2. In the Edit Member Tags dialog box that appears, specify the values and click OK, as shown in the following figure.



#### 4.7.7 Edit a member

You can set an alias and role for members in the organization. This makes it easier for you to search for members in the organization.

##### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Members tab.
4. Find the target member and click Edit in the Actions column.

5. You can change the member information in the dialog box that appears, as shown in *Figure 4-210: Edit a member*.

Figure 4-210: Edit a member

Figure 4-210 shows the "Edit User Info" dialog box. It contains the following fields and options:

- \* Account:** A text input field (blurred).
- \* Alias:** A text input field containing "test001".
- Alias Description:** The alias must be 1 to 50 characters in length and can contain letters, numbers, Chinese characters, underscores (`_`), forward slashes (`/`), backslashes (`\`), vertical bars (`|`), parentheses (`( )`), and square brackets (`[ ]`).
- Set as Admin:** A checked checkbox.
- Buttons:** "Cancel" and "OK" buttons at the bottom right.

6. Click OK to save the changes.

### 4.7.8 Remove a member

This topic describes how to remove a member from the organization.

#### Context

Only administrators of the organization have the permission to remove members from the organization. Before you remove a member, if the member is added to a workspace, you must remove the member from the workspace first. Otherwise, you cannot remove the member from the organization.



#### Note:

This operation is irreversible. After a member is removed from an organization, you need to add the member to the organization again when needed. Proceed with caution.

## Procedure

1. [Log on to the Quick BI console.](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Members tab.
4. Find the target member and click Remove in the Actions column.
5. Click OK to remove the member.

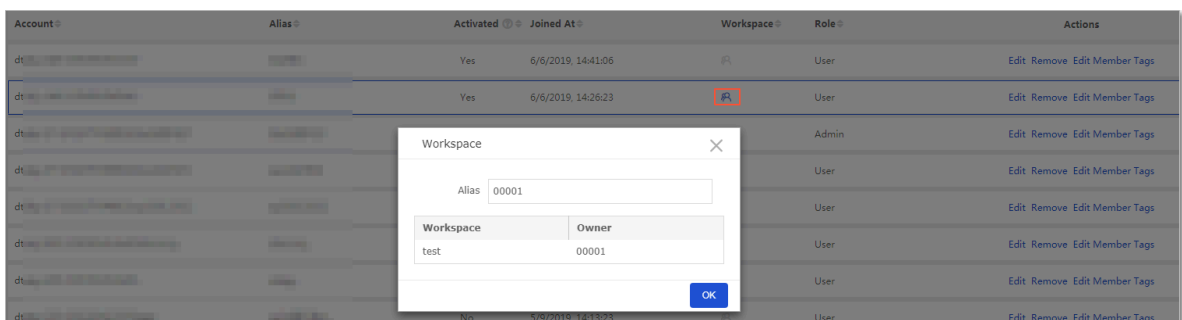
### 4.7.9 Query the workspace that a user belongs to

You can query the workspace that a user belongs to.

## Procedure

1. [Log on to the Quick BI console.](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Members tab.
4. Find the target member and click the Workspace icon in the Workspace column, as shown in [Figure 4-211: View the workspace that a user belongs to.](#)

Figure 4-211: View the workspace that a user belongs to



5. Click OK to close the dialog box.

### 4.7.10 Search for members

You can search for a specific member in the organization by its alias or Alibaba Cloud account.

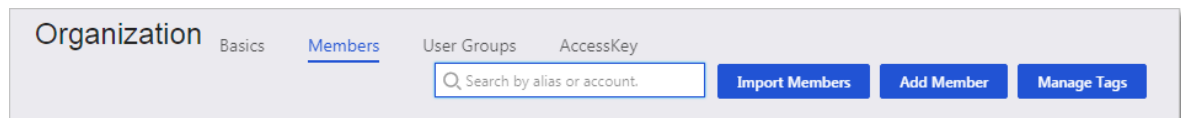
## Procedure

1. [Log on to the Quick BI console.](#)



2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Organization page.
3. On the Organization page, click the Members tab.
4. Enter the alias or Alibaba Cloud account of the target member in the search box, and click the Search icon, as shown in *Figure 4-212: Search for a member in the organization*.

Figure 4-212: Search for a member in the organization



## 4.7.11 Workspaces

### 4.7.11.1 Overview

A workspace is managed by its administrators. The role of a workspace administrator is assigned to members by the administrator that creates the workspace. A workspace administrator can specify other members in the workspace as administrators of the workspace.

Workspace management includes:

- Create a workspace
- Edit a workspace
- Edit the default workspace

### 4.7.11.2 What is a workspace?

A workspace allows members in the same organization to collaborate on tasks. In a workspace, each member plays a role in creating and modifying data sources, datasets, workbooks, and BI portals. Data objects are created and stored in their workspaces. Each workspace has its own data objects.

A workspace has the following properties:

- Name
- Description
- Permission settings: Works to Be Made Public: This check box is selected by default. You can clear this check box to deny public access to dashboards in this workspace. Works to Be Shared: This check box is selected by default. You can

clear this check box to disallow data objects to be shared with members from other workspaces.

- **Preference settings:** You can select **Use Technical Names** or **Use Field Descriptions**. Datasets to be created in this workspace will be named based on the settings. Existing datasets are not affected.

You can enter member accounts to fuzzy search members and assign roles to the members. Different roles have different access permissions to data in a workspace. Each member can be assigned with multiple roles and must have at least one role.

Roles include admin, developer, analyst, and viewer.

#### Roles and permissions

Permissions of a role are fixed and cannot be changed. When you grant permissions to a member, assign the appropriate role to the member. The following tables list the permissions of each role in different scenarios.

- **Supported operations on data objects**

Table 4-3: Access to data objects

Permission	Developer	Analyst	Viewer
Data sources and datasets	Yes	No	No
Workbooks	Yes	Yes	Yes
Dashboards	Yes	Yes	Yes
BI Portals	Yes	Yes	Yes

- **Supported operations on data**

Table 4-4: Supported operations on data

Permission	Developer	Analyst	Viewer
Create a data source	Yes	No	No
Modify a data source	Developers can only modify data sources created by themselves.	No	No

Permission	Developer	Analyst	Viewer
Delete a data source	Developers can only delete data sources created by themselves.	No	No
Use data sources	Yes	No	No
Create a dataset	Yes	No	No
Modify a dataset	Developers can only modify datasets created by themselves.	No	No
Delete a dataset	Developers can only delete data sources created by themselves.	No	No
Use datasets	Yes	Yes	No

- Supported operations on workbooks

Table 4-5: Supported operations on workbooks

Permission	Developer	Analyst	Viewer
Create a workbook	Yes	Yes	No
Modify workbooks	Developers can only modify workbooks created by themselves.	Analysts can only modify workbooks created by themselves.	No
Delete a workbook	Developers can only delete workbooks created by themselves.	Analysts can only delete workbooks created by themselves.	No
Preview a workbook	Yes	Yes	Yes
Share a workbook	Developers can only share workbooks created by themselves.	Analysts can only share workbooks created by themselves.	No

Permission	Developer	Analyst	Viewer
Reference a workbook	Yes	Yes	No

- Supported operations on dashboards

Table 4-6: Supported operations on dashboards

Permission	Developer	Analyst	Viewer
Create a dashboard	Yes	Yes	No
Modify a dashboard	Developers can only modify dashboards created by themselves.	Analysts can only modify dashboards created by themselves.	No
Delete a dashboard	Developers can only delete dashboards created by themselves.	Analysts can only delete dashboards created by themselves.	No
Preview a dashboard	Yes	Yes	Yes
Share a dashboard	Developers can only share dashboards created by themselves.	Analysts can only share dashboards created by themselves.	No
Reference a dashboard	Yes	Yes	No
Publish a dashboard	Developers can only publish dashboards created by themselves.	Analysts can only publish dashboards created by themselves.	No

- **Supported operations on BI portals**

Table 4-7: Supported operations on BI portals

Permission	Developer	Analyst	Viewer
Create a BI portal	Yes	Yes	No
Modify a BI portal	Developers can only modify BI portals created by themselves.	Analysts can only modify BI portals created by themselves.	No
Delete a BI portal	Developers can only delete BI portals created by themselves.	Analysts can only delete BI portals created by themselves.	No
View a BI portal	Yes	Yes	Yes
Share a BI portal	Developers can only share BI portals created by themselves.	Analysts can only share BI portals created by themselves.	No

#### 4.7.11.3 Differences between the personal workspace and a workspace

A user's workspace is called the personal workspace. Differences between the personal space and a workspace are as follows:

- The personal workspace is automatically created after you log on to the Quick BI console for the first time. A workspace is manually created by an administrator of the organization.
- You cannot create a new personal workspace or delete the personal workspace.
- You cannot add other members to the personal workspace. Therefore, you cannot collaborate or share data objects with other members.

#### 4.7.12 Create a workspace

This topic describes how to create a workspace.

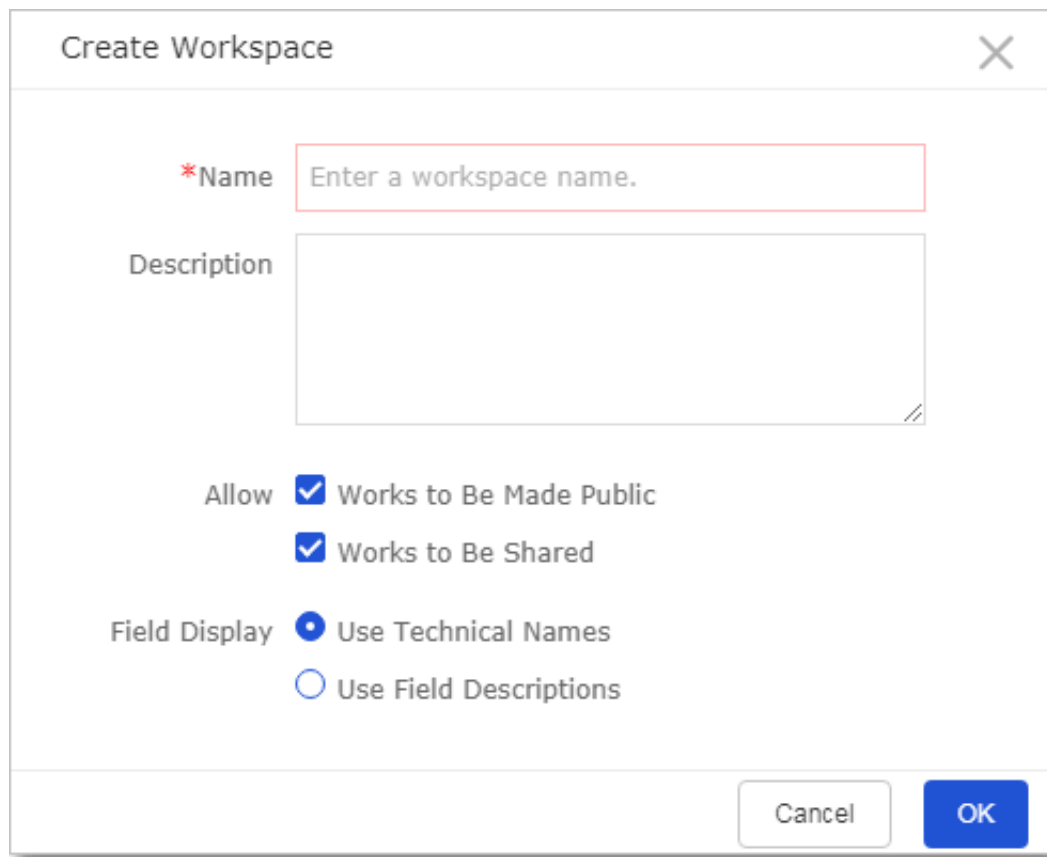
##### Procedure

1. [Log on to the Quick BI console](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon.

3. Choose Workspaces > Create Workspace.

4. In the Create Workspace that appears, enter a name for the workspace, as shown in *Figure 4-213: Create a workspace*.

Figure 4-213: Create a workspace

A screenshot of the 'Create Workspace' dialog box. The dialog has a title bar with 'Create Workspace' and a close button (X). Inside, there is a form with the following fields and options: a required name field labeled '\*Name' with a red border and placeholder text 'Enter a workspace name.'; a description field labeled 'Description' with a larger text area; an 'Allow' section with two checked checkboxes: 'Works to Be Made Public' and 'Works to Be Shared'; a 'Field Display' section with two radio buttons: 'Use Technical Names' (selected) and 'Use Field Descriptions'. At the bottom right are 'Cancel' and 'OK' buttons.

5. Click OK to create the workspace.

#### 4.7.13 Edit workspace information

The information about the personal workspace can be edited by the owner only.

The information about a workspace can be edited by an administrator of the workspace.

##### Procedure

1. *Log on to the Quick BI console.*
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon.
3. In the left-side navigation pane, click Workspaces.
4. Click the Settings tab.

**5. Click Edit Workspace and edit the information about the workspace.**

Figure 4-214: Edit the information about the workspace

The screenshot shows the 'Edit Workspace' interface. At the top, there are three tabs: 'Settings' (highlighted in blue), 'Members', and 'Embedded Reports'. The 'Settings' tab contains several input fields and checkboxes. The 'Name' field contains the text 'test'. The 'Created At' field shows a timestamp '6/6/2019, 14:25:23'. The 'Owner' field contains '00001'. To the right of these fields is a large text area labeled 'Description'. Below the input fields, there are two checkboxes, both of which are checked: 'Allow Works to Be Made Public' and 'Allow Works to Be Shared'. At the bottom of the form, there are two radio buttons under the label 'Field Display': 'Use Technical Names' (which is selected) and 'Use Field Descriptions'. A prominent blue button labeled 'Edit Workspace' is located at the bottom center of the form.

**6. Click OK to save the new information.**

### 4.7.14 Leave a workspace

This topic describes how to leave a workspace.

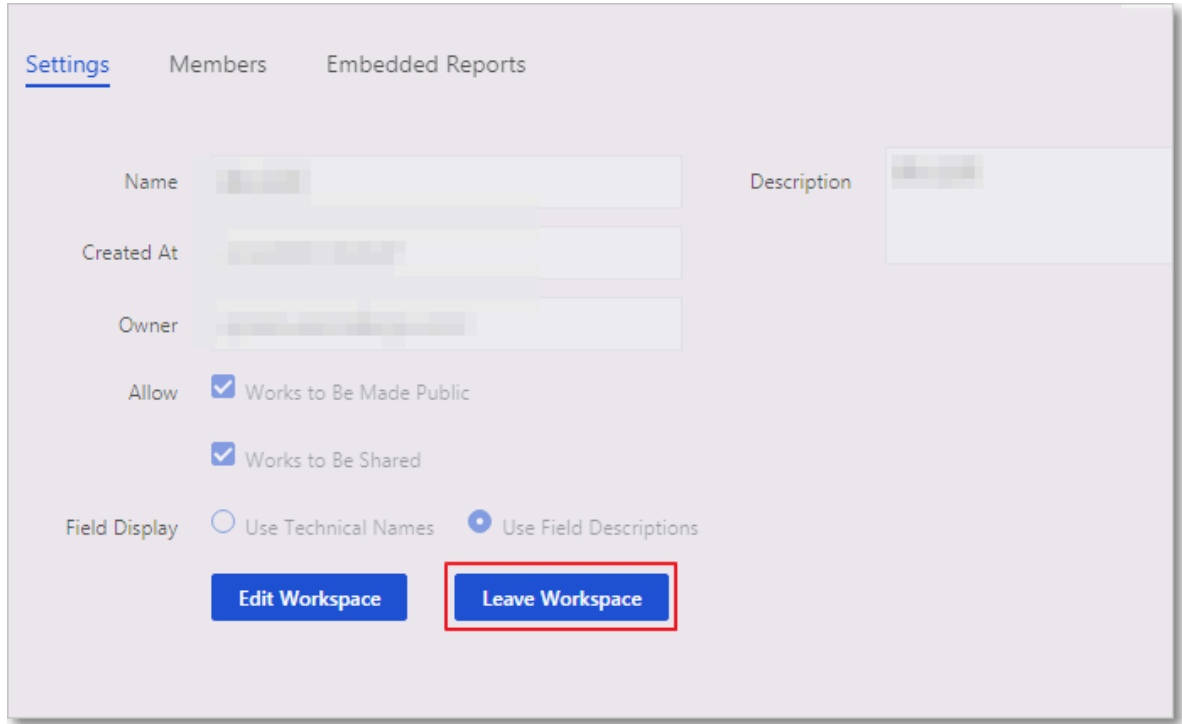
#### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the **Settings** icon.
3. In the left-side navigation pane, click **Workspaces**.
4. Find the target workspace and click the **Settings** tab.

**5. Click Leave Workspace to leave the current workspace, as shown in [Figure 4-215](#):**

*Leave the workspace.*

Figure 4-215: Leave the workspace



#### 4.7.15 Transfer a workspace to another owner

This topic describes how to transfer a workspace to another owner.

##### Context

If the owner of a workspace needs to be removed from the workspace, you can transfer the workspace to another member. The new owner does not need to be an administrator of the workspace. You can transfer the workspace to any member in the workspace.

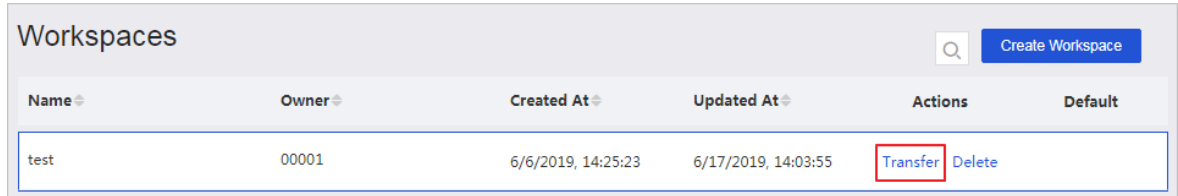
##### Procedure

1. [Log on to the Quick BI console.](#)
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon.
3. In the left-side navigation pane, click Workspaces.
4. Find the target workspace and click Transfer in the Actions column.



5. Enter the alias of the new owner and click OK, as shown in [Figure 4-216: Transfer a workspace](#).

Figure 4-216: Transfer a workspace



Name	Owner	Created At	Updated At	Actions	Default
test	00001	6/6/2019, 14:25:23	6/17/2019, 14:03:55	<a href="#">Transfer</a> <a href="#">Delete</a>	

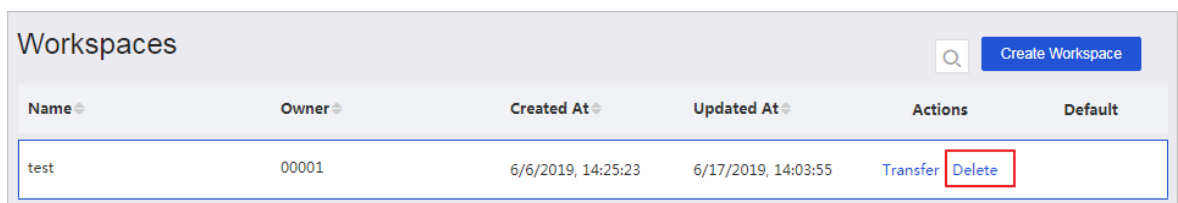
### 4.7.16 Delete a workspace

This topic describes how to delete a workspace.

#### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon.
3. In the left-side navigation pane, click Workspaces.
4. On the Workspaces page, find the target workspace and click Delete in the Actions column, as shown in [Figure 4-217: Delete the workspace](#).

Figure 4-217: Delete the workspace



Name	Owner	Created At	Updated At	Actions	Default
test	00001	6/6/2019, 14:25:23	6/17/2019, 14:03:55	<a href="#">Transfer</a> <a href="#">Delete</a>	

### 4.7.17 Add a member to a workspace

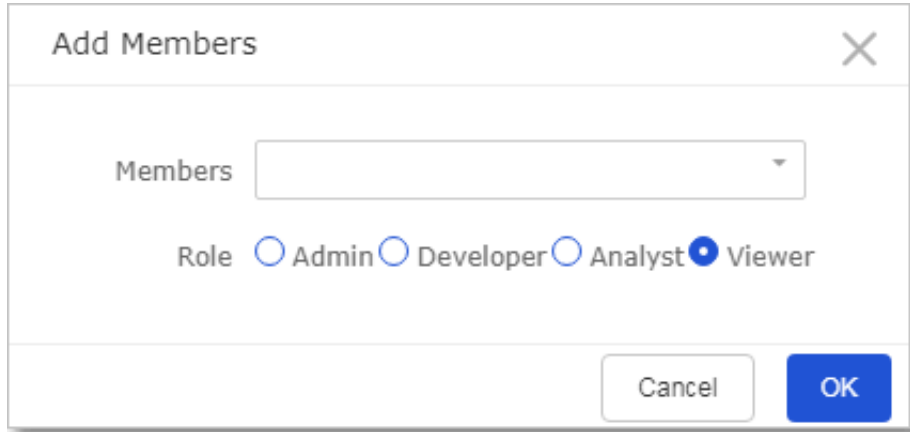
This topic describes how to add a member to a workspace.

#### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Workspaces page.
3. Click the Members tab.
4. Click Add Members.

5. Enter the member account and specify a role for the member, as shown in [Figure 4-218: Add a member to the workspace](#).

Figure 4-218: Add a member to the workspace

A dialog box titled "Add Members" with a close button (X) in the top right corner. It contains a "Members" label followed by a text input field. Below this, there is a "Role" label followed by four radio button options: "Admin", "Developer", "Analyst", and "Viewer". The "Viewer" option is selected. At the bottom right, there are two buttons: "Cancel" and "OK".

#### 4.7.18 Edit settings of a workspace member

This topic describes how to edit settings of a workspace member.

##### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Workspaces page.
3. Click the Members tab.
4. Find the target member and click Edit in the Actions column.
5. Change the settings of the member and click OK.

#### 4.7.19 Search for a member in a workspace

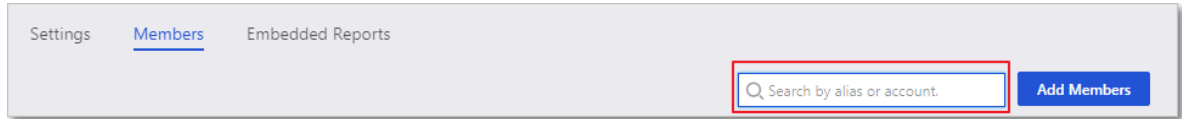
You can search for members on the Members tab page.

##### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Workspaces page.
3. On the Workspaces page, select the target workspace.
4. Click the Members tab. All members in the current workspace are listed on this page.
5. Enter an alias or account in the search box.

6. Click the Search icon to search for the member, as shown in [Figure 4-219: Search for a member in the workspace](#).

Figure 4-219: Search for a member in the workspace



## 4.7.20 Delete a member from a workspace

This topic describes how to delete members from a workspace.

### Procedure

1. [Log on to the Quick BI console](#).
2. On the homepage of the Quick BI console, or in a workspace, click the Settings icon and navigate to the Workspaces page.
3. On the Workspaces page, click the Members tab.
4. Find the target member and click Delete.
5. Select a new owner from the drop-down list. Data object owned by the member to be deleted will be transferred to the new owner.
6. Click OK to delete the member.

## 4.8 Permissions

### 4.8.1 Overview

Permission management includes data object management and row-level permission management.

Data objects include data sources, datasets, workbooks, dashboards, and BI portals. Data object management includes the management of data objects in your personal workspace and in a workspace.

## 4.8.2 Data objects

**Data objects include data sources, datasets, dashboards, workbooks, and BI portals.**

Share data objects in a workspace

**You can share workbooks, dashboards, and BI portals with other users. Shared works can be accessed by other users in the read-only mode. They cannot modify or delete the data objects, or save a copy of the data objects.**

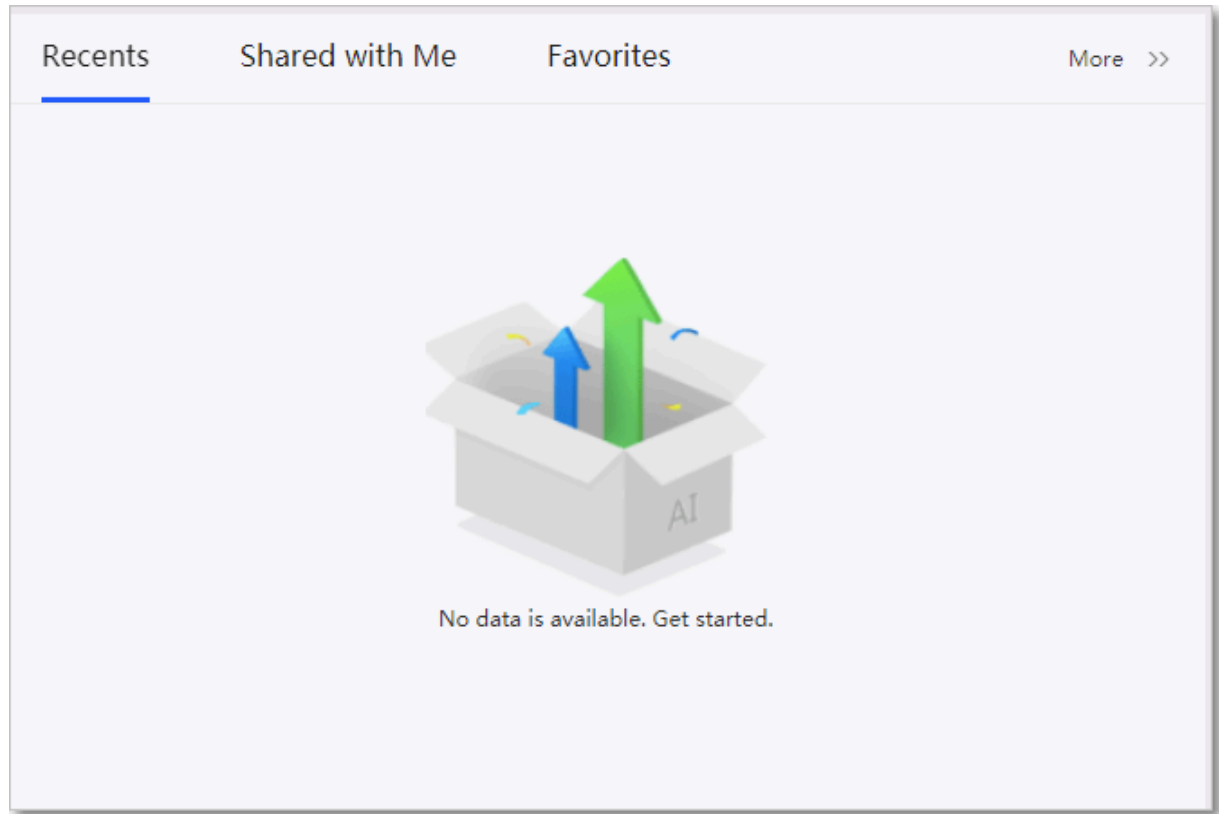
- **Only the owner of the data objects and administrators of the workspace have the permission to share data objects.**
- **If the Works to Be Shared check box is cleared in the workspace settings, no data objects in the workspace can be shared.**
- **Currently, data objects can be shared within their workspace only. You cannot share data objects with Alibaba Cloud accounts that are excluded from the organization.**

**By default, all data objects are accessible to members of the workspace.**

**You can share data objects with specific users within the organization. The user that you want to share data objects with does not need to be a member of the current workspace.**

Users that you share a data object with can view and access the data object on the **Shared with Me** tab page, as shown in [Figure 4-220: View shared data objects](#).

Figure 4-220: View shared data objects



You can also make dashboards public. Public dashboards can be accessed by all Internet users. We recommend that you do not make dashboards public if they contain business data.

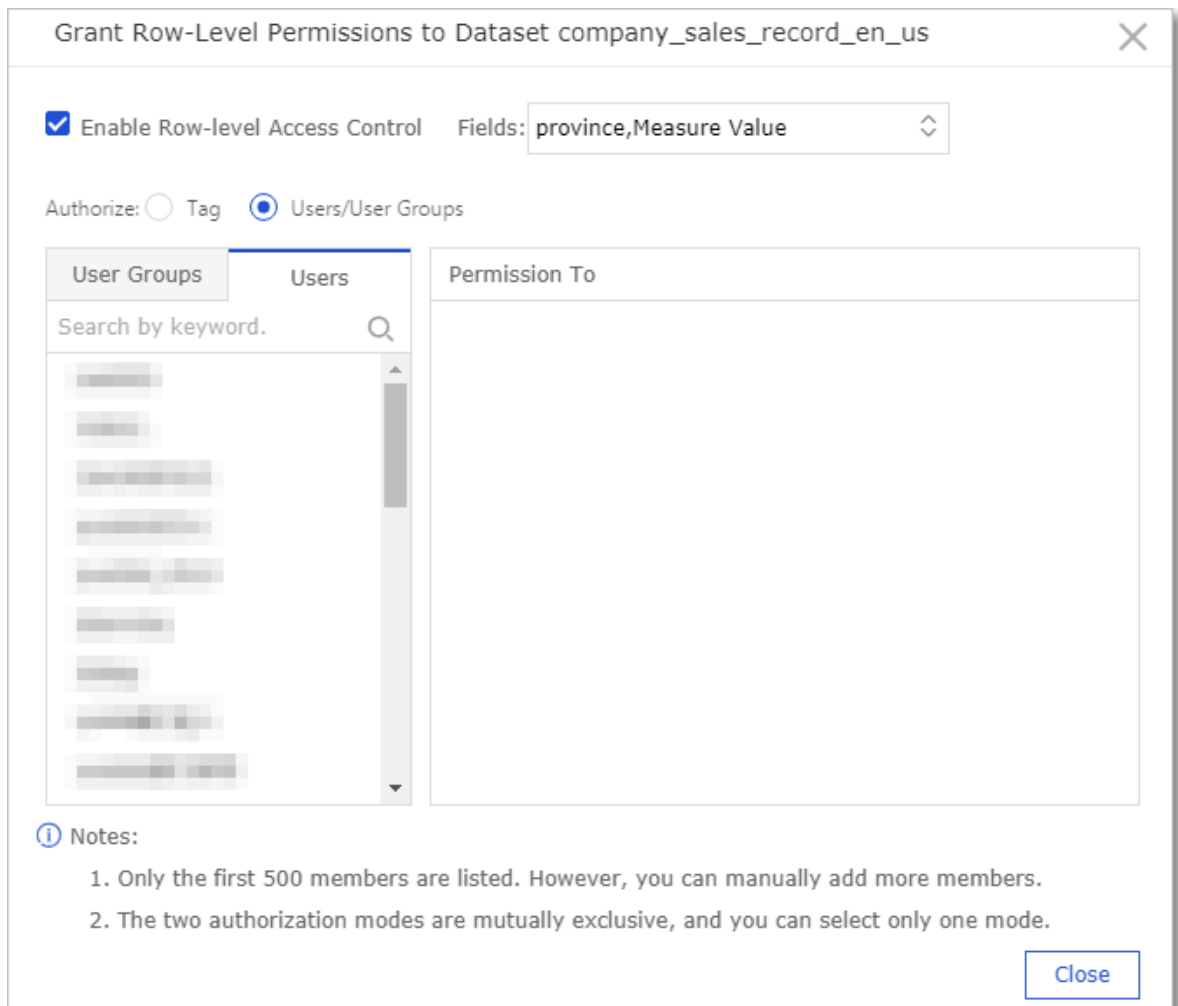
### 4.8.3 Row-level permission management

Row-level permissions are granted based on a specific dataset. Quick BI supports the following authorization modes: Users/User Groups and Tag. The Use/User Groups mode applies to scenarios that involve a small number of members. The Tag mode applies to scenarios that involve a large number of members. The Tag mode authorizes all users at once instead of authorizing users or user groups separately. In scenarios that involve a large number of members, the Tag mode reduces costs and simplifies the procedure to grant row-level permissions. This makes it easier for you to manage row-level permissions.

Users/User Groups mode

#### 1. Log on to the Quick BI console.

2. Select the target workspace. For more information about creating workspaces, see [Create a workspace](#).
3. In the left-side navigation pane, click Datasets.
4. Select the target dataset. Click the More icon in the Actions column or right-click the dataset.
5. Select Grant Row-Level Permissions.
6. Select the Enable Row-Level Access Control check box and the Users/User Groups check box to enable row-level permissions.
7. Click the drop-down icon of Fields and select the fields that the authorization is based on, for example, province and Measure Value, as shown in the following figure.



Elements of Measure Value are the measures in the dataset. By granting row-level permissions based on the Measure Value field, you can specify the measures available to different users.

8. In the Permission To area, click the province field and a list containing all values of the province field appears.
9. Select a user and choose values from the province field to grant permissions to the user, as shown in the following figure.

Grant Row-Level Permissions to Dataset company\_sales\_record\_en\_us

☒ Enable Row-level Access Control Fields: province, Measure Value

Authorize: ☐ Tag ☒ Users/User Groups

**User Groups** **Users**

Search by keyword. Q

QuickBI0415  
venus\_verify  
hln\_verify  
venus  
hln  
Quick\_BItest001  
Quick\_BItest  
quick\_BI  
zs

**Permission To**

✓ **Configured Permissions**

✓ Measure Value

✓ province

✓ **Inherited Permissions (Lock...)**

✓ Measure Value

✓ province

**Select** **Specify**

Search by keyword. Q

☐ All  
☐ Anhui  
☐ Beijing  
☐ Fujian  
☐ Gansu  
☐ Guangdong  
☐ Guangxi

**Add**

**Notes:**

1. Only the first 500 members are listed. However, you can manually add more members.
2. The two authorization modes are mutually exclusive, and you can select only one mode.

**Close**

In this example, the user can view data of Shanghai and Yunnan. Data of other provinces is unavailable to this user.



**Note:**

If you grant permissions based on a field of a dataset, you need to specify whether all users in the workspace have the permission to access this dataset. Otherwise, when other users attempt to access reports created based on the dataset, the system denies all access requests by default.

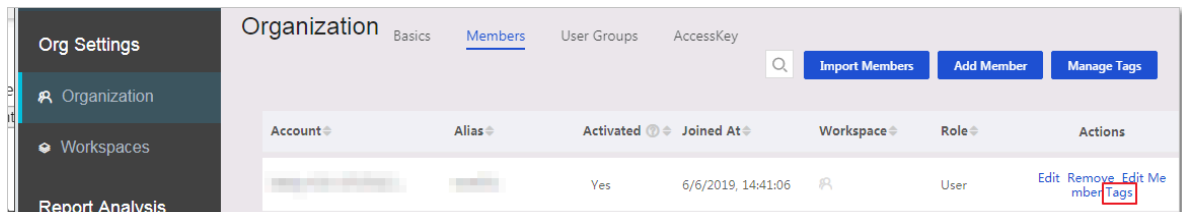
10. Click Add to grant permissions.

## Tag mode

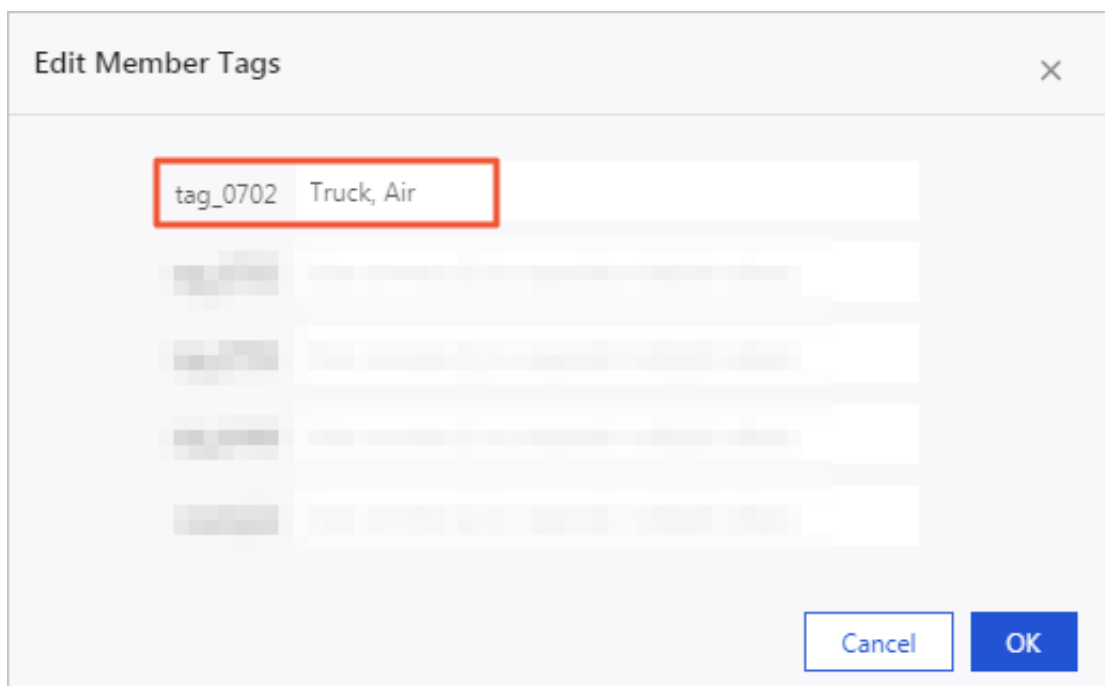
**Example: Authorize a user to access the Truck and Air data of the shipping\_type dimension in the dataset company\_sales\_record\_en\_us.**

**Edit member tags**

1. Click the Settings icon and click Organization. Find the target user and click Edit Member Tags, as shown in the following figure.



2. In the Edit Member Tags dialog box, set the value of the example tag to Truck, Air and click OK.



After you edit the member tag, you need to specify the tag in the Grant Row-Level Permissions dialog box.

**Specify the member tag**

1. Find the dataset company\_sales\_record\_en\_us and click the More icon in the Actions column, or right-click the dataset and select Grant Row-Level Permissions.



2. Select the Enable Row-Level Access Control check box and the Tag check box to enable row-level permissions.
3. Set Fields to shipping\_type, Tag to example, and click OK.

Grant Row-Level Permissions to Dataset company\_sales\_record\_en\_1105

☒ Enable Row-level Access Control    Fields: shipping\_type

Authorize: ☒ Tag    ☐ Users/User Groups

Field	Tag	Actions
shipping_type	example	Delete

Cancel    OK

After the member tag is specified, the user can only access the Air and Truck data of the shipping\_type dimension.

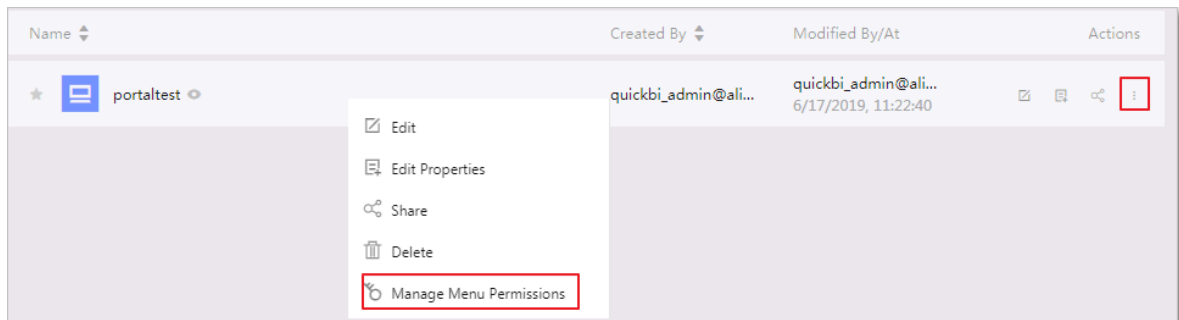
#### 4.8.4 Configure BI portal menu permissions

Workspace administrators can grant BI portal menu permissions to users.

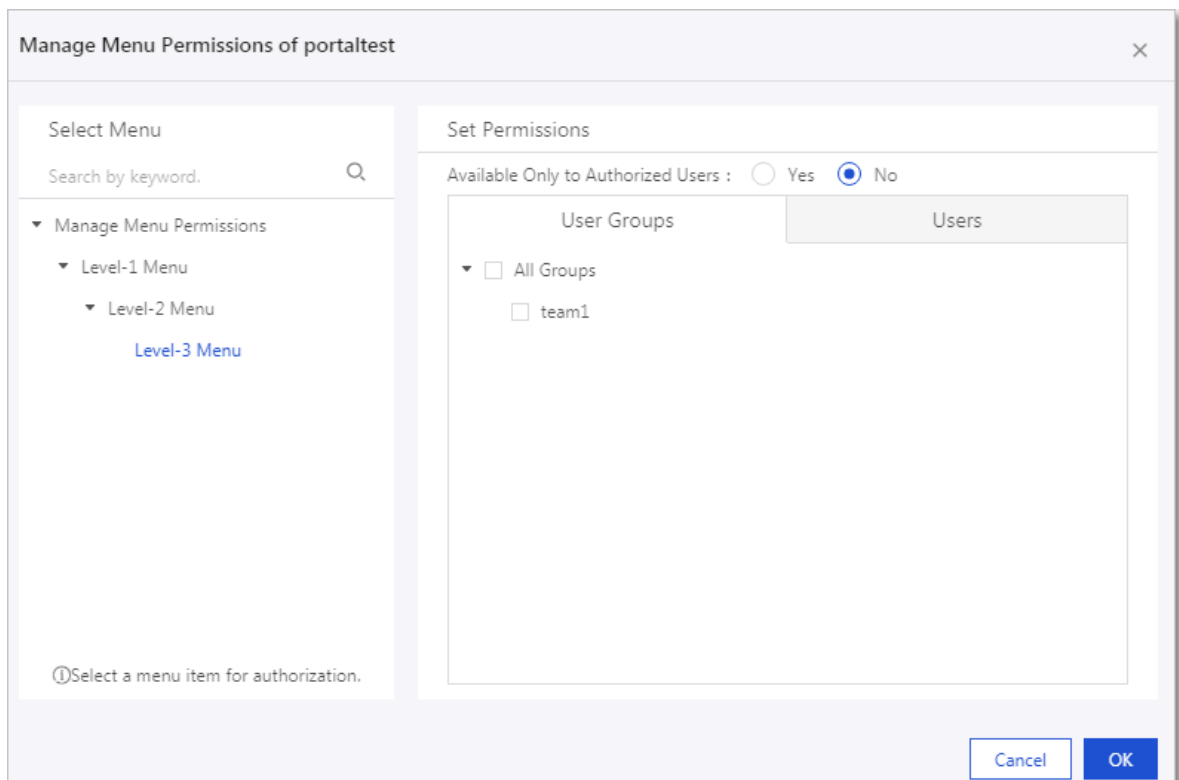
You can grant menu permissions to a specific user or user group. Procedure:

1. Log on to the Quick BI console.
2. Select the target workspace. For more information about creating a workspace, see [Create a workspace](#).
3. In the left-side navigation pane, click BI Portals.

- On the BI Portals page, select the target portal and click the More icon in the Actions column, or right-click the target portal and select Menu Permissions, as shown in the following figure.



- In the Manage Menu Permissions dialog box, select the target menu, specify whether the menu is available only to authorized users, and select the users or user groups that you need to authorize.



**Note:**

For more information about Available Only to Authorized Users, see the following description:

- Yes:** Only authorized user groups and users have the permission to view the specified menu.
- No:** All users have the permission to read this menu.

6. Click OK.

### 4.8.5 Share data objects in the personal workspace

Only the owner of a data object has the permission to share the data object.

#### Context

In the personal workspace, you can share workbooks, dashboards, and BI portals. Shared data objects can be accessed by other users in the read-only mode. They cannot modify or delete the data objects, or save a copy of the data objects.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. In the left-side navigation pane, click Dashboards.
3. On the Dashboards page, select the target dashboard and click the Share icon in the Actions column.
4. Enter the usernames that you want to share the dashboard with, and specify the expiration date.
5. Click Save to share the dashboard.

### 4.8.6 Share a data object in a workspace

In a workspace, you can share workbooks, dashboards, and BI portals. Shared data objects can be accessed by other users in the read-only mode. They cannot modify or delete the data objects, or save a copy of the data objects.

#### Context

Only the owner of the data objects and administrators of the workspace have the permission to share data objects. Data objects can be shared within their workspace only. You cannot share data objects with Alibaba Cloud accounts that are excluded from the organization.

If the Works to Be Shared check box is cleared in the workspace settings, no data objects in the workspace can be shared.

#### Procedure

1. [Log on to the Quick BI console.](#)
2. Select the target workspace.
3. In the left-side navigation pane, click Dashboards.

4. On the Dashboards page, select the target dashboard and click the Share icon in the Actions column.
5. Enter the alias or accounts of the users that you want to share the dashboard with, and specify the expiration date.
6. Click Save to share the dashboard.

#### 4.8.7 Publish data objects that are stored in a personal workspace

You can publish data objects that are stored in a personal workspace. All Internet users can visit the URLs that point to published data objects. We recommend that you do not publish data objects that include sensitive business data.

##### Prerequisites

You have purchased Quick BI.

##### Context

In a personal workspace, you can publish dashboards and workbooks.

This topic takes a dashboard as an example to describe how to publish data objects in a personal workspace.

##### Procedure

1. [Log on to the Quick BI console.](#)
2. Select a workspace.
3. Click Dashboards to go to the Dashboards page.
4. Select the target dashboard and click the More icon and select Make Public.
5. Specify an expiration date and click Make Public.

A URL is generated and appears in the Make Public dialog box. You can copy and paste the URL into the address bar of your browser, and then access the dashboard by using the URL.

#### 4.8.8 Make a data object public in a workspace

After you make a data object public, every Internet user can access the data object. Therefore, we recommend that you do not make data objects public if they contain business data.

##### Procedure

1. [Log on to the Quick BI console.](#)

2. Select the target workspace.
3. In the left-side navigation pane, click Dashboards.
4. Select the target dashboard and click the Make Public icon in the Actions column.
5. Specify the expiration date and click Make Public.

Copy and paste the generated URL to the address bar in your browser. You can then access the dashboard.

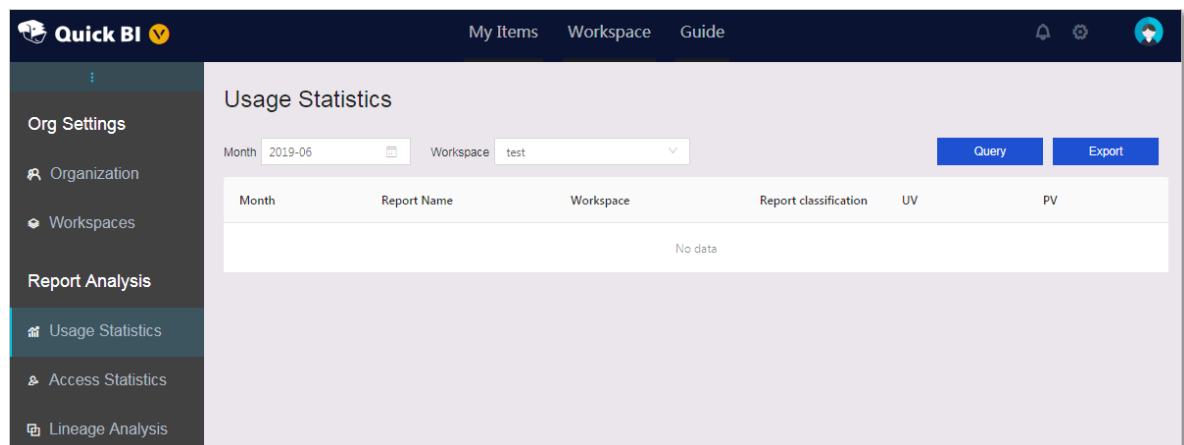
## 4.9 Report statistics

### 4.9.1 Usage statistics

Usage statistics allow you to track the UV and PV of a specific month.

1. On the homepage of the Quick BI console, click the Settings icon.
2. In the left-side navigation pane, click Usage Statistics.
3. On the Usage Statistics page, select a month and workspace and click Query, as shown in [Figure 4-221: Usage statistics](#).

Figure 4-221: Usage statistics



4. You can click Export to export the statistics data to a local device. The statistics data is exported in the Excel format.

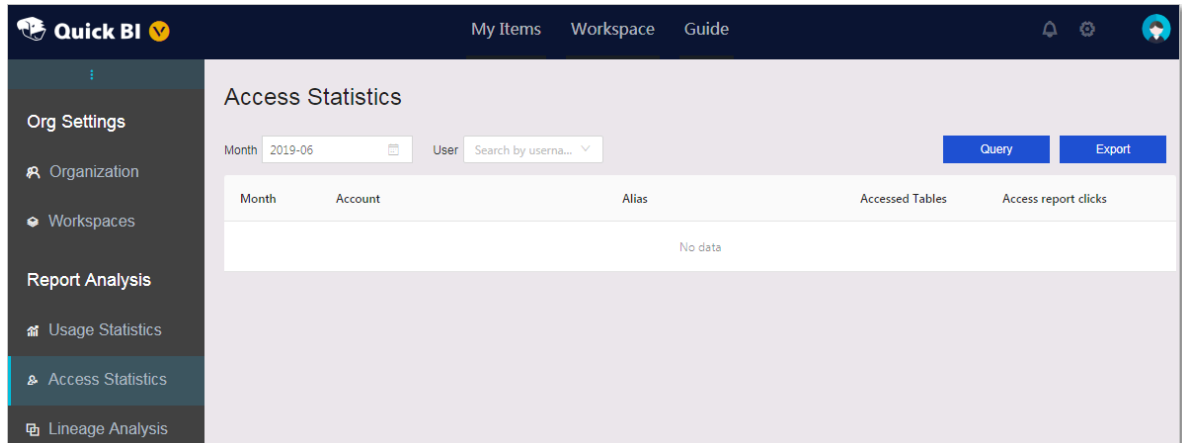
### 4.9.2 Lineage analysis

Lineage analysis allows you to query information about a specific report, including the workspace, report type, dataset, charts, and data source name.

1. On the homepage of the Quick BI console, click the Settings icon.

2. In the left-side navigation pane, click Lineage Analysis.
3. On the Lineage Analysis page, select the workspace, report type, and report name of the target report, and click Query.

Figure 4-222: Lineage analysis



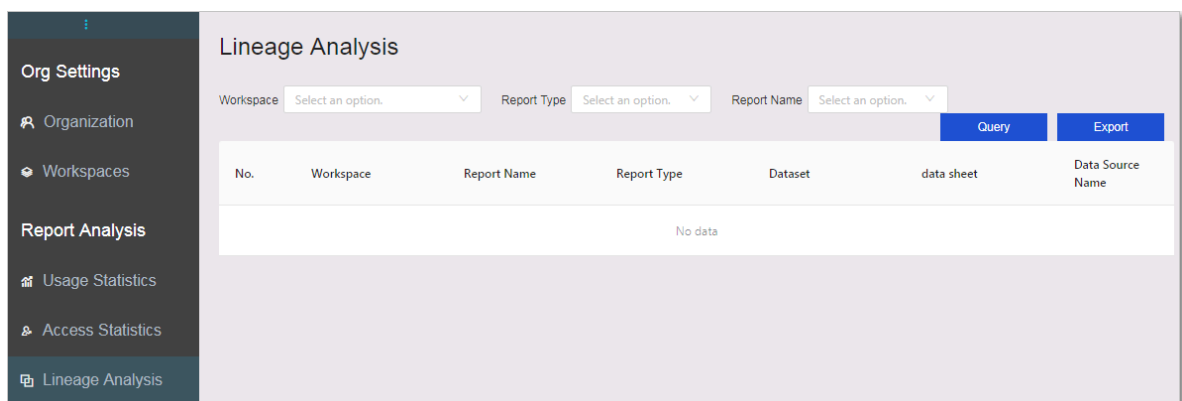
4. You can click Export to export the analysis data to a local device. The analysis data is exported in the Excel format.

### 4.9.3 Access statistics

Access statistics allow you to track the number of reports that you have created during a specific month, and the number of clicks on a specific report.

1. On the homepage of the Quick BI console, click the Settings icon.
2. In the left-side navigation pane, click Access Statistics.
3. On the Access Statistics page, select a month, enter the member alias, and click Query.

Figure 4-223: Access statistics



- 4. You can click Export to export the statistics data to a local device. The statistics data is exported in the Excel format.**

## 5 DataQ - Smart Tag Service

---

### 5.1 What is DataQ - Smart Tag Service?

Generally, DataQ - Smart Tag Service is a tag-oriented service, which establishes a unified logic model across multiple schemas. By using the tag model view, developers can integrate the data service modules with profile analysis, rule warnings, text mining, personalized recommendations, relational networks, and other business scenarios. This helps developers use APIs to quickly build applications.

Alibaba Cloud DataQ - Smart Tag Service provides a data IDE to accelerate the development and implementation of big data applications. This product helps developers integrate various big data products based on their business needs, which reduces most of the engineering workload that is necessary for building big data applications. By using the product together with relevant industry application solutions, developers who are less experienced in development of big data applications can quickly build big data applications. This can help realize the true value of big data over a relatively short period of time.

DataQ - Smart Tag Service can help data developers to build models based on data tables. This helps extract the business data and convert the data to objects that can be understood by business personnel to accelerate application development. For example, doctors can understand patients, doctors, diseases, medical records, and other real objects, rather than multiple tables of unrelated facts and figures, such as a user table. DataQ - Smart Tag Service tags allows you to build a business-oriented model and convert data to objects that can be understood throughout the medical industry. For example, if a patient is considered as an object, the object must be assigned with age, gender, blood type, pregnancy, and other tag information. Meanwhile, a link can be established between the patient and their past medical records. The object that has tags can be quickly understood and analyzed by the doctor.

DataQ - Smart Tag Service helps you to build objects and provides actual content for applications to analyze. Meanwhile, application developers can intuitively



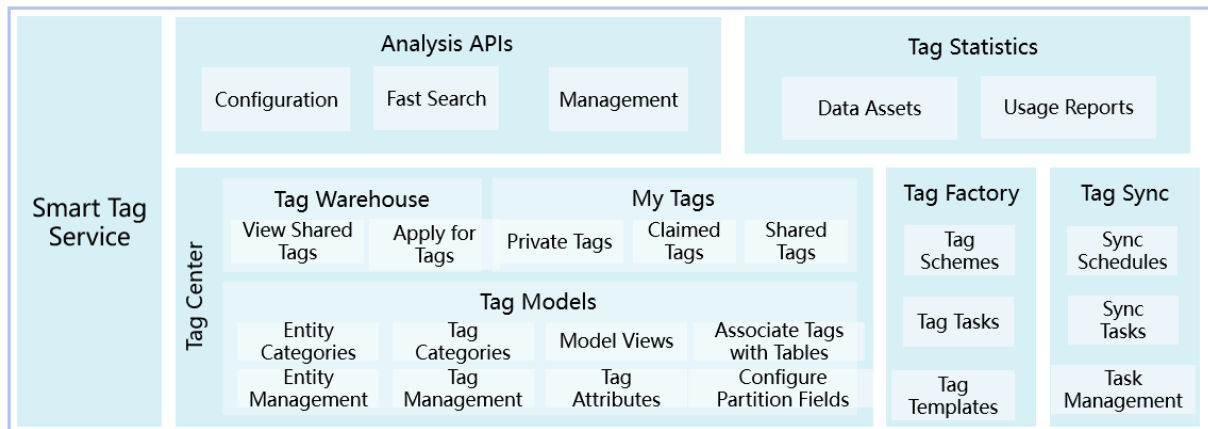
understand the data objects and directly process, derive, and call the business-oriented objects and tags.

DataQ - Smart Tag Service provides the following benefits:

- Simplifies the integration with complex systems because application developers do not need to have a deep understanding of multiple underlying computing and storage resources.
- Helps the IT team to manage data usage by providing data service APIs. This helps to avoid any duplication and redundancy of resources.

The IT team can share tags that are frequently used in business scenarios. You can apply for using these shared tags. After obtaining authorization to use these tags, you can perform corresponding computations by calling APIs. You can also generate code that can be independently deployed by configuring parameters in the console. This helps provide an easy way to build the corresponding big data product.

DataQ - Smart Tag Service includes the homepage, Tag Center, Analysis APIs, Tag Factory, Dashboards, and Tag Sync modules.



## 5.2 Quick start

### 5.2.1 Log on to the DataQ console

This topic describes how to log on to the DataQ console.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment

personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.

- We recommend that you use the Chrome browser.

## Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username `super`. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. In the left-side navigation pane, choose Big Data > DTBoost.
6. Select a department in the drop-down list and click DTBoost.

If this is the first time you log on to DataQ as an Alibaba Cloud RAM user, the administrator must create a workspace and add your RAM user account to the workspace. For more information, see [Create workspaces](#). If this is not your first login, you can skip this step.



### Note:

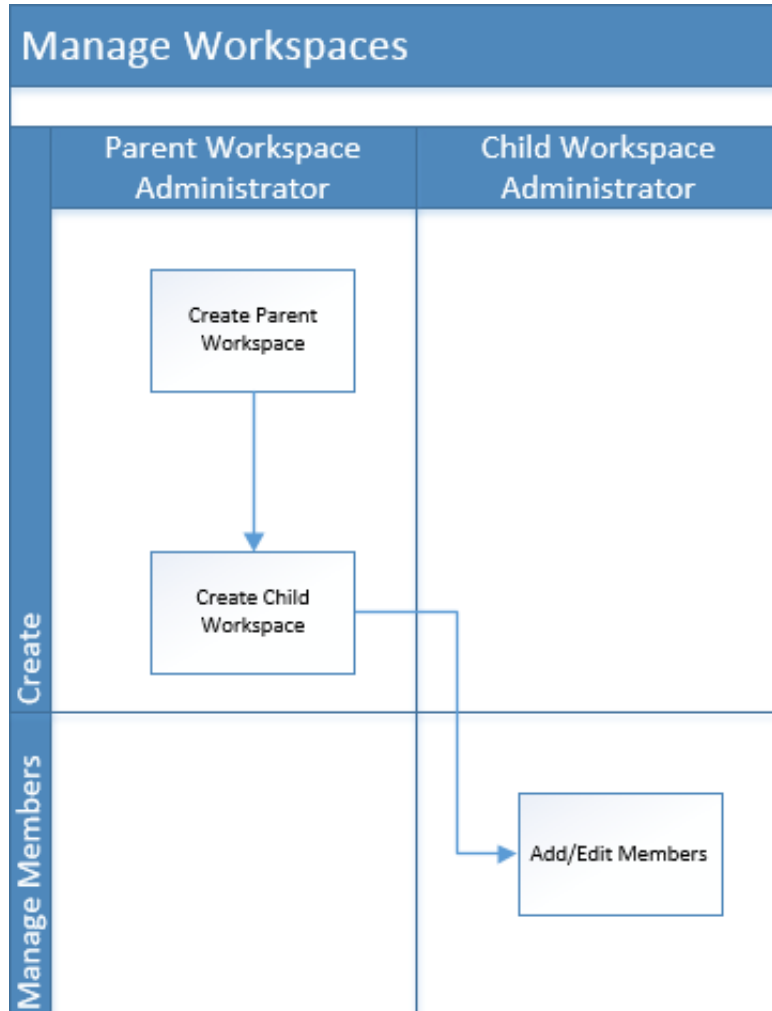
Before you log on to DataQ for the first time, the tenant administrator needs to add your account to the workspace. After your account is added to the workspace, the homepage is automatically displayed each time you log on to DataQ.

7. Go to the Homepage, you can click the menus in the top navigation bar to perform all corresponding operations.

## 5.2.2 Create workspaces

If this is the first time you log on to DataQ as an Alibaba Cloud RAM user, the administrator must create a workspace and add your RAM user account to the workspace.

The workflow for creating a workspace is shown in the following figure.



1. *Log on to the DataQ console.*
2. Move the pointer over the upper-right corner, click Workspaces from the shortcut menus.
3. Enter the Workspaces page, click Create in the right-side workspace.
4. In the Create Workspace dialog box that appears, enter the required information to complete the configurations.

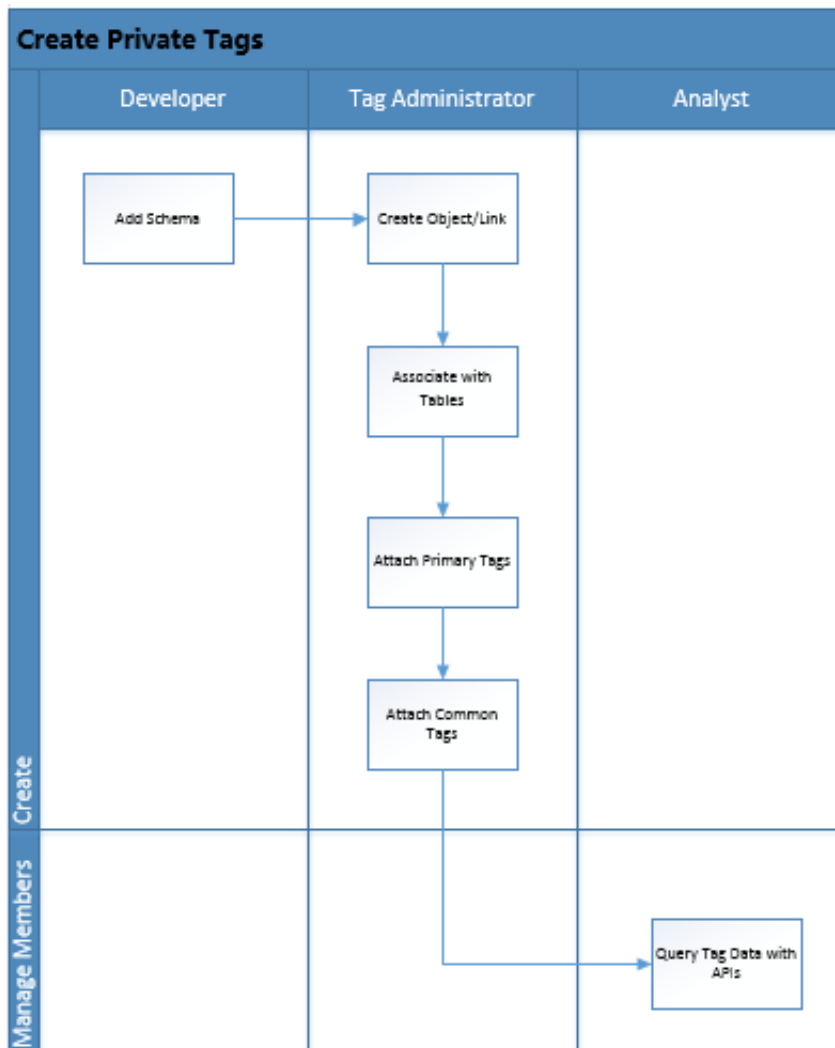
5. Click OK. A new workspace is created.

After a workspace is created, you can view it in the workspace list. You can also create a child workspace based on the selected parent workspace by clicking Create in the right-side workspace.

6. After the workspace is created, click Edit Members in the Actions column to add or edit members of the workspace.

### 5.2.3 Create private tags

The workflow for creating a private tag is shown in the following figure.



1. In the top navigation bar, select Tag Center. In the left-side navigation pane, select Schemas.

DataQ - Smart Tag Service supports the following schemas: ApsaraDB for RDS, MaxCompute, and AnalyticDB.



**Note:**

We recommend that you use ApsaraDB for RDS.

2. In the upper-right corner of the right-side workspace, click Add Schema.

- Add MaxCompute as the schema. The following table lists the configuration items.

Configuration item	Description
Schema code	The schema code can contain letters , numbers, and underscores (_) and must start with a letter.
Schema type	MaxCompute.
In VPC	No.
Project Name	The MaxCompute project name.
Endpoint	Use the default values without modifying them.
tunnelEndpoint	Use the default values without modifying them.
AccessKey ID and AccessKey secret	Your AccessKey ID and AccessKey secret that are issued by Alibaba Cloud.
Description	The description of the MaxCompute schema.

- Add ApsaraDB for RDS as the schema. The following table lists the configuration items.

Configuration item	Description
Schema code	The schema code can contain letters , numbers, and underscores (_) and must start with a letter.
Schema type	ApsaraDB for RDS.
In VPC	No.

Configuration item	Description
Domain name	The host address of the schema.
Database name	The database name that is used in ApsaraDB for RDS.
Port	The port number that can access the ApsaraDB for RDS instance.
Username	The username that is used to log on to ApsaraDB for RDS.
Password	The password that is used to log on to ApsaraDB for RDS.
Description	The description of the ApsaraDB for RDS schema.

- Add AnalyticDB as a schema. The following table lists the configuration items.

Configuration item	Description
Schema code	The schema code can contain letters , numbers, and underscores (_) and must start with a letter.
Schema type	AnalyticDB (ADS).
In VPC	No.
Domain name	The host address of the schema.
Database name	The database name that is used in AnalyticDB (ADS).
Port	The port number that can access the AnalyticDB (ADS) instance.
Username	The username that is used to log on to AnalyticDB (ADS).
Password	The password that is used to log on to AnalyticDB (ADS).
Description	The description of the AnalyticDB (ADS) schema.

3. Enter all the required information, and then click OK.

**Note:**

You must configure a whitelist for all types of schema excluding MaxCompute.

4. In the left-side navigation pane, click Tag Models to open the Tag Models page. On the right-side workspace, you can search for all objects or objects of the current category on this page.
5. On the upper-right corner of the Tag Models page, click Create to create an object or link.
6. Select an object or link, click Manage Tables, and the Associated Tables page appears.

On the upper-right corner of the page, click Associate Tables and configure the association.

7. Then, click Next to go to the Associate Tables page. On the page, the columns that do not attach tags are displayed. Click a column name and a Primary Key configuration pane appears on the right side. After you turn on the Primary Key switch and enter the required information, click Attach in the upper-right corner to complete the operation.
8. Then, select other column names, turn off the Primary Key switch and enter required information to attach common tags. After the operation is complete, click Attach.



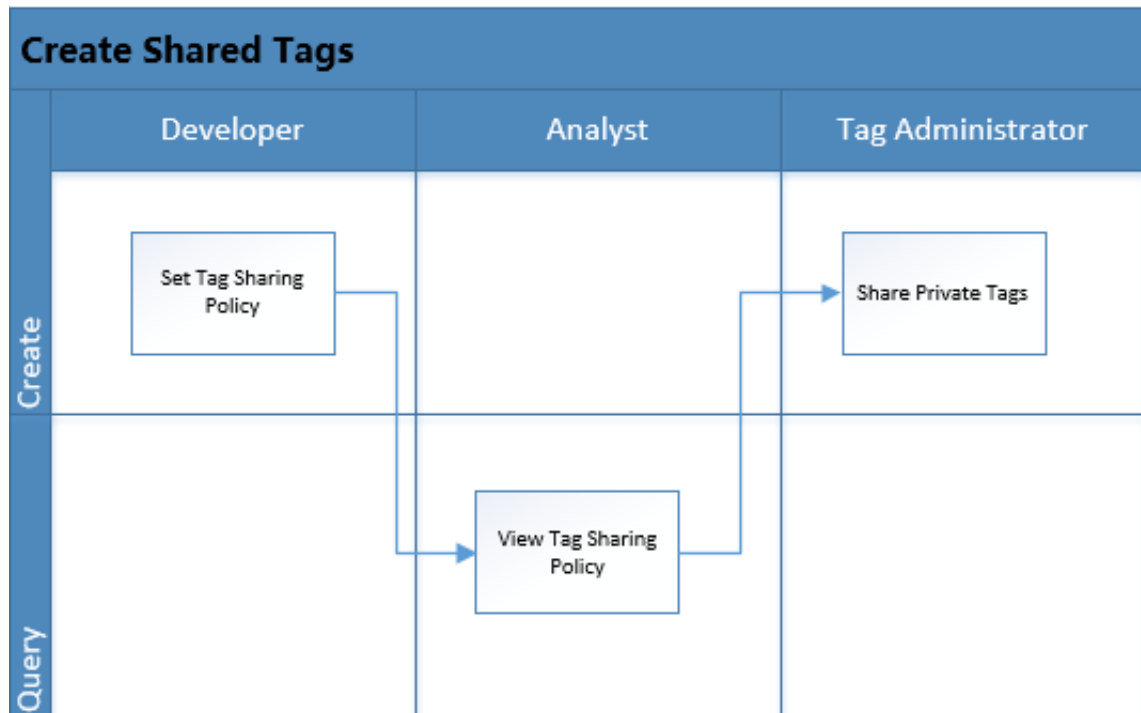
**Note:**

If you create a link between two objects and want to attach tags to the link, you need to attach tags to the two objects at the same time.

9. In the top navigation bar, choose Analysis APIs > API Factory to open the corresponding page. In the left-side navigation pane, select a schema from the Schema drop-down list. Enter the search statement in the query editor and run the code. At the bottom, the query results are displayed.

## 5.2.4 Create shared tags

The workflow of creating a shared tag is shown in the following figure.



1. Choose Tag Center > My Tags, and click the Private Tags tab. You can search for private tags by selecting a schema.

2. Select the tag to be shared and click Public in the Actions column.

You can also select multiple tags and click Public at the bottom to share multiple private tags at a time.

3. In the Share Tags dialog box, select the tags, enter a reason, and click OK.

4. Click the Shared Tags tab to view shared tags.

On the Shared Tags page, you can revoke tag sharing and view authorization. If you revoke tag sharing for a shared tag, the tag becomes invisible to other users.

5. On the top of the Shared Tags tab page, click Tag Policy. In the dialog box that appears, all workspaces of the current schema appear in the Visible Range section by default. You can remove some workspaces to change the visible range as required.

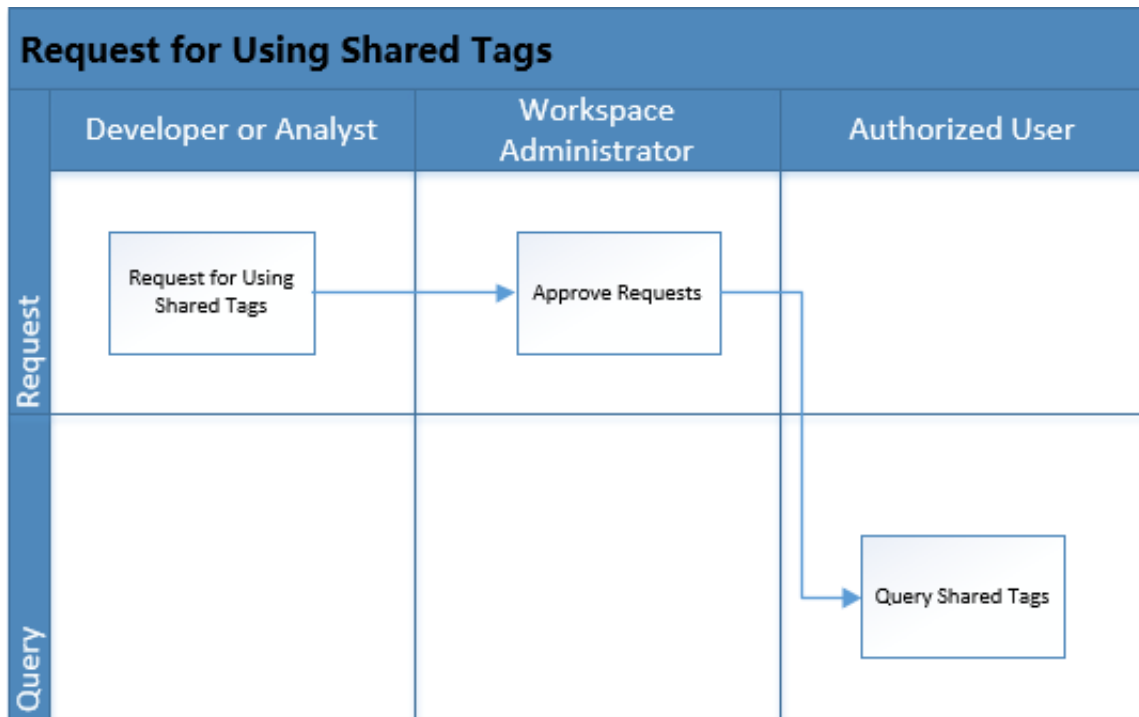
6. On the Private Tags tab page, select one or more shared tags and click Revoke to revoke tag sharing for these tags.

In the dialog box that appears, enter the description and reason, and click OK to complete the operation.

## 5.2.5 Apply for using shared tags

The workflow of applying for using a shared tag is shown in the following figure.





1. Choose Tag Center > Tag Warehouse. On the Tag Warehouse page, all shared tags appear.



**Note:**

You cannot apply for using the shared tags of your own workspace. However, you can apply for using the shared tags of the upper-level workspace.

2. On the shared tag list, select a shared tag and click Apply in the Actions column to apply for using the shared tag.

You can also select multiple shared tags and click Apply for Tags at the bottom to apply for using these shared tags at a time.

3. After the application is submitted, wait for a reviewer to review the application. The reviewer can decide whether to approve or reject the application. If you want to view the status of your application, click Applications in the message that appears after the application is submitted.

You can also cancel the application on the Applications page.

To cancel the application, click Revoke in the Actions column. The application enters the Revoked state.

4. After the reviewer approves the application on the Approvals page, click My Tags in the left-side navigation pane. Then, click the Claimed Tags tab to view the tags that you are authorized to use.



**Note:**

You can only view the shared tags of the upper-level workspace that you are authorized to use on the Claimed Tags tab.

If you do not need to use some tags, select these tags and click Release Tag at the bottom.

After the tags are released, you can choose Tag Center > Tag Warehouse to view the tags released on the Claimed Tags tab.

## 5.3 Analysis APIs

### 5.3.1 Overview

The Analysis APIs page of DataQ - Smart Tag Service consists of three modules: APIs, API factory, and fast search.

The APIs are exported for special calls by various applications through the TQL capability provided by Analysis APIs. This is based on the behavior track and tag construction of relationships between objects and links, and is designed to help achieve rapid application development.

On the Analysis API page, you can create, view, debug APIs, and manage the API categories. API factory is a business function module built on a tag-based view. You can use tags as dimensions to perform unified computations for data across multiple computing resources by configuring APIs or calling API operations.

The combination of API factory and logic modeling reduces the workload and offers high scalability. Especially when the big data environment needs to consolidate data from multiple systems, it is difficult to design a single plan to meet all data requirements. This dynamic logical modeling method has high scalability.

From the perspective of applications, the tag model allows you to calculate and query detailed data in the institute tag system without using the complex data structure. This also helps you to optimize the process of data development and application development.

### 5.3.2 APIs

On the APIs page, all APIs are displayed on a list. You can view, edit, publish, and delete APIs.

The API list allows you to analyze and query the generated APIs and configure parameters during the analysis and query process. You can manage APIs, view details, and debug APIs. Publishing APIs to the API store by one click is also supported if you have deployed API Service Bus.

- To view the API details, click Details in the Actions column.
- To edit the code of an API, click Edit in the Actions column.
- To publish the API to API store, click Publish in the Actions column.
- To delete an API, click Delete in the Actions column.
- In the upper-left corner of the workspace, click the Create Category icon, and the Create Category dialog box appears. Enter the category name, and click OK.

### 5.3.3 API factory

The API Factory page allows you to query tags of business objects, shared tags, and object-link models by using Tag Query Language (TQL). You can view the query results in table or JSON format. By debugging an API on the API Factory page, you can check each process of TQL and its time performance.

1. Choose Analysis APIs > APIs. On the page that appears, click Create API in the upper-right corner. Alternatively, you can choose Analysis APIs > API Factory to create an API.
2. Select the schema, analysis subject (object or link), query type (TQL or JSON), and code sample.
3. Enter code in the query editor and configure variables in the right-side pane.
4. After the query statements are created, click Test to view the results, SQL statements that are run, and procedure details.
5. Click Save as Virtual Table. In the Save as Virtual Table dialog box, enter the virtual table name and description, and click OK.
6. Click Generate API. In the Generate API dialog box, enter the API name, API path, API category, and API description, and click OK.

The API factory supports multiple queries at the same time. For example, you can enter an ID card number to query the information of a person. If you enter multiple

ID card numbers, you can query the information of the corresponding persons at a time.

If the selected schema is a MaxCompute partitioned table, you need to add features related to default partition query.

1. Select the schema, analysis subject (object or link), query type (TQL or JSON), and code sample. Turn on the Filter Queried Partitions switch.
2. Enter code in the query editor and click Filter Queried Partitions in the upper-right corner.
3. In the Filter Condition drop-down list, select one of the following conditions: all partitions, maximum partitioning column value, minimum partitioning column value, fixed value, and system date. Click Use Default Filter Condition or Do Not Set Filter Condition. You must set the default filter condition when you configure an associated table. Then, click Test to view the partition information.



**Note:**

When the associated schema is a MaxCompute partitioned table, you can specify the partition to be queried by the analysis API. If you do not specify a partition, all partitions will be queried. If a default filter condition has already been set for this table, the default filter condition is used by default when you perform a query.

4. View the results, SQL statements that are run, and procedure details.

### 5.3.4 Fast search

The Fast Search page allows you to set filter conditions and parameters to generate charts and data.

1. Choose Analysis APIs > Fast Search to open the corresponding page. In the right workspace, select a schema from the Select Schema drop-down list. On the left side of the workspace, the tag category of the selected schema is displayed. To search for tags, you can add filtering conditions by selecting and dragging common tags from the tag category.



**Note:**

Currently, fast search only supports the following four schemas: Analytic DB (ADS), ApsaraDB for RDS (RDS), Table Store (OTS), and Oracle.

2. After you drag a common tag to the Filter Conditions area, you can configure a filtering condition by selecting one of the following data types: enumeration, numeric, and date.
3. You can configure the filter conditions by dragging multiple common tags and specifying data types. For example, you have configured a filter to search the employees whose salary is greater than 20,000. The name, position number, department number, birth date, and salary are displayed in the returned table. Above the returned table, click Export to export and check the result in *Computer \My Downloads*.

The search results can be saved and published as APIs.

- a. Click Save as API. In the dialog box that appears, enter the API name, path, group, and description.
  - b. After you save and publish the API, you can click APIs to view the published APIs.
4. Configure chart parameters. For example, use employee ID as the x-axis and salary as the y-axis to generate a chart.
  5. Export a chart. Select different aggregate functions (such as count, min, and max functions) and chart types (such as bar chart), and click OK to display the chart. Then, you can click Export Chart to export the chart as an image.

## 5.4 Dashboards

### 5.4.1 Overview

The Dashboards page consists of three modules: datasets, report configurations, and report permissions.

The Dashboards page provides an intuitive data view and analysis reports. This offers business personnel a lot of useful information such as tag query results. After a report is generated, it can be published to authorized users for viewing. This provides an easy way for authorized users to view and share data analysis reports.

### 5.4.2 Manage datasets

Configure filters

1. Choose Dashboards > Datasets to open the corresponding page. On the Datasets page, you can view, test, unpublish, and delete existing datasets.

2. Click **Test** at the bottom. In the dialog box that appears, configure the filtering conditions and click **Test** to view the test results.
3. Enter TQL statements in the query editor. If the filtering conditions exist in the **Configure Filters** section, you need to configure the filters.
4. Click **Save and Test**. A message indicating that the dataset is saved appears. Click **Publish Online**.



**Note:**

**If you want to unpublish the dataset, click **Move Offline**.**

Create a group

**Choose **Dashboards > Datasets** to open the corresponding page. Click the new folder icon to create a group**

Create a dataset

**Choose **Dashboards > Datasets** to open the corresponding page. Click the new file button, enter the dataset name and description, select the dataset location, and click **OK**.**

## 5.4.3 Manage reports

Configure a report

**Choose **Dashboards > Report Configurations** to open the corresponding page. On the page, you can not only create a group and report but also edit or unpublish an existing report.**

- When a report is saved but not published, move the pointer over the report and the **Edit** button appears. You can click **Edit** to open the editing page and modify the report.
- When a report has been published, move the pointer over the report and the **View** button appears. You can click **View** to open the report, and click **Modify** to unpublish the report and modify it offline.

**To modify the basic information and style, click **Edit**.**

**The style information includes global settings, title settings, ticker board settings, and interaction settings.**

- **Global Settings:** Allows you to adjust text style settings, including font, alignment, and spacing.
- **Title settings:** Allows you to modify title name and title style settings, including font size, color, and font weight.
- **Ticker board settings:** Allows you to modify font, alignment, prefix, text style, numeric style, and suffix settings.
- **Interaction settings:** Allows you to set the callback ID.

#### Create a report

1. Choose **Dashboards > Report Configurations** to open the corresponding page. In the top of central pane, click the new folder icon to create a report group.
2. Enter the group, and click **Create Report**.
3. On the report, click **Edit** to open the editing page. Click **Add Visual**. Select a dataset and visual type, enter a visual name, and configure data used in the visual.
4. After the visual is created, click **Add** and publish it.

### 5.4.4 Report permissions

#### Manage roles

1. Choose **Dashboards > Report Permissions** to open the corresponding page. On the **Role Management** page, click **Create Role**. Enter a role name, description, select visible report groups and members, and click **Create**.

After the role is created, you can view the role on the **Role Management** page.

2. On the **Role Management** page, click **Edit** in the **Actions** column of a role to modify the role.
3. On the **Role Management** page, click **Delete** in the **Actions** column of a role to delete the role.

#### Authorize users

1. Choose **Dashboards > Report Permissions** to open the corresponding page. Select **User Authorization** to view the list of authorized users.
2. Click **Edit Role** in the **Actions** column of a user ID. In the **Role Management** dialog box that appears, you can modify the roles corresponding to the user ID.

## 5.5 Tag factory

### 5.5.1 Overview

The tag factory module provides the business tag processing functions, including the tag schemes and tag tasks.

On the Tag Schemes page, you can create tag schemes by configuring TQL and algorithms. On the Tag Tasks page, you can run the tag tasks that are generated by the configured tag schemes.

### 5.5.2 Tag schemes

#### 5.5.2.1 Overview

A tag scheme is used to define the logic of derived tags, including the type, tag configuration, scheduling configuration, and parameter configuration of the tag scheme.

When creating derived tags, you must define the tag generation logic, the algorithms based on existing tags, the result fields corresponding to the new tags, and tag objects to be associated. If the result is a MaxCompute partitioned table, you must also configure which output field is used as the partitioning field.

The task includes the following two scheduling types: one-time schedule and recurring schedule. The tag generation logic supports TQL statements, common functions, and logical expressions.

#### 5.5.2.2 Create tag schemes

You can create a tag scheme of the following two types: TQL-based and algorithm-based.

Click Create Tag Scheme, select a scheme type, and enter the scheme name and description.

Create a TQL-base tag scheme

1. Select TQL as the scheme type to open the creation page. In the upper-left corner, select a source schema and destination schema. In the query editor, enter TQL statements, specify an attached object or link, and set returned fields and variables.



**Note:**



**You can only select MaxCompute schemas on the Select Schema page.**

If returned fields exist in the right-side pane, click the text box in the Tag Name column. In the drop-down list, select an existing tag and attach the tag to the field as a primary tag or common tag. If no tags exist, click Create Tag to create one.

2. In the Create Tag dialog box that appears, enter the required information, specify whether the tag is a primary tag, and then click OK.
3. On the tag configuration page, check whether the tag is created.

Click Variable Settings in the right-side pane to expand the variables settings. If variables exist, their data types and default values are the same as those of the variables in the TQL statements.

You can click Test to view returned results in table or JSON format, SQL statements that are run, and procedure details.

4. Click Next in the upper-right corner to open the Offline Task page. Select a schedule mode to configure the scheduling policy.
5. Preview and save the tag scheme. Click Next in the upper-right corner to open the Preview and Save page. Check the configurations and click Save.

Create an algorithm-based tag scheme

1. Select an algorithm.

Select Algorithm as the scheme type to open the Algorithms page. Select an algorithm and click Next in the upper-right corner.

2. Configure algorithm inputs.

On the Algorithm Input Table page that appears, select a schema, enter the TQL scripts of algorithm inputs, and set variables. In the upper-right corner, click Next.

3. Configure algorithm outputs.

On the Algorithm Outputs page that appears, configure the entity that is attached to an output table and a primary tag or common tags attached to returned fields. Click Next.



**Note:**

**You can only select a MaxCompute schema when configuring a tag scheme by TQL statements.**

If returned fields exist in the right-side pane, click the text box in the Tag Name column. In the drop-down list, select an existing tag and attach the tag to the field as a primary tag or common tag. If no tags exist, click Create Tag to create one.



**Note:**

If the primary tag of an existing object has been attached to a field, all new tags are attached to fields as common tags. You must specify values for the fields marked with an asterisk (\*).

4. After completing the configuration, click OK.

5. Configure algorithm parameters.

On the Configure Algorithm page that appears, configure the parameters for the algorithm, and click Next in the upper-right corner.

6. Configure a scheduling policy for offline tasks.

Select a schedule mode to configure the scheduling policy for offline tasks. Click Next and then click Save.

7. Preview and save the tag scheme.

Check the configurations and click Save.

### 5.5.2.3 Submit tag schemes

On the Tag Schemes page, you can change, submit, and delete tag schemes.

To submit a tag scheme, click Submit in the Actions column. For a TQL-based tag scheme, the system will create a MaxCompute table and an entity, and also configure associated tables and tags for the tag scheme. For a tag scheme that is configured by algorithms, the system will create a MaxCompute table, an algorithm workflow, and an entity. It also configures associated tables and tags for the tag scheme.

After the tag scheme is submitted, the Submit button changes to Execute and the status of the tag scheme is updated to Submitted.

After a tag scheme that is configured by text TQL or an algorithm is submitted, the status of the tag scheme is updated to Submitted and the View Details, Execute, Tasks, Delete, and Clone buttons appear in the Actions column.

#### 5.5.2.4 Run tag schemes

On the Tag Schemes page, you can click **Execute** in the Actions column to run a submitted tag scheme.

Each time you run a tag scheme, the **Configure Parameters for Task Execution** dialog box appears. It displays the execution parameters. The parameters usually have the default values that are set during the creation of the tag scheme. You can change the default values according to specific business scenarios. The execution parameters of text TQL are displayed in the same way because they are both TQL parameters. The execution parameters of algorithms include execution parameters of both TQL and algorithms.



**Note:**

If the parameter has a default value, you can change it or not, but you must ensure that it has a value.

Click **Execute** in the Actions column of a tag scheme. In the **Configure Parameters for Task Execution** dialog box that appears, configure the default values of the execution parameters. You can change the default value, and click **OK**.

- Run text TQL-based tag schemes
- Run algorithm-based tag schemes

#### Recurring tasks

1. If the task is a recurring task, click **Deploy**.
2. In the **Configure Parameters for Task Execution** dialog box that appears, click **OK**.
3. On the Tag Tasks page, a task appears and its status is shown as **Scheduling**.

#### One-time tasks

1. If the task is a one-time task, click **Execute**.
2. In the **Configure Parameters for Task Execution** dialog box that appears, click **OK**.
3. On the Tag Tasks page, a task appears and its status is shown as **Running**.

### 5.5.3 Tag tasks

Currently, you can generate tag tasks by configuring tag schemes with the following two methods: write TQL statements or configure algorithms.

On the Tag Schemes page, click Create Tag Scheme. In the dialog box that appears, you can select a scheme type. You can change, submit, run, and delete tag schemes by clicking the corresponding button in the Actions column.



**Note:**

- You can change a tag scheme again only when the creation or submission failed.
- You can run a tag scheme only after it is submitted.
- After the tag scheme is run, you can view the corresponding tasks on the Tag Tasks page.

The Tag Tasks page displays the operator, start time, end time, and status of a task. You can view the execution parameters and logs. You can also search for tasks by submission date or task type.

After the tag scheme is configured, you need to run it to generate a derived tag.

On the Tag Tasks page, you can manage the tasks generated by task schemes and schedule these tasks. You can run a tag scheme after it is modified and submitted.

A task will be generated each time when you click Execute to run a tag scheme. You can set execution parameters for a recurring task. After the task starts scheduling, you can view the task instances, running status, and logs for each scheduled task.

## 5.6 Tag sync

### 5.6.1 Overview

Tag sync is one of the most important functions of processing cross-computing data flow for DataQ - Smart Tag Service. It includes sync schedules, sync tasks and task O&M.

When data is required by the corresponding data service, the tag center can collect the data distributed across multiple storage systems. The data is then subscribed to the location where the data service needs to compute.

## 5.6.2 Sync schedules



**Note:**

When you plan to synchronize data from a source schema to a target schema, you need to select a scheduling mode.

1. Choose Tag Sync > Sync Schedules to view all the schedules in the form of a list.
2. On the sync schedule list, choose More > Details in the Actions column of a sync schedule to view the sync schedule configuration.
3. In the upper-right corner of the Schedules page, click Create to create a new sync schedule.

The following section describes how to synchronize data from other schemas to MaxCompute schemas, and synchronize data from MaxCompute schemas to other schemas.

Synchronize data from another schema to a MaxCompute schema

1. Configure a sync schedule.
2. Configure sync tags.
3. Configure sync parameters. You must specify a value of retention period.



**Note:**

When tags of another schema are synchronized to a MaxCompute schema, you must turn on the Partitioned Table switch and configure a partitioning column name and value. The partition settings are as follows:

- Partition name: user-defined.
- Partition value: user-defined. You can also specify a partition expression of the DateTime data type.

4. Click Next to preview the configuration and then click Save.

Synchronize data from a MaxCompute schema to another schema

1. Configure a sync schedule.



**Note:**

When synchronizing data from another schema to an AnalyticDB schema, you must synchronize the data to a standard table or dimension table.

- Standard table: A table that stores data in a database.

- **Dimension table:** A table that stores detailed information about specified attributes of a standard table.

2. Configure sync tags.

3. Configure sync parameters.



**Note:**

When synchronizing data from a MaxCompute schema to another schema, you must turn on the Partitioned Table switch and configure a partitioning column name and value.

When setting the partition value, you need to query the partitioning column value of the MaxCompute database where the source data is located.

4. Click Next to preview the configuration and then click Save.

Start a task sync schedule

After creating a sync schedule, you can start the sync schedule to generate sync tasks by using any of the following two modes: Run Now and Recurring.

- **Run now:** The task schedule is immediately run after you click OK.
- **Recurring:** The task schedule is run at a specified date and time. You can specify a validity date and frequency. The frequency includes minute, hour, day, week and month. You need to select the required settings that correspond to the specified frequency.

### 5.6.3 Sync tasks

After a sync schedule is started, a sync task is generated. You can view the task status, scheduling mode, and operations.

Choose Tag Sync > Sync Tasks > Tasks to view all tasks displayed in the form of a list. You can view the status of a task or terminate a running task.

### 5.6.4 Task O&M

On the Task O&M page, you can view the detailed log of the sync task instance.

Choose Tag Sync > Task O&M > Task Instances to open the corresponding page.

Above the list of the task instances, click the blank box next to Select Schedule and select a sync schedule in the drop-down list. Then, you can view the instance status and logs.



**Note:**

**You can only view instance logs when the instance is in the Succeeded status.**

## 5.7 Other features

### 5.7.1 Homepage

The homepage displays the BI reports for smart tag analysis, showing the smart data assets for you from the business perspective.

### 5.7.2 Tag center

#### 5.7.2.1 Overview

The tag center is a cross-computing storage that supports logical and dynamic modeling based on physical models (object-link-tags model). It integrates with data services to provide data modeling and data management tools for big data application and development.

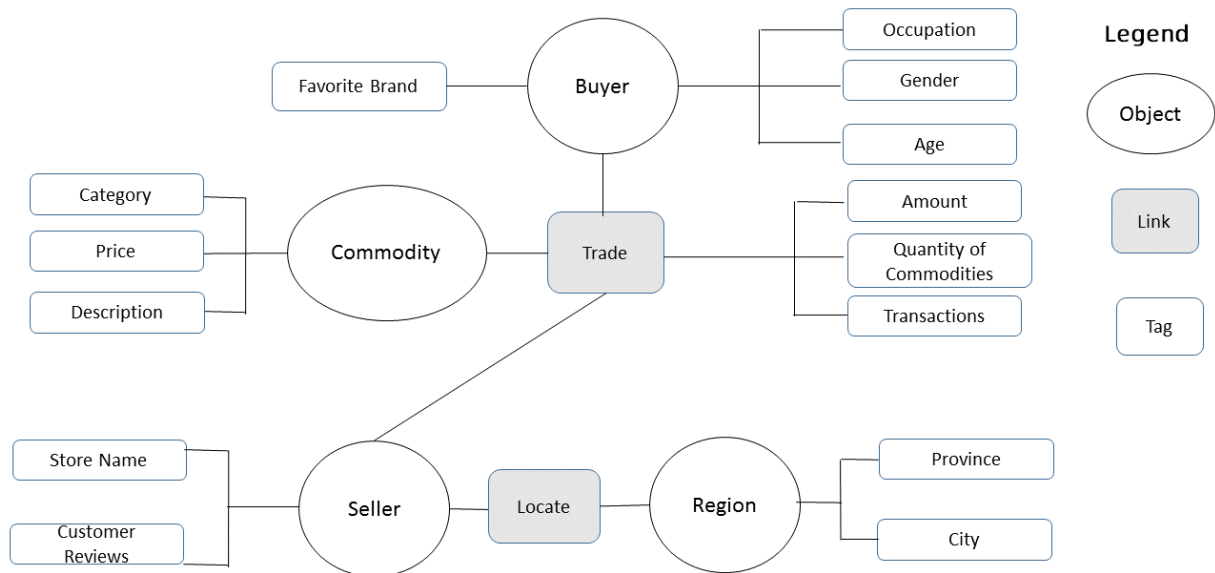
The data model view of an enterprise can be presented through visual methods. It is easy for business personnel, developers, and database administrators to gain a deeper insight into enterprise data assets.

The tag center is used to build a logic model across computing and storage resources based on existing data tables. This allows you to manage, process, and query data at the tag model layer without interacting with underlying big data computing and storage resources. The tag center is more important when the data architecture is complex and the combination of multiple computing and storage resources is required.

The tag modeling method is widely used in precision marketing, personalized recommendation, user profiling, credit scoring, and other big data applications based on detailed data computing. A tag is the minimum unit of description for a user object and represents an abstract expression of a specific objective fact for an object. The abstract expression (such as attributes, behaviors, and interests) is a data modeling method from the business perspective. For example, attributes include gender (the tag value is male or female) and age (the tag value is the actual age). Behaviors include turnover, bookmarks, and location. Interests include preference for multiple keywords. A tag can be a column consisting of values,

enumerated values, and multiple key values, or a fact table consisting of multiple fields (subjects, predicates, objects and time).

In terms of conceptual model, the tag system is a tag-based description methodology. This is built around multiple objects (buyer, seller, commodity, enterprise, and equipment) and the links between objects (transactions).



In traditional modeling, the concept and logic model are first designed according to business needs, and then the physical data tables are processed and sorted based on the logic model. In tag modeling, the logic model is directly built based on existing physical data or models. With the parsing of different data service agents, you can perform various computations on the model view without preprocessing a large amount of physical data.

Tags are created based on the data of physical tables. In a cross-computing context, you may experience differences in query languages and performance between multiple computations. Therefore, tags created on logical requests may not be computed. In general, each tag that you have defined still needs to be associated with the corresponding physical table. However, in Smart Tag Service, you can define a computing logic of a query as a temporary tag in the corresponding data service. When computing logic is related to cross-computing, it needs to be converted into data for physical tables to avoid errors.



### 5.7.2.2 The overview chart

The overview chart provides a graphical way for you to view all objects, relationships, and attributes between these objects. You can also view the tags attached to objects in a two-dimensional manner. You can view and analyze the entire tag model through the overview chart.

The overview chart of the Tag Center module displays the various models you have created and the relationships between links and objects in the model. In the upper-left corner of the overview chart, you can enter a keyword in the search box to quickly search an entity.

To view the details of an entity (an object or a link), you can click the object or link. Alternatively, you can also right-click the object or link and select Object Details. In the right-side pane that appears, it displays three tabs: Details, Tags, and Associated Tables. The Details tab page displays basic information about the entity. You can enter a description in the Description text box. The Tags tab page displays the tags that are attached to the entity. You can create a new tag to attach the entity. The Associated Tables tab page displays the associated tables.

### 5.7.2.3 Tag warehouse

Tag warehouse stores shared tags. You can view and apply for shared tags in tag warehouse.

The tag warehouse has the following functions:

- **View shared tags:** You can view shared tags by workspace, filter tags by tag category, and search for shared tags by keyword.
- **Apply for tags:** You can select the required tags and apply for permissions to use the tags. You can apply for multiple tags at a time, and check the application status in approval process.

Display shared tags

Choose Tag Center > Tag Warehouse to open the corresponding page. Select a tag category in the Tag Categories section to filter tags in different categories based on the selected workspace. On the tag list, you can also click on a tag to display details of the tag.

Apply for using shared tags

**On the tag list, click Apply in the Actions column of a tag. In the dialog box that appears, enter a reason and click OK. If you want to cancel the operation, click Cancel.**

**On the tag list, select the check boxes next to the Tag Code column, and click Apply for Tags. In the dialog box that appears, enter a reason and click OK. If you want to cancel the operation, click Cancel.**

### 5.7.2.4 My tags

**The My Tags page displays private tags, claimed tags, and shared tags.**

**As a department member, you can view, search, modify, and share private tags. You can also perform fast search, share multiple tags, revoke tag sharing, and detach private tags.**

**Shared tags are the tags shared by my workspace and sub workspaces to the tag warehouse. These tags are also authorized tags for other users to use. If you want to use the shared tags in the tag warehouse, you need to click Apply to submit an application and the workspace administrator needs to approve the application.**

Display tags

**Choose Tag Center > My Tags to open the corresponding page. You can select a category to filter tags.**

**You can select a category to filter tags in the Tag Categories section. You can also click Edit Categories in the upper-right corner of the section to modify the categories.**

**The tag list is classified into three tabs: Private tags, Claimed tags, and Shared Tags. In the upper-right of the list, you can search for tags by tag name or tag code.**

- **Private tags**

**On the Private Tags tab, you can share one or more tags, revoke tag sharing, and detach one or more tags from a table.**

- **Claimed tags**

**The Claimed Tags tab displays the approved tags that you have applied to use in the tag warehouse. These tags are displayed by submission time.**

- **Shared tags**

**The Shared Tags tab displays the tags that are shared on the Private Tags page.**

**You can set the tag sharing policy to configure the visible range of shared tags.**

#### **Edit categories**

**To add, edit, and delete tag categories, click Edit Categories in the upper-right corner of the Tag Categories section.**

- **Select a category and click the Create icon. In the Create Category dialog box that appears, enter or select the required information, and click OK. If you want to cancel the operation, click Cancel.**
- **Select a category and click the Edit icon. In the Edit Category dialog box that appears, enter or select the required information, and click OK. If you want to cancel the operation, click Cancel.**
- **To delete a category, select the category and click the Delete icon. If you want to cancel the operation, click No in the message box that appears.**

#### **Perform fast search on tags**

**Choose My Tags > Private Tags to open the corresponding page. Select one or more tags and click Fast Search to check the entities that attach with these tags.**

**In the dialog box that appears, if you do not need to perform fast search on one or more tags, you can click on these tags in the Tags column. These tags are displayed with a grey background and not listed in the tag category on the Fast Search page. After you select the tags that you want to perform fast search, click OK.**

**To view the data and results, choose Analysis APIs > Fast Search to open the corresponding page.**

### **5.7.2.5 Tag models**

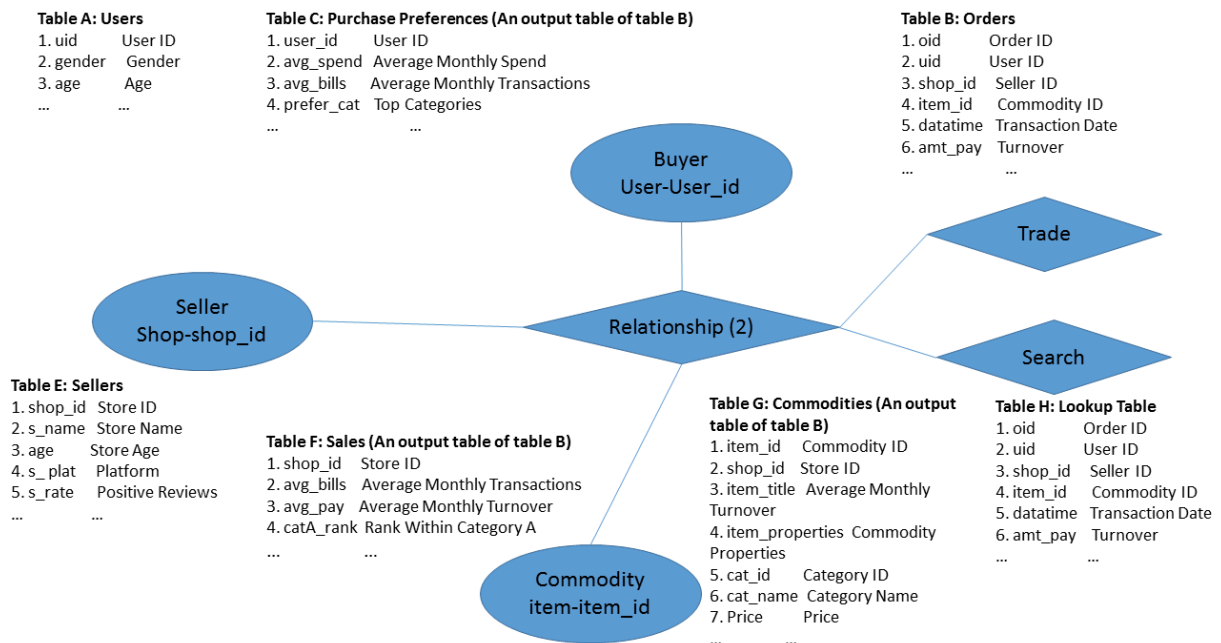
**Tag modeling is a network-based modeling method that is based on the three elements of an OLT model: object, link, and tag. This method is used to build tag models for data distributed across different databases.**

**An object is used to describe a real object such as device, personnel, and address , which corresponds to physical data tables (usually property tables). In this table , the primary key represents the object and other columns are tags (namely, the properties of the described object).**

Links are relationships, events, and actions between different objects. They correspond to physical tables, which are usually fact tables. For example, a deal, repair, and ride.

A tag is attached to an object or link to describe attributes. For example, the device production time and the device usage frequency are tags of a device. The repair time and number of repairs are the tags of a repair.

Compared with the metric-dimension system, this modeling method is more suitable for the description and expression of detailed data. Detailed data mainly consists of a fact table. The concept of links corresponding to the fact table is introduced to show a clear representation of the relationships between multiple objects. This concept is conducive to management and expression during analysis. On the business side, it is similar to the conceptual model design and easier to understand.



After modeling, you can convert the model in the preceding table to the logical relationship that are shown in the preceding figure. Transaction tables are mapped to the links, while the amount and time are the tags of the links. The user tables and commodity tables are mapped to buyer and commodity respectively, while gender and age are the tags of the buyer.

## Manage object-link model or categories

**Object-link model management is the primary function of tag center to configure the logic model. It can read meta information from different database sources and integrate the meta information as an entity or link. Multiple tables describing the same object (primary key) can be accumulated into a large wide table at the logic layer. The composite primary key tables can be considered as links during creation and used to associate multiple objects. Other descriptive fields are defined as tags based on your requirements.**

**Choose Tag Center > Tag Models to open the corresponding page. On the page, you can create, edit, and delete categories.**

- **Create a category**

- 1. Choose Tag Center > Tag Model to open the corresponding page.**
- 2. Above the list of the tag categories, click the Create icon.**
- 3. In the Create dialog box that appears, select Category as the type, enter a category name and description, and select a location. You can select the root category or an existing sub category under the root directory.**
- 4. Click OK and you can view the create category in the list of tag categories.**

- **Edit a category**

**To edit the root category or a sub category, select the root category or sub category and click the Edit icon.**

- **Delete a category**

**To edit the root category or a sub category, select the root category or sub category and click the Delete icon.**

## View tag models

**Choose Tag Center > Tag Models to open the corresponding page. On the page, you can view the information of tag models, including model categories, objects, and links.**

**You can select a tag model in model category list, and click the Edit icon above the tag category list. In the dialog box that appears, modify the model name, description, and location.**

**In the Actions column of an object or link, you can click Manage Tables, Details , Edit, and Delete to perform the corresponding operations.**

- **Manage Tables:** Allows you to view schemas, table names, association modes, the primary table switch, number of tags, and operations.
- **Details:** Allows you to view the basic information, data assets, and tags of an object or link.
  - **Basic Information:** Basic information includes the object or link name, the person who create the object or link, and the workspace to which the object or link belongs.
  - **Data Assets:** This section displays the total data volume of physical tables associated with the entity, and the number of data records and table names of each table associated with the entity.



**Note:**

Data assets only support statistics on objects or links of associated physical tables in ApsaraDB for RDS and AnalyticDB schemas. Statistics on objects or links of associated physical tables in a MaxCompute schema is not currently supported.

- **Tags:** the list of tag categories.
- **Edit entities**
- **Delete entities**

**You can search for objects or links that are in the current category and all categories.**

Create a value mapping table

- 1. Choose Tag Center > Tag Models to open the corresponding page. On the page, click Details in the Actions column of an object or link.**
- 2. On the Details page, click Create.**
- 3. In the Create Tags dialog box that appears, enter a tag code and tag name, select a category, a data type, and a value type, and enter a tag description based on the displayed information. Then, click OK.**
- 4. Select a value mapping table and click Edit Value Mapping Table.**

5. In the Edit Value Mapping Table dialog box, enter or select the required information and click OK.

In the Edit Value Mapping Table dialog box, click Get from Table, the dialog box switches to the automatic creation mode. Enter or select the required information, and click OK.

6. Select a tag and click Edit in the Actions column of the tag. In the Edit Tag dialog box, enter or select the required information and click OK.

To delete a tag, click Delete in the Actions column of the tag.

### 5.7.2.6 Model views

Choose Tag Center > Model Views to open the corresponding page. On the page, you can create a category or canvas by clicking the Create Category icon.

### 5.7.2.7 Schemas

Schema management supports communications between multiple computing and storage resources to obtain meta information.

Currently, DataQ - Smart Tag Service allows you to manage the following computing and storage resources.

- ApsaraDB for RDS
- MaxCompute
- AnalyticDB
- Table Store
- DataHub
- Realtime Compute

Choose Tag Center > Schemas to open the corresponding page. On the Schemas page, you can view the added schemas. You can also view data tables, and edit and delete a schema.

- View tables

Click View Tables in the Actions column of a schema to go to the Virtual Tables page.

On the Virtual Tables page, you can view virtual table list and delete virtual tables. In the upper-right corner, click Create Virtual Table. In the dialog box

that appears, enter a table name and description, specify a table definition, and click OK.

- **Edit a schema**

To edit a schema, click **Edit** in the **Actions** column of the schema on the **Schemas** page.

- **Delete a schema**

To delete a schema, click **Delete** in the **Actions** column of the schema on the **Schemas** page.

#### Add a schema

Before establishing big data applications through DataQ - Smart Tag Service, you must first authorize the relevant schemas to obtain the meta information to build data model views.

Currently, DataQ - Smart Tag Service supports the following cloud computing resources as the schemas:

- **ApsaraDB for RDS**
- **MaxCompute**
- **AnalyticDB**
- **Table Store**
- **DataHub**
- **Realtime Compute**
- **Galaxy**
- **Oracle**

### 5.7.2.8 Data import

You can import task files into schemas (only MaxCompute schema is currently supported) to create tags.

Log on to the DataQ - Smart Tag Service, choose **Tag Center > Data Import** to view the import tasks. Currently, only files in the **TXT** and **CSV** formats can be imported.

#### Import CSV files

1. In the upper-right corner, click **Import Task**, and select the file to import.



2. At the bottom of the dialog box, select **Import Unassociated Tags** or **Import Associated Tags** according to your business needs.

- **Import unassociated tags**

Select the schema where the file is to be imported. Generally, the table name is generated by default, you can modify the name. In the center section, select a data type in the **Target Field Type** column of each row, and click **Import**.

On the **Import Tasks** page, you can view the imported tasks. An imported table can be associated with a tag of an object or link.

- **Import associated tags**

Select an entity such as an object, associated table field, and table field type.

Enter a tag name, code, and category, and select the schema where the file is to be imported. You can change the destination table. Then, click **Import**.

After the import is complete, you can view the imported tables in the import task list.

3. Next, choose **Analysis APIs > API Factory** to open the corresponding page. On the page, you can enter code and run the statements in the query editor. Then, verify that the Boolean value of the tag is set to true in the result.

### 5.7.3 Tag Apps

In the **DataQ - Smart Tag Service** console, move the pointer over **Tag Apps**, and select **Profile Analysis** or **Data Exploration** from the shortcut menu to open the corresponding page by one click.

### 5.7.4 Manage workspaces

Log on to the **DataQ - Smart Tag Service** console. Click **System Settings** in the upper-right corner and select **Workspaces** from the shortcut menu. On the **Workspaces** page, you can view the list of workspaces, change members in the workspace, edit workspace information, and delete workspaces.

#### Role permissions

The following table lists the permissions of each role.

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
Workspace permissions	ws-read	The permission to access a workspace.	√	√	√	√	√
	ws-update	The permission to update a workspace.	√	√	-	-	-
	-	The permission to create a sub workspace.	√	-	-	-	-
	-	The permission to delete a workspace.	√	-	-	-	-
	-	The permission to edit a workspace.	√	-	-	-	-
Role permissions	role-read	The permission to view a role.	√	√	√	√	√

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	role-grant	The permission to authorize user role.	√	√	-	-	-
Data permissions	-	The permission to view data.	√	√	√	√	√
	-	The permission to update data.	√	-	-	-	-
	-	The permission to create data.	√	-	-	-	-
	-	The permission to delete data.	√	-	-	-	-
Schema permissions	schema-read	The permission to view a schema.	√	√	√	√	√
	schema-create	The permission to create a schema.	√	√	√	-	-

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	schema-update	The permission to update a schema.	√	√	√	-	-
	schema-delete	The permission to delete a schema.	√	√	√	-	-
Links permissions	-	The permission to view a link.	√	√	√	√	√
	-	The permission to create a link.	√	√	√	-	-
	-	The permission to update a link.	√	√	√	-	-
	-	The permission to delete a link.	√	√	√	-	-
Object permissions	entity-read	The permission to view an object.	√	√	√	√	√

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	entity-create	The permission to create an object.	√	√	√	√	-
	entity-update	The permission to update an object.	√	√	√	-	-
	entity-delete	The permission to delete an object.	√	√	√	-	-
Tag category permissions	ct-read	The permission to view a tag category.	√	√	√	√	√
	ct-create	The permission to create a category.	√	√	√	-	-
	ct-update	The permission to update a category.	√	√	√	-	-

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	ct-delete	The permission to delete a tag category.	√	√	√	-	-
Tag permissions	tag-read	The permission to view a tag.	√	√	√	√	√
	tag-create	The permission to create a tag.	√	√	√	√	-
	tag-update	The permission to update a tag.	√	√	√	-	-
	tag-delete	The permission to delete a tag.	√	√	√	-	-
	Tag-grant	The permission to approve the use of a tag.	√	√	√	-	-

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	-	The permission to associate an object, link, or tag to a table.	√	√	√	√	-
	-	The permission to share a private tag.	√	√	√	-	-
	-	The permission to revoke sharing for a tag.	√	√	√	-	-
	-	The permission to apply for the use of a tag shared by others.	-	√	√	√	-
	-	The permission to view an application.	-	√	√	√	-

Permission set	Permission	Description	System administrator	Workspace administrator	Workspace developer	Workspace analyst	Workspace visitor
	-	The permission to cancel an application.	-	√	√	√	-
	-	The permission to review an application.	-	√	-	-	-
	-	The permission to set a tag sharing policy.	√	√	√	-	-
	-	The permission to view a tag sharing policy.	√	√	√	√	√
Homepage permissions		The permission to view core data assets by configuring display items.	√	-	-	-	-



## 6 E-MapReduce (EMR)

---

### 6.1 What is EMR?

EMR is a managed cluster platform that simplifies running big data frameworks, such as Hadoop, Spark, Kafka, and Storm. EMR provides you with one-stop big data processing and analysis services, such as managing clusters, jobs, and data.

EMR is a service that is based on ZStack and uses open-source Apache Hadoop and Spark to process and analyze vast amounts of data. You can use components, such as Apache Hive, Apache Pig, and HBase, in the Hadoop and Spark ecosystems to process and analyze data. You can also use EMR to import and export data from Alibaba Cloud data stores and databases, such as OSS and ApsaraDB for RDS.

### 6.2 Introduction

#### 6.2.1 Prerequisites

Before using EMR services, you need to familiarize yourself with information, such as the hardware and software configuration of a cluster, and the procedure to deploy a service.

#### 6.2.2 Introduction

##### 6.2.2.1 Software configuration

You need to install an operating system and big data components on each ECS instance of an EMR cluster.

##### 6.2.2.2 Software environment

[Table 6-1: Software requirements](#) lists the software requirements.

Table 6-1: Software requirements

Software	Description
Operating system	CentOS 7 64-bit kernel-3.10.0-693.2.2.el7.x86_64
JDK	OpenJDK 1.8.0

### 6.2.2.3 Supported components

*Table 6-2: List of components* lists the components supported by EMR.

Table 6-2: List of components

Component	Version
Hadoop	2.7.2-emr-1.2.14
Hive	2.3.3
Tez	0.9.1
Spark	2.3.1
Oozie	4.2.0
Hue	4.1.0
Zeppelin	0.7.1
Sqoop	1.4.7
Knox	0.13.0
ZooKeeper	3.4.12
Ganglia	3.7.2
Pig	0.14.0
Kafka	2.11_1.0.1
HBase	1.1.1
Phoenix	4.10.0
Presto	0.188

### 6.2.2.4 Introduction to components

This section describes big data components that run on an EMR cluster.

- **Hadoop**

- **YARN**

YARN provides task scheduling and cluster resource management.

- **HDFS**

HDFS is a distributed file system.

- **Hive**

Hive is a Hadoop-based open-source data warehouse software that provides an SQL-like interface for data processing and analysis. Hive uses tables to store and manage data.

- **Spark**

Spark is a memory-based distributed computing framework that supports batch and real-time computing, SQL statements, and machine learning.

- **Oozie**

Oozie is a job scheduler that supports workflow orchestration by building a directed acyclic graph (DAG). Oozie supports multiple types of jobs.

- **Hue**

Hue is an open-source user interface for visualizing data. Hue supports multiple components, such as Hadoop, Hive, Oozie, and HBase.

- **Sqoop**

Sqoop is a tool designed for migrating data between HDFS and relational databases.

- **ZooKeeper**

ZooKeeper is an open-source and distributed service for coordinating applications. ZooKeeper is similar to Google Chubby and an important component of Hadoop and HBase. ZooKeeper is a centralized service that provides consistent services for distributed applications. These services include configuration maintenance, naming, distributed synchronization, and group services.

- **Kafka**

Kafka is a high-throughput messaging system that provides a variety of features, such as high-throughput, scalability, high reliability, and high-performance. Example applications of Kafka include real-time compute, log processing, and data aggregation.

- **HBase**

HBase is an open-source, distributed, and column-oriented data store. HBase is a component of the Apache Hadoop project. Different from typical relational databases, HBase is a data store that is designed to store unstructured data. HBase is a column-oriented rather than row-oriented data store.

- **Phoenix**

**Phoenix provides SQL-like statements that allow you to perform data analysis on HBase data.**

- **Presto**

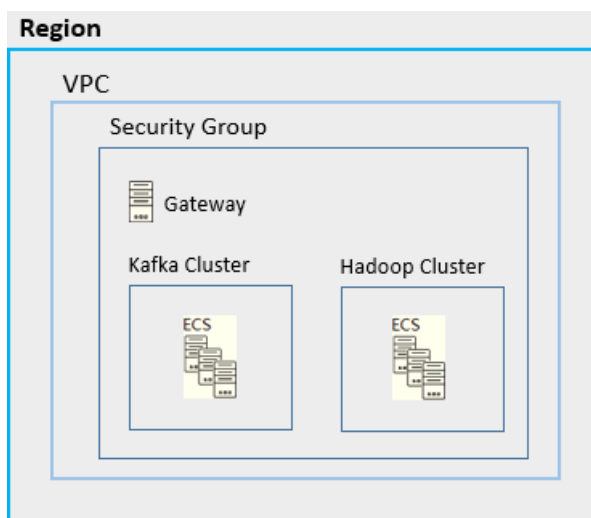
**Presto is a distributed SQL query engine for retrieving large datasets from one or more data sources.**

## 6.2.3 Introduction

### 6.2.3.1 Hardware architecture

**For more information about the hardware architecture of an EMR cluster, see [Figure 6-1: Hardware architecture](#).**

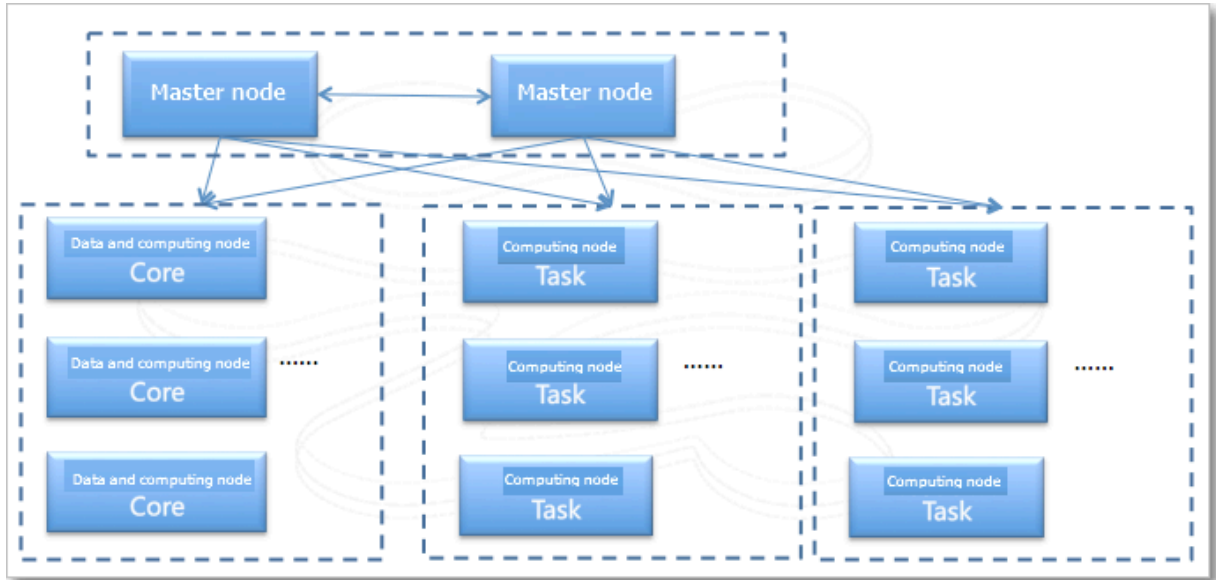
Figure 6-1: Hardware architecture



### 6.2.3.2 Cluster architecture

For more information about the architecture of a cluster, see [Figure 6-2: Architecture](#).

Figure 6-2: Architecture



### 6.2.3.3 Hardware requirements

EMR provides services based on ECS instances. All specifications of ECS instances are available for you to deploy a cluster node.

The minimum and recommended specifications of cluster nodes are listed in the following table.

Table 6-3: List of hardware components

Hardware	Description
CPU	<p><b>Minimum:</b> quad-core.</p> <p><b>Recommended:</b> 32-core.</p> <p>The standalone mode is supported.</p>
Memory	<p><b>Minimum:</b> 32 GB.</p> <p><b>Recommended:</b> equal to or greater than 64 GB.</p>

Hardware	Description
Disk	<p><b>Master node</b></p> <ul style="list-style-type: none"> <li>• <b>System disk:</b> 1 cloud disk (equal to or greater than 100 GB).</li> <li>• <b>Data disk:</b> 1 cloud disk (equal to or greater than 500 GB).</li> </ul> <p><b>Core node</b></p> <ul style="list-style-type: none"> <li>• <b>System disk:</b> 1 cloud disk (equal to or greater than 500 GB).</li> <li>• <b>Data disks:</b> <ul style="list-style-type: none"> <li>- You can choose local disks. The total number of local disks changes based on your requirements. We recommend that you use 4 to 12 disks.</li> <li>- Cloud disks: We recommend that you use 4 disks. The total number of cloud disks changes based on your requirements.</li> </ul> </li> </ul> <p><b>Task node</b></p> <ul style="list-style-type: none"> <li>• <b>System disk:</b> 1 cloud disk (equal to or greater than 100 GB).</li> <li>• <b>Data disks:</b> 4 cloud disks (equal to or greater than 500 GB). The number of disks and capacity of each disk change based on your requirements.</li> </ul>
Network	<p>EMR supports both classic networks and VPC networks.</p> <p>We recommend that you use VPC networks.</p>

## 6.2.4 Introduction

### 6.2.4.1 Deployment

This topic describes available deployment modes for a cluster and supported cluster services.

### 6.2.4.2 Deployment modes

This topic describes the available deployment modes for an EMR cluster.

EMR supports the following deployment modes:

- Hybrid

EMR supports full-cluster hybrid deployment mode, which means that all components can be deployed in one cluster. Each node in the cluster can provide more than one service.

- Independent

Only one service is deployed on each EMR cluster.

### 6.2.4.3 Supported services

This topic describes services that you can deploy on an EMR cluster.

For more information about services supported by EMR, see [Table 6-4: List of services](#).

Table 6-4: List of services

Service	Component	Deployment
Hadoop HDFS	NameNode	Deployed on a master node. In a high-availability (HA) cluster, NameNode is deployed on two master nodes.
	DataNode	Deployed on a core node.
	ZKFC	Deployed on a master node. In an HA cluster, ZKFC is deployed on two master nodes.

Service	Component	Deployment
	<b>JournalNode</b>	<ul style="list-style-type: none"> <li>• In a non-HA cluster , JournalNode is deployed on a master node, first core node, and second core node.</li> <li>• In an HA cluster , JournalNode is deployed on two master nodes and the first-created core node.</li> </ul>
	<b>KMS</b>	Deployed on a master node. Only supports single-node deployment.
	<b>HttpFS</b>	Deployed on a master node. Only supports single-node deployment.
<b>Hadoop YARN</b>	<b>ResourceManager</b>	Deployed on a master node. In an HA cluster, NameNode is deployed on two master nodes.
	<b>NodeManager</b>	Deployed on core nodes.
	<b>JobHistory</b>	Deployed on a master node. Only supports single-node deployment.
	<b>TimeLineServer</b>	Deployed on a master node. Only supports single-node deployment.
	<b>WebAppProxyServer</b>	Deployed on a master node. Only supports single-node deployment.
<b>Hive</b>	<b>HiveServer</b>	Deployed on a master node. In an HA cluster, HiveServer is deployed on two master nodes.
	<b>HiveMetaStore</b>	Deployed on the master node. In an HA cluster, HiveMetaStore is deployed on two master nodes.



Service	Component	Deployment
Spark	JobHistory	Deployed on a master node. Only supports single-node deployment.
Ganglia	GMond	Deployed on all nodes to collect information.
	GMetad	Deployed on a master node. Only supports single-node deployment.
HBase	HMaster	Deployed on a master node. In an HA cluster, HMaster is deployed on two master nodes.
	HRegionServer	Deployed on core nodes.
	ThriftServer	Deployed on a master node. Only supports single-node deployment.
ZooKeeper	ZooKeeper	<ul style="list-style-type: none"> <li>• In a non-HA cluster, JournalNode is deployed on a master node, first core node, and second core node.</li> <li>• In an HA cluster, JournalNode is deployed on two master nodes and the first core node.</li> </ul>
Hue	Hue	Deployed on a master node. In an HA cluster, Hue is deployed on two master nodes.
Oozie	Oozie	Deployed on a master node. In an HA cluster, Oozie is deployed on two master nodes.
HAS	HASServer	Deployed on the master node. In an HA cluster, HASServer is deployed on two master nodes.

Service	Component	Deployment
Knox	Knox	Deployed on the master node. In an HA cluster, Knox is deployed on two master nodes.

## 6.3 Introduction

### 6.3.1 User operations

This topic describes how to use E-MapReduce (EMR) services.

You can use the following methods to access cluster services.

- Access cluster services by using a gateway and submit compute jobs to a cluster. You can also install specific standalone applications on a gateway to manage jobs
- 
- Access cluster services from other locations, such as independent application services.

### 6.3.2 Create a RAM role

Before using E-MapReduce, you must create a RAM role to authorize E-MapReduce to access your cloud resources.

For more information about how to create a RAM role, see RAM roles in the *Alibaba Cloud Apsara Stack Console*.

### 6.3.3 Log on to the E-MapReduce console


This topic describes how to log on to the E-MapReduce console.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

#### Procedure

1. Open your browser.

2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/`manage, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username super. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. In the menu bar, choose  > Big Data > E-MapReduce.
6. On the E-MapReduce page, after a prompt appears, click Authorize to assign a default role to the account.



**Note:**

If you log on using an authorized department, skip this step.

7. Select your region and department from the drop-down lists respectively and click EMR to go to the EMR console.

## 6.3.4 Gateway

### 6.3.4.1 Gateway

This topic describes the role of a gateway.

An EMR gateway supports the following clients:

- Hadoop
- Spark
- Hive
- Oozie
- ApsaraDB RDS for HBase

The EMR gateway also supports Kerberos authentication for cluster access control.

The EMR gateway stores the latest host IP information for all Hadoop cluster hosts in the `/etc/hosts` path for you to copy for other usages.

### 6.3.4.2 Log on to an EMR gateway

Each developer has an independent gateway with advanced configuration.

To obtain the access address and password, contact the administrator.

To log on to an EMR gateway, use the `ssh root@gateway_ip` command.

The gateway assigns you a root account for managing nodes and a Hadoop cluster authenticated account for performing operations such as submitting jobs.

You can use the `su user_account` command to switch to the authenticated account.



#### Note:

`user_account` represents the authenticated account assigned to you.

### 6.3.4.3 Service environment

The following services have already been installed on the gateway.

The paths on which services are installed and configured are shown in the following table.

Table 6-5: Service environment

Service	Description
HDFS	<p>The default HDFS access domain is prefixed with <code>hdfs://emr-cluster</code>, where <code>emr-cluster</code> represents the cluster name.</p> <p>Installation path: <code>/user/&lt;username&gt;</code>. For example, if your username is <code>ali</code>, the default HDFS user path is <code>/user/ali</code>.</p> <p>The path where Hive data is stored: <code>/user/hive/warehouse/</code>.</p>

Service	Description
MapReduce	Installation path: <code>/usr/lib/hadoop-current</code> Configuration path: <code>/etc/ecm/hadoop-conf</code>
Spark	Installation path: <code>/usr/lib/spark-current</code> Configuration path: <code>/etc/ecm/spark-conf</code>
Hive	Installation path: <code>/usr/lib/hive-current</code> Configuration path: <code>/etc/ecm/hive-conf</code>
Oozie	Installation path: <code>/usr/lib/oozie-current</code> Configuration path: <code>/etc/ecm/oozie-conf</code>
HBase	Installation path: <code>/usr/lib/hbase-current</code> Configuration path: <code>/etc/ecm/hbase-conf</code>
Phoenix	Installation path: <code>/usr/lib/phoenix-current</code> Configuration path: <code>/etc/ecm/phoenix-conf</code>

**Note:**

- All installation paths are included in the Path file.
- The Hosts file on the gateway has already been configured.
- The cluster requires an update for the Hosts file when the nodes in the cluster change. However, the administrator will update the file for you.

#### 6.3.4.4 Security authentication

By default, Kerberos authentication is enabled for all EMR clusters. To obtain the Kerberos username and password, contact the cluster administrator.

The following example shows how to view authentication information for user ali and how to extend the authentication expiry date.

- Username: ali
- Password: ali123
- principal: aliyun@EMR.xxxxxx.COM

You can use the `klist` command to view the authentication information on the gateway.

For example:

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: xxx@EMR.xxxx.COM
Valid starting Expires Service principal
10/19/2017 10:49:16 07/03/2023 18:49:16 krbtgt/EMR.xxxxx.COM@EMR.xxxxx
.COM
renew until 10/21/2017 10:49:16
```

According to the output, the local gateway remains valid until 2023.

You can use the `kinit` command to perform authentication. The following command is an example of keeping Kerberos authentication valid on the gateway for a lifetime.

```
kinit -l 1000h
```

Obtain the Keytab file

The gateway provides a Keytab file that never expires. The Keytab file is stored in the root user path `/home/${user}/${user}.keytab`.

For example, if your username is `ali`, the Keytab file is stored in the `/home/ali/ali.keytab` path.

## 6.3.5 Jobs

### 6.3.5.1 Jobs

This topic introduces commands used to submit different types of jobs.

### 6.3.5.2 Hadoop MapReduce Job Configuration

EMR clusters support running Hadoop MapReduce jobs.

Generally, you can use a `hadoop` command to submit a Hadoop MapReduce job.

```
hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.7.2.jar pi 10 10
```

### 6.3.5.3 Submit a Spark job

EMR clusters support running Spark jobs.

The commands to run different types of Spark jobs are shown as follows.

- **Spark-Core**

```
spark-submit --class org.apache.spark.examples.SparkPi --master yarn
-client --driver-memory 512m --num-executors 1 --executor-memory 1g
--executor-cores 2 /usr/lib/spark-current/lib/spark-examples-1.6.3-
hadoop2.7.2.jar 10
```

- **Spark-SQL**

```
spark-sql -e "select * from demo"
```

- **Spark-Streaming**

**Same to the command to run a Spark core job.**

```
spark-submit --class org.apache.spark.examples.SparkPi --master yarn
-client --driver-memory 512m --num-executors 1 --executor-memory 1g
--executor-cores 2 /usr/lib/spark-current/lib/spark-examples-1.6.3-
hadoop2.7.2.jar 10
```

- **Spark-shell**

```
spark-shell
```

### 6.3.5.4 Submit a Hive job

**EMR clusters support running Hive jobs.**

**Procedure:**

1. **Log on to the Hive console.**

**Enter the hive command to log on to the Hive console.**

2. **Enter the following commands.**

```
hive -e "select * from ali.xxx"
hive -f example.hql
```

### 6.3.5.5 Oozie

#### 6.3.5.5.1 Oozie

**You can use Oozie to schedule MR, Spark, and Hive jobs.**

**The HDFS domain and jobTracker settings for these jobs are as following:**

```
nameNode=hdfs://emr-cluster
```

```
jobTracker=emr-header-1.cluster-xxxxx:8032
```

**xxxxx represents your cluster ID.**

### 6.3.5.5.2 Schedule a Hadoop MapReduce job

**This topic introduces how to use Oozie to schedule a MapReduce job.**

job.properties

```
nameNode=hdfs://emr-cluster
jobTracker=emr-header-1.cluster-xxxxx:8032
```

**xxxxx represents your cluster ID.**

workflow.xml

**Define a map-reduce action to run MR jobs.**

**Demo path:** `/home/${user}/examples/apps/map-reduce/`

### 6.3.5.5.3 Schedule a Spark job

**Using a Spark action to run Spark jobs causes a Kerberos authentication issue. To resolve this issue, use a Shell action to run Spark jobs.**

job.properties

```
nameNode=hdfs://emr-cluster
jobTracker=emr-header-1.cluster-xxxxx:8032
oozie.use.system.libpath=true
```

**xxxxx represents your cluster ID.**

workflow.xml

**You need to add the following command lines for Kerberos authentication.**

```
<credentials>
 <credential name='hcat_auth' type='hcat'>
 <property>
 <name>hcat.metastore.uri</name>
 <value>thrift://emr-header-1.cluster-xxxxx:9083</value>
 </property>
 <property>
 <name>hcat.metastore.principal</name>
 <value>hive/_HOST@EMR.xxxxx.COM</value>
 </property>
 </credential>
</credentials>
```

**You can find the demo on the following path:** `/home/${user}/examples/apps/spark`  
/



### 6.3.5.5.4 Schedule a Hive job

This topic describes how to use Oozie to schedule a Hive job.

job.properties

```
nameNode=hdfs://emr-cluster
jobTracker=emr-header-1.cluster-xxxxx:8032
oozie.use.system.libpath=true
jdbcURL=jdbc:hive2://emr-header-1.cluster-xxxxx:10000/default
jdbcPrincipal=hive/emr-header-1.cluster-xxxxx@EMR.xxxxx.COM
```

**xxxxx represents your cluster ID.**

workflow.xml

**You need to add the following command lines for authentication.**

```
<credentials>
 <credential name="hs2-creds" type="hive2">
 <property>
 <name>hive2.server.principal</name>
 <value>${jdbcPrincipal}</value>
 </property>
 <property>
 <name>hive2.jdbc.url</name>
 <value>${jdbcURL}</value>
 </property>
 </credential>
</credentials>
```

**You can find the demo on the following path:** /home/\${user}/examples/apps/hive2 /

## 6.3.6 Workflow

### 6.3.6.1 Workflow

After creating an E-MapReduce cluster, you can create workflow projects so that multiple jobs can be run simultaneously or sequentially.

### 6.3.6.2 Manage projects

This topic describes how to create a project, add users, and associate clusters.

Create a project

1. Log on to the [Alibaba Cloud E-MapReduce console](#) and go to the Cluster Management page.
2. Find the target cluster ID and click Manage in the Operation column.

3. On the page that appears, click the Workflow tab on the top to go to the Project List page.

If you use an Alibaba Cloud account, you can view all projects under your account, including projects of RAM users under your account. If you are a RAM user, you can only view projects for which you have the development permission. You can only use an Alibaba Cloud account to grant the development permission of the project to RAM users. For more information, see [Add a member](#).

4. In the upper right corner, click Create Project. The Create Project dialog box appears.
5. Enter the project name and description, and click Create.



**Note:**

You can only add project members with an Alibaba Cloud account. The Create Project button is available only when you log on to the E-MapReduce console with an Alibaba Cloud account.

#### Add a member

After creating a project, you can authorize RAM users to operate the project.

1. On the Project List page, find the target project and click Details in the Operation column.
2. Click the User Management tab.
3. Click Add User to add a RAM user under your Alibaba Cloud account to the project.

The RAM user becomes a member of the project and has the permissions to view and develop jobs and workflows under the project. If you want to remove the RAM user from the project, click Delete in the Operation column.



**Note:**

You can only add project members with an Alibaba Cloud account. The User Management tab is available only when you log on to the E-MapReduce console with an Alibaba Cloud account.

#### Associate a cluster with the cluster

After creating a new project, you need to associate a cluster with the project so that workflows in the project can be executed on the cluster.

1. On the Project List page, find the target project and click Details in the Operation column.
2. Click the Cluster Settings tab.
3. Click Add Cluster and select a subscription or pay-as-you-go cluster from the drop-down list. Clusters created for executing temporary jobs are not listed here.
4. Click OK.

To disassociate the cluster from the project, click Delete in the Operation column.



**Note:**

You can only associate cluster by using an Alibaba Cloud account. The Cluster Settings tab is available only when you log on to the E-MapReduce console with an Alibaba Cloud account.

To set queues and users for submitting jobs to the cluster, click Modify Configuration in the Operation column. The following table lists parameters for setting queues and users.

Table 6-6: Parameters

Parameter	Description
Default job submission user	Specifies the default Hadoop user who submits the job to the associated cluster in the project. The default value is <code>hadoop</code> , and there can be only one default user.
Default submit job queue	Specifies the default queue to which the jobs are submitted in the project. If you do not set this parameter, jobs will be submitted to the default queue.
Job submission user whitelist	Specifies Hadoop users who can submit jobs to the associated cluster. If you want to specify more than one user, separate them with commas (,).
Job submission queue whitelist	Specifies queues of the associated cluster that jobs in the project can run in. If you want to specify more than one queue, separate them with commas (,).
Client whitelist	Configures the client that can be used to submit jobs. The E-MapReduce master node or the E-MapReduce Gateway can be selected. Currently, your self-built Gateways deployed on ECS instances are not listed here.

### 6.3.6.3 Edit a job

After creating a project, you can create a job under the project.

Create a job

In the project, you can create jobs such as Shell, Hive, Spark, SparkSQL, MapReduce, Sqoop, and Pig.

1. Log on to the [Alibaba Cloud E-MapReduce console](#) and go to the Cluster Management page.
2. Click the Manage link in the Operation column.
3. Click the Workflow tab on the top to enter the Project List page.
4. Click Design Workflow to the right of the specified project and to go to the Edit Jobs page.
5. In the left-side navigation pane, right-click a folder as required and select Create Job from the drop-down list.
6. In the Create Job dialog box, enter a name and description for the job and select a job type.

Once selected, the job type cannot be modified.

7. Click OK.



**Note:**

You can also right-click the folder to create a subfolder, rename the folder, and delete the folder.

Develop a job

For more information about how to configure jobs, see the [Hadoop MapReduce job configuration](#) section of the E-MapReduce user guide.



**Note:**

When you click the Insert an OSS UNI button and select OSSREF as a File Prefix, E-MapReduce downloads OSS files to your cluster and add these files to a specified classpath.

Basic job settings

In the upper-right corner, click Job Settings. The Job Settings dialog box appears.

Table 6-7: Job settings description

Item	Description
Retries	Sets the number of retries when this job fails during the workflow running. This option will not take effect when you run the job on the Edit Job page.
Actions on failures	Sets whether to continue running the next job or suspend the current workflow when this job fails during the workflow running.
Resources	If you want to add resources such as jar packages or UDF that a job execution depends on, you must upload these files to OSS. When you select a resource, you can use this resource in a job directly.
Configuration parameters	Specifies the values of the parameters that the job uses. Take specifying.

### Advanced job settings

In the Job Settings dialog box, click the Advanced Settings tab.

Table 6-8: Advanced settings

Item	Description
Mode	Includes the YARN and LOCAL modes. YARN: Jobs are submitted by allocating resources on YARN through Launcher. LOCAL: The job runs on a specified local host.
Environment variables	The environment variables used to run the job. You can also export environment variables in the job script.
Scheduling parameters	Includes information, such as YARN queues of a job, vCPU, memory, and Hadoop user. If you do not specify these parameters, a job uses the default values of the Hadoop cluster.

### Execute a job

After the development and configuration of a job are complete, you can click Run in the upper-right corner to run the job.

## View logs

After you execute a job, you can view run logs on the Log tab at the bottom of the Query page. Click Details to go to the detailed log page of the job. You can view submit logs and Yarn container logs.

### 6.3.6.4 Design a workflow

An E-MapReduce workflow can be represented as a directed acyclic graph (DAG). You can pause, stop, and resume workflows. You can also view the running status of workflows in the web UI.

## Create a workflow

1. Log on to the [Alibaba Cloud E-MapReduce console](#) and go to the Cluster Management page.
2. Click the Manage link in the Operation column.
3. Click the Workflow tab on the top to enter the Project List page.
4. Click Workflows in the Actions column for a project. Click the Workflows tab and go to the Workflows page.
5. Right-click the target folder and click Create Workflow.
6. In the Create Workflow dialog box, enter a name and description for the workflow. Select a cluster from the Target Cluster drop-down list.

You can select an existing cluster (Subscription or Pay-As-You-Go) that is associated with the project to run the workflow, or you can use a cluster template to create a temporary cluster for running the workflow.

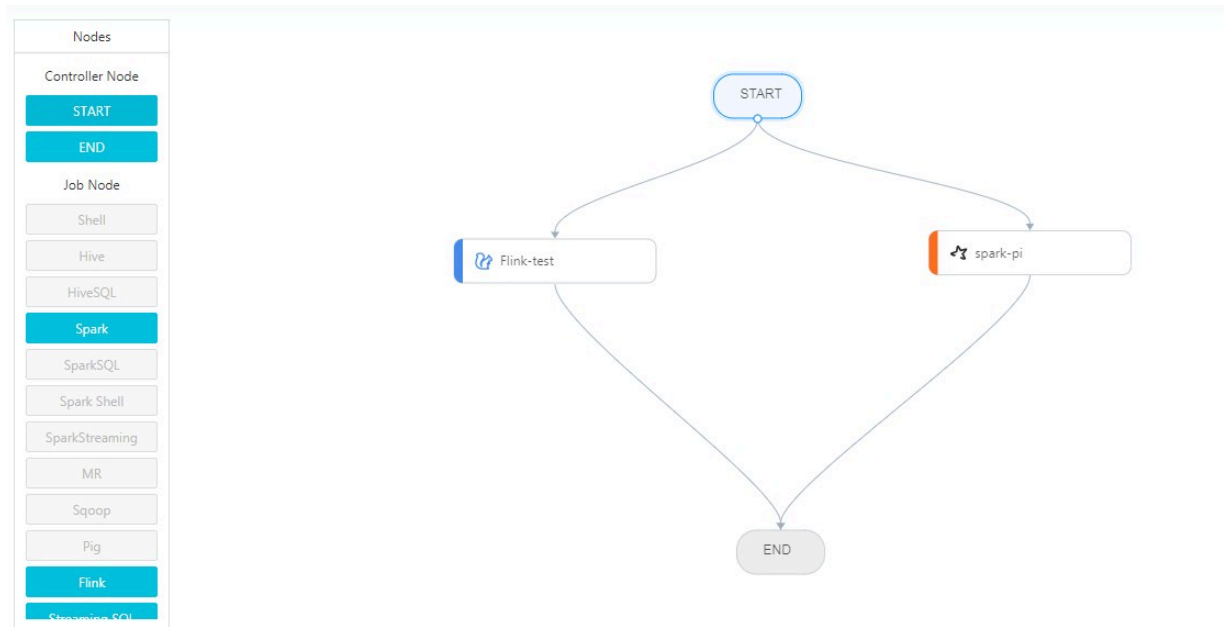
7. Click OK.

## Edit a workflow

You can drag and drop multiple types of job nodes on the Workflows canvas and connect the nodes to schedule jobs. After the jobs are scheduled, drag the END

controller node, drop it on the canvas, and connect it to a job node to complete the workflow design.

Figure 6-3: Edit a workflow



Configure a workflow

**In the upper-right corner of the Workflows canvas, click Configure to schedule a workflow.**

Table 6-9: Schedule description

Parameter	Description
Target cluster	You can select the cluster that runs the workflow.
Scheduling policy	<p>After workflow scheduling is enabled, you can choose a scheduling policy, including time scheduling and dependency scheduling.</p> <ul style="list-style-type: none"> <li>• <b>Time-based Scheduling:</b> Sets a start time, an end time, and a cycle for scheduling the workflow. During this period, the workflow is executed according to the cycle you have set.</li> <li>• <b>Dependency-based Scheduling:</b> Selects a project and selects a dependent workflow. The workflow is scheduled only when the dependent workflow finishes. You can select a maximum of one dependent workflow.</li> </ul>

## Run a workflow

**After the design and configuration are complete, click Run to run the workflow.**

## View and operate a workflow instance

**After you run the workflow, click the Records tab page to view the running status of the workflow instance. Click Details for the workflow instance to view the running status of the workflow instance. You can also pause, resume, stop, or rerun the workflow instance.**

Figure 6-4: Workflow running details

Workflow Instance Info

DAG

ID: FI-A-12

Workflow ID: F-3C9-1A

Status: RUNNING

Start Time: Aug 20, 2019, 13:54:49

Name: 111

Target Cluster: C-58-1F17

Duration: 1.806 Seconds

End Time:

Dependent Workflow:

Project	Workflow	Workflow Instance ID	Run At	Scheduled Time
No Data				

Refresh

Pause Workflow

Resume Workflow

Stop Workflow

Rerun Workflow Instance

Job Instance ID	Name ↕	Target Cluster	Job Type	Job Submission Node	Start Time ↕	Ended At ↕	Duration	Status ↕	Actions
FNI-C-13	Flink-test	C-58C47D12FB336F17	FLINK	emr-13	Aug 20, 2019, 13:54:49		1.808 Seconds	SUBMITTING	Details
FNI-3-1A	spark-pi	C-58C47D12FB336F17	SPARK	emr-1A	Aug 20, 2019, 13:54:50		0.808 Seconds	PREP	Details

Table 6-10: Operation description

Operations	Description
Suspend a workflow	The running job instance will continue to run, but the subsequent job instances will not. You can click Resume Workflow and the system will continue to run the subsequent jobs after the job instance is suspended.
Cancel a workflow	All running jobs in the workflow are stopped.
Rerun a workflow	The workflow runs from the START node.

### 6.3.7 Component endpoints

**EMR provides a unified management platform for most core components. You can go to the Web UIs of these components by using the links.**

- **HDFS UI**

**https://{cluster EIP}:8443/gateway/cluster-topo/hdfs/**



- Yarn UI

<https://{cluster EIP}:8443/gateway/cluster-topo/yarn/>

- SparkHistory UI

<https://{cluster EIP}:8443/gateway/cluster-topo/sparkhistory/>

- Ganglia UI

<https://{cluster EIP}:8443/gateway/cluster-topo/ganglia/>

- Oozie UI

<https://{cluster EIP}:8443/gateway/cluster-topo/oozie/>

- Hue UI

<http://{cluster EIP}:8888/>

## 6.4 Cluster O&M

### 6.4.1 Cluster O&M

After creating an EMR cluster, further O&M is required for the cluster to run jobs.

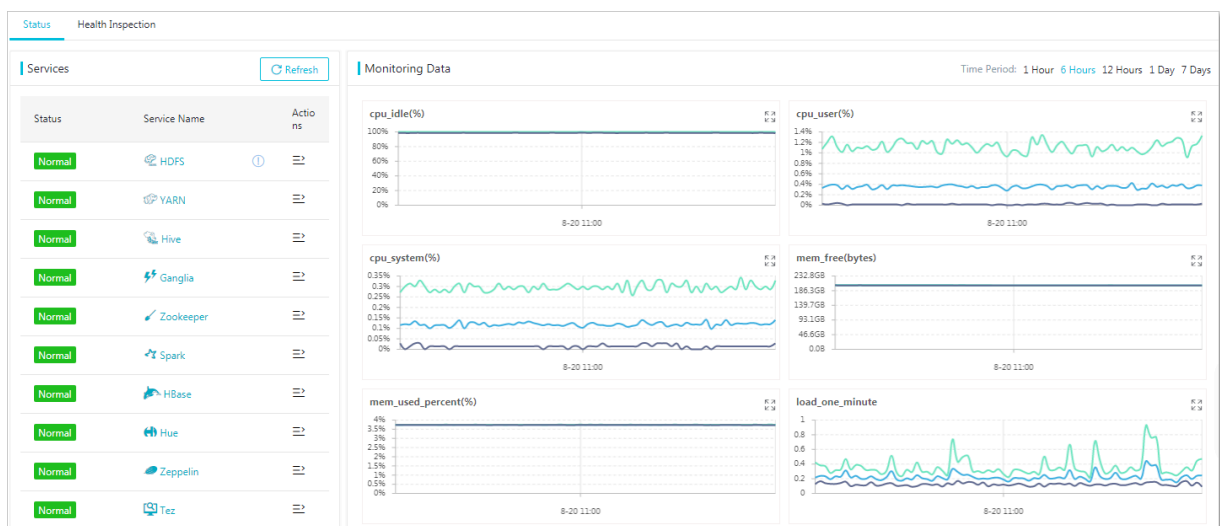
### 6.4.2 Component O&M

EMR enables you to use GUI to perform O&M for components that run on a cluster.

#### Service monitoring overview

EMR allows you to monitor all services in the back end, and update data and view history records in real time.

Figure 6-5: Service monitoring overview



Per-service detailed monitoring and periodical health check

**If a service contains multiple sub-services, you can view detailed information about all sub-services. EMR provides a set of default health check settings for all services . Any detected issues are automatically resolved as the services are updated.**

Service configuration change

**EMR allows you to add new parameters on the service configuration page of the console. Once you modify a parameter, EMR prompts all associated services that will be affected by the change. EMR supports rolling restart that ensures the high availability of the services.**

### 6.4.3 Basic operation environment and software environment O&M

**This topic describes the maintenance of the basic operation environment and software environment.**

**Currently, the basic operation environment of clusters is managed and maintained by the intelligent management system of EMR. The intelligent management system can automatically resolve problems such as hardware damage and failures by performing migration in the back end.**

**It monitors the clusters' application environment and all services in real-time and is able to immediately bring up any collapsed process. This pattern applies to most scenarios. For scenarios to which the pattern does not apply, contact the on-duty personnel to recover manually. In most cases, to resolve these problems, you only need to modify the service settings, such as modify the memory size.**



**Note:**

- **We recommend that you do not change the cluster settings. If your change causes an issue, submit a ticket so that we can help you locate the cause and fix the issue.**
- **Consult engineers for issues involved in business scenarios and non-technical issues of the cluster environment.**

## 7 Graph Analytics

---

### 7.1 What is Graph Analytics?

**Graph Analytics is a visual analysis platform for relationship networks. Graph Analytics is widely used in Alibaba Group and Ant Financial for risk control including anti-fraud, anti-theft, and anti-money laundering solutions. Graph Analytics provides solutions for multiple industries, including public security protection, taxation, customs, banking, insurance, and the Internet.**

**Graph Analytics is designed to facilitate multi-source data integration, computing applications, visual analytics, and intelligent businesses. Based on relationship networks, Graph Analytics can visualize the properties of objects and reveal the relationship among objects.**

**Graph Analytics provides features including relationship networks, search networks, intelligent networks, information cubes, intelligent judgement, collaboration and sharing, and dynamic modeling. It visualizes data and integrates machine computing capabilities with human cognition. This allows you to gain insight into massive data and obtain information and knowledge directly and efficiently.**

### 7.2 Quick Start

#### 7.2.1 Log on to Administration Console of Graph Analytics


**Administration Console is the data configuration platform for Graph Analytics. In Administration Console, you can configure the data sources, objects, links, events, and other advanced configuration items. Before you use Graph Analytics to analyze graphs, you must log on to Administration Console to perform the related configurations.**

##### Prerequisites

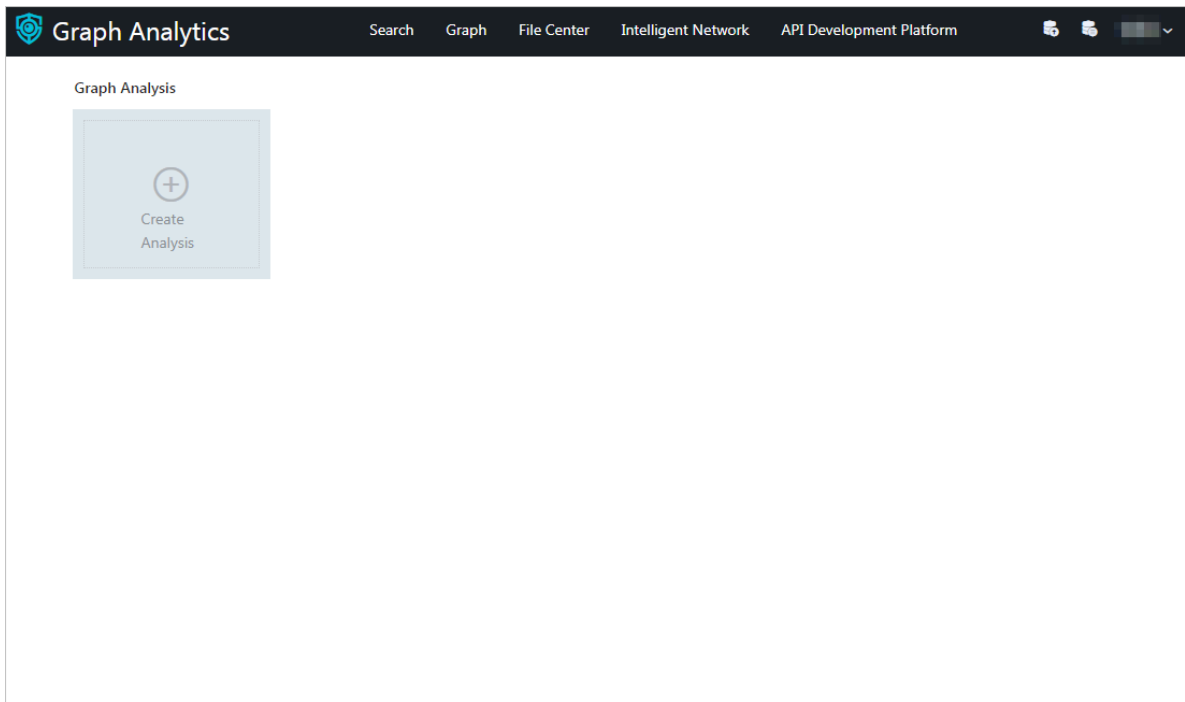
- **Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.**

- We recommend that you use the Chrome browser.

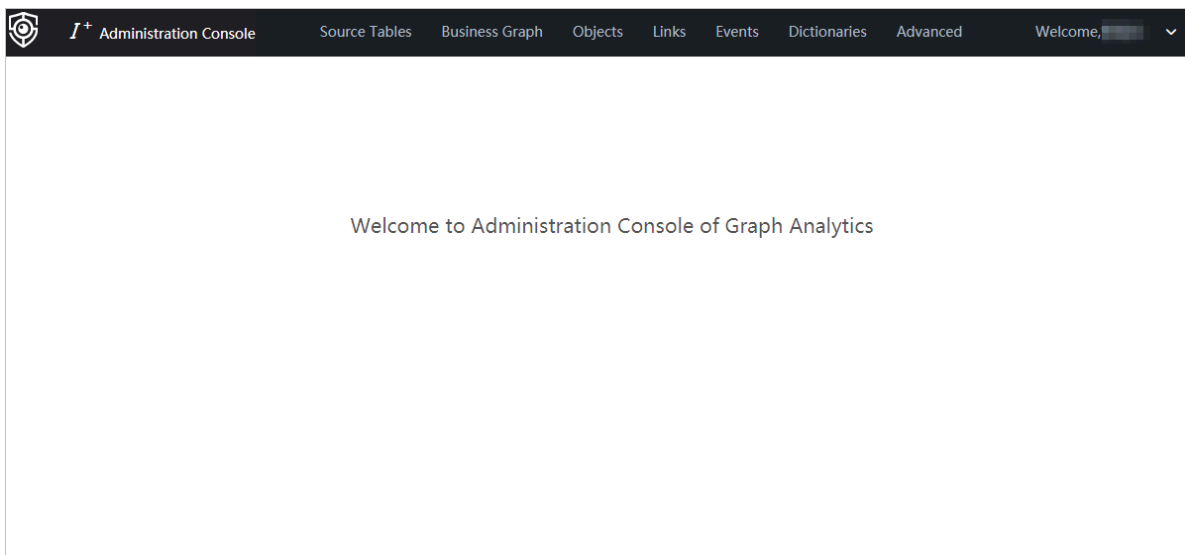
## Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username **super**. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. Click the  icon in the upper-left corner, and choose Big Data > Graph Analytics.  
Or, in the left-side Custom Menu, choose Big Data > Graph Analytics.

6. On the Graph Analytics page, select a Region and a Department, and click **iplus**.  
The homepage of Analytics Workbench appears.



7. Move your mouse pointer to the username in the upper-right corner, and select **Administration Console**. The page of Administration Console appears.



## 7.2.2 Create data sources

Before you perform a relationship analysis, you must integrate data that you want to analyze, typically databases, into Graph Analytics. These databases will be used

as data sources. In Graph Analytics, every data source is unique and can only be added once.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have obtained the data source address, user name, password, port number, and other information, and the data source is accessible.

### Context

A data source is one of the entries for new objects, links, and events. It is also the only entry for objects, links, and events to map to a data table. Objects, links, and events that are created on the Object Information, Link Information and Event Information pages are logical business objects, links, and events without mappings.



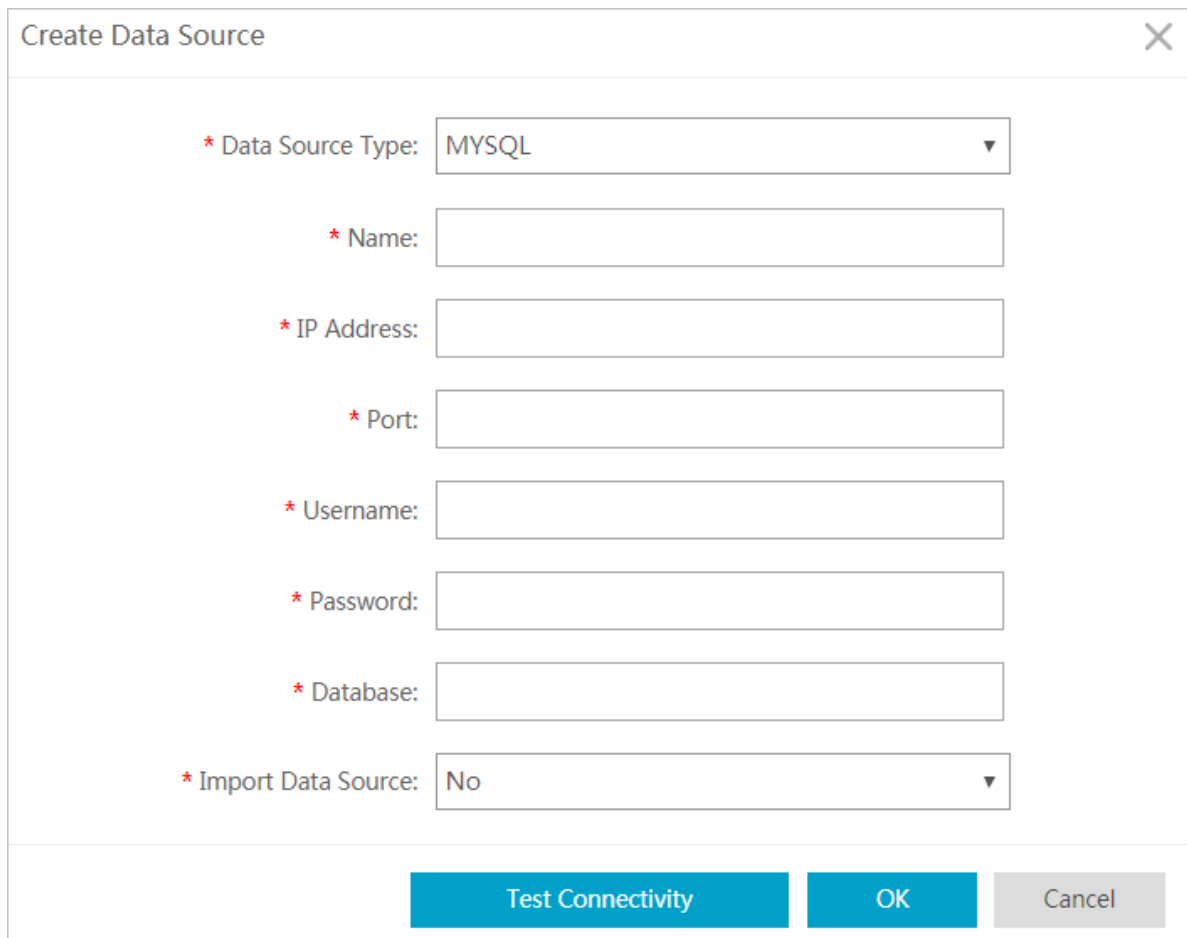
#### Note:

Objects, links, and events created through the data source only take effect after you log on again.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Source Tables to go to the Data Sources page.

3. On the Data Sources page, click the Create Data Source icon in the left-side navigation pane. The Create Data Source dialog box appears.



4. On the Create Data Source dialog box appears, specify the data source information as needed.

Table 7-1: Data source parameters

Parameter	Configuration method
Data Source Type	Select a data source type as needed. Supported data source types include: MYSQL, ORACLE, RDS, and GREENPLUM.
Name	The name of the data source. It can be user-defined.
IP Address	The IP address or domain name of the data source.
Port	The port number of the data source.
Group	The department or group to which the database belongs.
Username and Password	The username and password used to connect to the data source.

Parameter	Configuration method
Database	The actual name of the data source.
Import Data Source	When the data source type is not set to the ORACLE type, you must specify whether the data source is imported to the Graph Analytics system. You can only import data in Analytics Workbench after you have configured the imported data source.

5. After you have configured the preceding parameters, click **Test Connectivity** to check whether the data source can be connected.

If the data source is connected properly, the interface prompts the test to be normal. If the data source cannot be connected, check if the information is correct and the data source itself is normal.

6. After the connection test is confirmed as successful, click **OK**.

### 7.2.3 Create OLEP models for tables

After you add a data source, you must create object, link, event, and property (OLEP) models for the tables in the data source as needed. Before you configure OLEP tables, prepare the tables for which you will create OLEP models, the columns of each table, and the business models to be configured. Referenced tables cannot be deleted.

#### Prerequisites

You have created an accessible data source.

#### Context

OLEP models include the following three types of mappings: table-to-object mappings, table-to-link mappings, and table-to-event mappings. You can create objects, links, and events when you create OLEP models. Afterward, you can view and configure these objects, links, and events on the **Object Information**, **Link Information**, and **Event Information** pages, respectively. You can configure these items to reflect your business semantics. An OLEP table serves as a source for configuring objects, first-degree links, and events. A table can be mapped to multiple objects, links, and events.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click **Source Tables** in the top navigation bar. The **Data Sources** page appears.



3. Click a data source in the left-side navigation pane. The data source details are displayed on the right side of the page.
4. Click the Not Added tab. All tables in the data source that have no OLEP models are displayed.

You can search for a table quickly and accurately by specifying the Table Name, Table Description, or Table Group parameter.

Figure 7-1: Tables that have no OLEP models

iplus\_

Data Connection Information

Edit Information

IP Address :

Port : 3306

Username :

Password :

Data Source Type : MYSQL

Database : iplus\_

Import Data Source : No

Network Type : Classic Network

Added to OLEP

Not Added

Table Name:

Table Description:

Table Group:

Search

Table Name	Table Description	Created Links/Objects	Table Group	Table Ro...	Last Updated At	Actions
cust_login_info_tmp				56351		<a href="#">Add to OLEP</a>
cust_regist_info_tmp				2454		<a href="#">Add to OLEP</a>

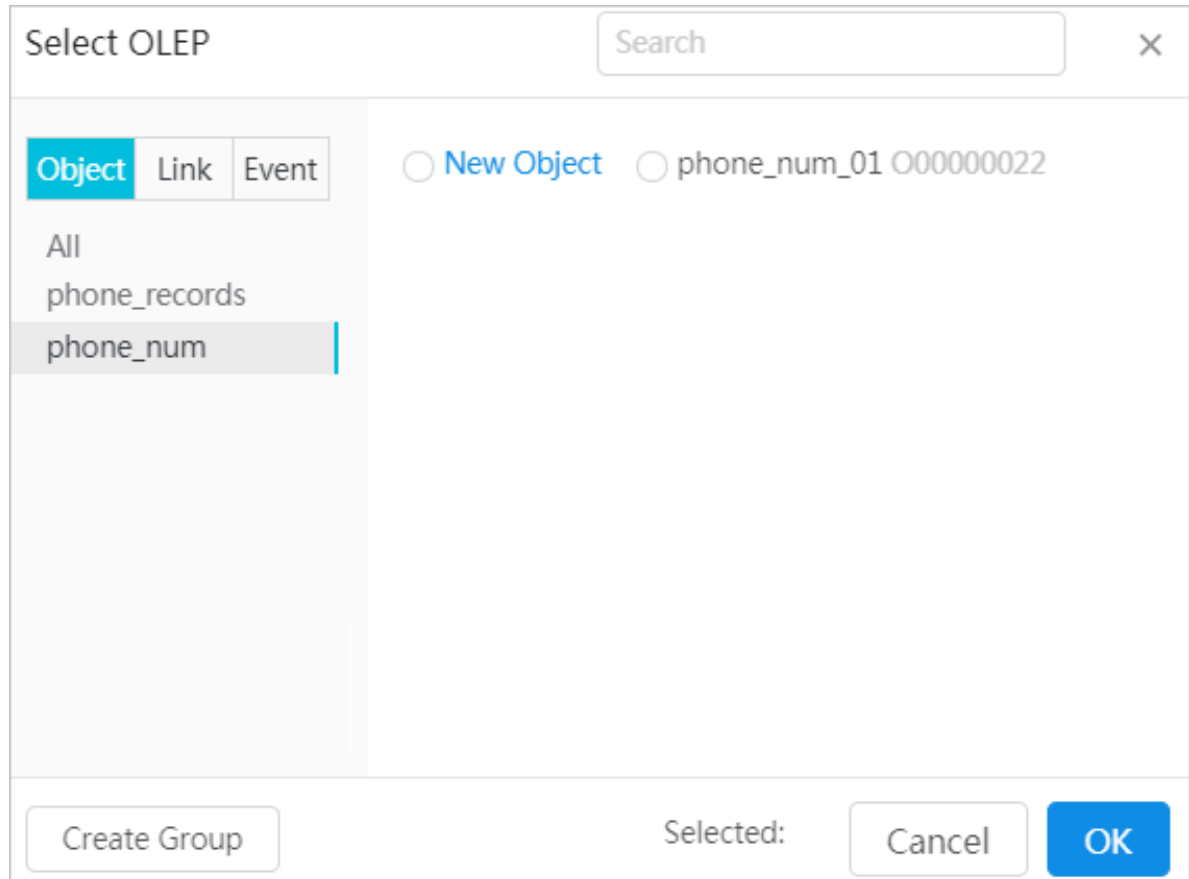
<

1

>

5. Select a table, and then click Add to OLEP in the Actions column. The Select OLEP dialog box appears.

Figure 7-2: Select OLEP dialog box



The Select OLEP dialog box contains the Object, Link, and Event tabs which are used to create mappings to objects, links, and events, respectively. For more information about how to create a mapping to an object, link, or event, see [step 6](#), [step 7](#), and [step 8](#).

If there are no existing object, link, or event groups that can meet your requirements, click Create Group to create a new object, link, or event group.

**6. Map the table to an object.**

- a) **Select New Object or an existing object and then click OK. The Map to Object dialog box appears.**

Figure 7-3: Map to a newly created object

Map to Object
×

\* Object Name: 
Group:

Property ID	Table Column	* Property Name	* Primary Key	Mapping <input checked="" type="checkbox"/>
O00000027P0001	<input type="text" value="identity_card"/>	<input type="text" value="identity_card"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
O00000027P0002	<input type="text" value="name"/>	<input type="text" value="name"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
O00000027P0003	<input type="text" value="phone_num"/>	<input type="text" value="phone_num"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

<
1
>

Cancel
OK

Figure 7-4: Map to an existing object

Map to Object
×

Object Name: 
Group: 
Add Property

Property ID	Table Column	Property Name	Primary Key
O00000022P0001	<input type="text" value="identity_card"/>	<input type="text" value="identity_card"/>	<input type="checkbox"/>
O00000022P0002	<input type="text" value="name"/>	<input type="text" value="name"/>	<input type="checkbox"/>
O00000022P0003	<input type="text" value="phone_num"/>	<input type="text" value="phone_num"/>	<input checked="" type="checkbox"/>

<
1
>

Cancel
OK

**b) Set the parameters as needed.**

- **New object:** Set parameters based on [Table 7-2: Parameters used to map the table to a new object](#).
- **Existing object:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing properties of the object, you can click Add Property to add new properties.

Table 7-2: Parameters used to map the table to a new object

Feature name	Description
Object Name	The user-defined object name. It must be unique.
Group	The object group to which the object belongs. All available object groups are displayed in the drop-down list.
Name	<p>The name of an object property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, property names are displayed instead of the actual table columns that are mapped to the properties.</p>
Mapping	Whether to enable the property mapping.
Primary Key	Sets a property as a primary key. Each primary key uniquely identifies an object. You must set one or more properties as primary keys for each object. You must enable Mapping for primary keys.

**c) Click OK.**

**7. Map the table to a link.**

- a) **Click the Link tab. All first-degree links to which the current table has been mapped are displayed.**

- b) **Select New Link or an existing link and then click OK. The Map to Link dialog box appears.**

Figure 7-5: Map to a newly created link

Map to Link
×

\* Link Name: 
Group:

\* Source Object: 
\* Target Object:

**Basic Information**

Property ID	Table Column	* Property Name	Mapping <input checked="" type="checkbox"/>
L00000016P0001	<input type="text" value="callee_num"/>	<input type="text" value="callee_num"/>	<input checked="" type="checkbox"/>
L00000016P0002	<input type="text" value="caller_num"/>	<input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>

<
1
>

**Source Property Mapping**

SourceObject Property:phone\_num\_01 - phone\_num
\* Link Property:

**Target Property Mapping**

TargetObject Property:phone\_num\_01 - phone\_num
\* Link Property:

Cancel
OK

Figure 7-6: Map to an existing link

Map to Link
×

Link Name: 
Group:

Source Object: 
Target Object: 
Create Property

**Basic Information**

Property ID	Table Column	Property Name
L00000014P0001	<input type="text" value="caller_num"/>	<input type="text" value="caller_num"/>
L00000014P0002	<input type="text" value="callee_num"/>	<input type="text" value="callee_num"/>

<
1
>

**c) Set parameters as needed.**

- **New link:** Set parameters based on [Table 7-3: Parameters used to map the table to a new link](#).
- **Existing link:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing link properties, you can click Add Property to add new link properties.

Table 7-3: Parameters used to map the table to a new link

Feature name	Description
Link Name	The user-defined link name. It must be unique.
Group	The link group to which the link belongs. All available link groups are displayed in the drop-down list.
Source	The source object of the link. You can select an object from the drop-down list. The Source Property Mapping parameter is available only after you set the Source Object parameter.
Target	The target object of the link. You can select an object from the drop-down list. The Target Property Mapping parameter is available only after you set the Target Object parameter.
Name	<p>The name of an event property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, the Property Name values are displayed as the table header in Details on the Graph page.</p>
Mapping	Whether to enable the property mapping.
Link Property in Source Property Mapping	The link property to which a primary key property of the source object is mapped.



Feature name	Description
<b>Link Property in Target Property Mapping</b>	<b>The link property to which a primary key property of the target object is mapped.</b>

d) Click OK.

**8. Map the table to an event.**

- a) **Click the Event tab. All events to which the current table has been mapped are displayed.**

- b) Select New Event or an existing event and then click OK. The Map to Event dialog box appears.**

Figure 7-7: Map to a newly created event

Map to Event

\* Event Name : 
Group:

**Basic Information**

Property ID	Table Column	* Property Name	* Primary Key	Mapping
E00000014P0001	<input type="text" value="callee_num"/>	* <input type="text" value="callee_num"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
E00000014P0002	<input type="text" value="caller_num"/>	* <input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

<
1
>

^ Primary Key Mappings of Correlated Objects
Add Mapping

phone\_num O00000022

phone\_num(O00000022P0003) :

phone\_num O00000022

phone\_num(O00000022P0003) :

Cancel
OK

Figure 7-8: Map to an existing event

Map to Event

Event Name : 
Group: 
Add Property

**Basic Information**

Property ID	Physical Table Field	Property Name	Primary Key
E00000014P0001	<input type="text" value="caller_num"/>	<input type="text" value="callee_num"/>	<input type="checkbox"/>
E00000014P0002	<input type="text" value="callee_num"/>	<input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>

<
1
>

**c) Set parameters as needed.**

- **New event:** Set parameters based on [Table 7-4: Parameters used to map the table to a new event](#).
- **Existing event:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing event properties, you can click Add Property to add new event properties.

Table 7-4: Parameters used to map the table to a new event

Feature name	Description
Event Definition Name	The user-defined event name. It must be unique.
Group	The event group to which the event belongs. All available event groups are displayed in the drop-down list.
Name	<p>The name of an event property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, the Property Name values are displayed as the table header in Details on the Graph page.</p>
Switch	Whether to enable the property mapping.
Primary Key	Sets a property as a primary key. Each primary key uniquely identifies an event. You must set one or more properties as primary keys for each event. Switch must be turned on for the properties that are set as primary keys.

Feature name	Description
Map Primary Keys to Correlated Objects	<p>Indicates the mappings between the primary keys of correlated objects and the event properties. At least two correlated objects are required. You can click Add Mapping to add more necessary mappings between the primary keys of correlated objects and the event properties.</p> <p>You must enable Mapping for the event properties to which the primary keys of the correlated objects are mapped.</p>

d) Click OK.

9. After you have created OLEP models for the table, click the Added to OLEP tab to check the results.

## 7.2.4 Add OLEP table columns

If a data table has been mapped to an object, link, or event, and the table still has unoccupied columns (columns that are not correlated with any object, link, or event), you can add these columns to the existing mappings as needed.

### Prerequisites

A data table has been mapped to an object, link, or event, but the table still has unoccupied columns.

### Context

Before you configure the OLEP table columns, sort out the columns for which you will create OLEP models and data types of the columns, especially the time columns

.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click Source Tables in the top navigation bar.
3. On the Data Sources page, click a data source in the left-side navigation pane, and then click the table to which you want to add the columns.

4. Click the Columns Not Added tab in the right-side area. In the Columns Not Added tab that appears, click Add in the Actions column.

Columns displayed in the Columns Not Added tab are not mapped to any object, link, or event.

5. In the Select OLEP dialog box that appears, select the object, link, or event to which the columns map, and then click OK.

The Select OLEP tab only displays the objects, links, and events that have mapped to the current data table.

The following example describes how to add columns to an object.

Select OLEP

Search

Object

☐ phone\_num\_01 000000022

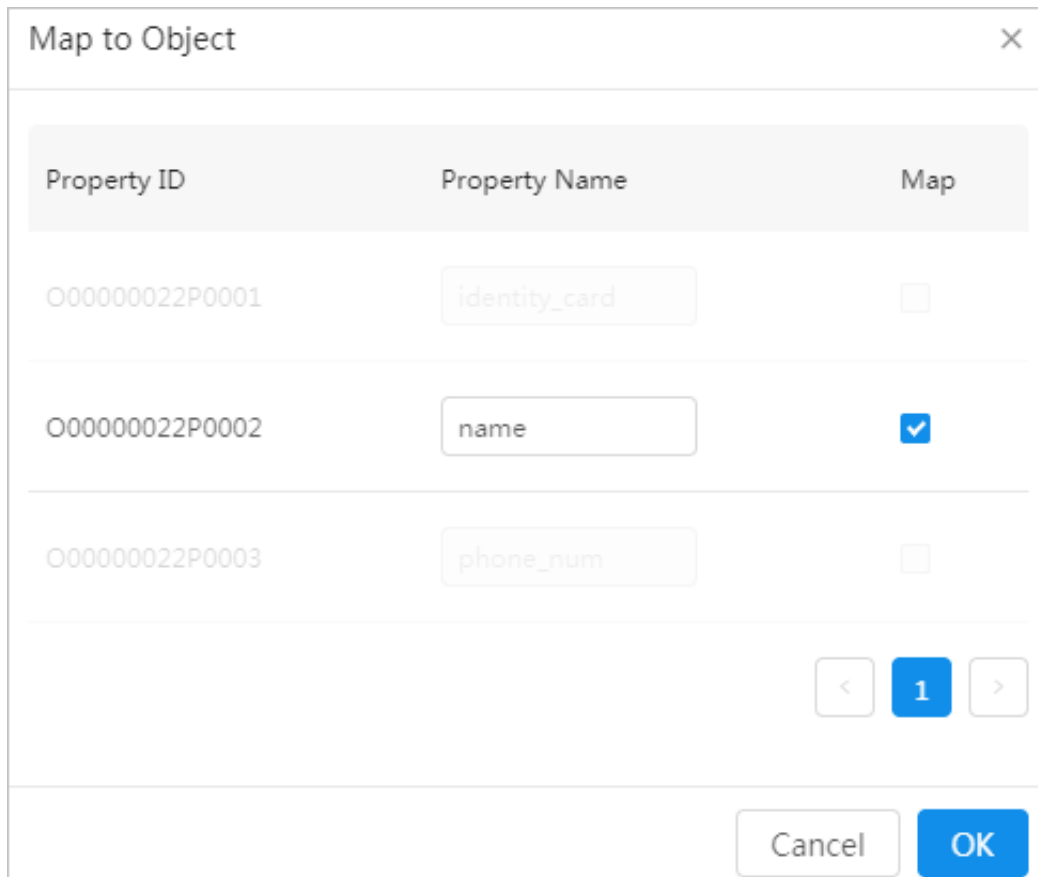
Link

Event

Cancel OK

6. In the Map to Object dialog box that appears, select the object properties to be mapped to the columns to be added.

Object properties that have been mapped to the current data table are gray and cannot be operated.



The 'Map to Object' dialog box displays a table with three columns: Property ID, Property Name, and Map. The first row shows 'O00000022P0001' with 'identity\_card' in a disabled text field and an unchecked checkbox. The second row shows 'O00000022P0002' with 'name' in a text field and a checked checkbox. The third row shows 'O00000022P0003' with 'phone\_num' in a disabled text field and an unchecked checkbox. At the bottom right are navigation arrows and a page number '1'. At the bottom are 'Cancel' and 'OK' buttons.

Property ID	Property Name	Map
O00000022P0001	identity_card	<input type="checkbox"/>
O00000022P0002	name	<input checked="" type="checkbox"/>
O00000022P0003	phone_num	<input type="checkbox"/>

< 1 >

Cancel OK

7. After you have configured the preceding parameters, click OK.

### 7.2.5 Configure object properties and business parameters

After you add an object, you need to configure the business parameters of the object based on your requirements so that you can view and analyze the object in Analytics Workbench.

#### Prerequisites

- Make sure that you have created a data source. For more information, see [Create data sources](#).
- Make sure that you have configured mappings between tables and objects, links, or events. For more information, see [Create OLEP models for tables](#).

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).

2. Click Objects on the top of the page.
3. In the left-side navigation pane of the Object Information page, click the name of the object to be configured and then click the Property Information tab on the right side.

If this object has been mapped to a physical table in the data source, the property information section displays the Data Source and Table information.

SV\_Account On

Object Information Property Information

Data Source: skyview Table: demo\_tranx\_id Save

Basic Information Add Row

Property ID	* Property Name	* Primary Key	* Show in Graph	Show in Properties	Conditional Query	Show in Statistics	Available	* Display Type	* Query Type	* Security Level	Actions
O00000072P0001	TRANX_ID	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	
O00000072P0002	BRANCH_NAM	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	
O00000072P0003	NAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	

Advanced

Add Image To : Logical Relation of Authorized Properties :

Location Settings Add Location

Trajectory Settings Add Trajectory

4. Set the basic configurations in the Basic Information page.

Click Add Row to add a property for a link in Basic information.


The configuration items in Basic Information are described in [Table 7-5: Description of basic configuration items](#).

Table 7-5: Description of basic configuration items

Configuration item	Description
Property ID	The ID of a property. It is automatically generated.
Property Name	The property name that is displayed on Analytics Workbench. Enter a name that describes your business.



Configuration item	Description
Primary Key	<p>You must select at least one primary key. After you complete the configuration, it cannot be modified or deleted.</p> <p>When you add a node in Analytics Workbench, you need to enter the physical table column mapped by the primary key property. For example, if the ID card number is the primary key, you need to enter the ID card number when you add an ID card node in Analytics Workbench.</p>
Show in Graph	<p>If you select this parameter, the Property Name of this property will be displayed in Graph in Analytics Workbench together with the object node. Otherwise, the property is not displayed. For example, if you select the ID card number, this number will be displayed in Graph together with the ID card object. If you select the name property at the same time, the name and the ID card number will be displayed together with the ID card object.</p> <p>If you select this parameter for a specific property, a bubble icon appears next to the option. You can click the bubble icon to set whether to show the Property Name in Graph. For example, if you select this parameter for the ID card number, and set to display the property name in Graph, the ID card property displayed in Graph will be ID card number: xxxxxx.</p>
Show in Properties	<p>If you select this parameter for a property, the property will be displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details tab and the Property tab. Otherwise, the property is not displayed.</p>
Conditional Query:	<p>If you select this parameter for a property, you can query the object based on this property in Target Object when you perform an analysis on the Graph page.</p>
Show in Statistics	<p>If you select this option for a property, the property will be displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; Statistics. Otherwise, the property is not displayed.</p>

Configuration item	Description
Available	<p>If you select this parameter for a property, the property takes effect and can be displayed in Analytics Workbench. This parameter must be selected for primary key properties.</p> <p>If any of the following parameters has been selected for the property: Primary Key, Show in Graph, Show in Properties, Conditional Query and Show in Statistics, the Available parameter is automatically selected for a property. The Available parameter is automatically deselected if you deselect all the preceding parameters.</p>
Display Type	<p>After you set the display type, the property is displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details tab and the Property tab based on the selected type.</p> <div>  <b>Note:</b> To display a property in the format of Dictionary, you need to configure a dictionary first. </div>
Query Type	The data type that is supported in the query condition of a property. For Display Type, if you select Dictionary, you must select Dictionary Option for Query Type.
Security Level	The security level for a property. A user with a lower security level cannot view the property.
Search Item Configuration	Associates this property with a search item so that the object can be searched by this property in Analytics Workbench. For more information about search item settings, see <a href="#">Configure a search item</a> .
Default Query Condition Settings	Defines the default condition used for an object query. If other properties are used as conditions for a query, this condition is also included by default.
Authorization Code	After the authorization code function has been enabled, only users with the required authorization code can access this property.
Derived Property	Sets a property as a derived property so that it can be generated automatically based on other properties. You can set the derivative method based on your requirements.

5. **Optional:** If you need to add multiple properties, you can refer to the preceding steps to add more properties.
6. **Optional:** Set the configurations in Advanced.

The configuration information of Advanced is described in [Table 7-6: Description of advanced configuration items](#).

Table 7-6: Description of advanced configuration items

Configuration item	Description
Add Image To	Specifies the avatar of the object that is displayed in Graph. Select a property of the object, and then set the URL of the image and the suffix of the image. Add Image To contains the prefix, the property, and the suffix. The prefix is the URL of the image, and the suffix is the image format.
Logical Relation of Authorized Properties	<p>The logical relationship between the authorization codes of properties in each record:</p> <ul style="list-style-type: none"><li>• <b>AND:</b> The current record is visible only to the users who meet all authorization code conditions of the properties in this record.</li><li>• <b>OR:</b> The current record is visible to the users who meet any one authorization code condition of the properties in this record.</li></ul>

7. Click Save.

## 7.2.6 Configure link properties and business parameters

After you add a first-degree link, you need to configure the properties and business parameters of the link based on your business requirements, so that you can view and apply this link in Analytics Workbench. This topic describes how to configure the properties and business parameters of a first-degree link.

### Prerequisites

You have created a first-degree link. For more information about how to create a link, see [Create a first-degree link](#).

### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click Links on the top of the page.

3. In the left-side navigation pane, click the link group that contains the first-degree link to be configured, and then click the link.
4. In the right-side area, click the Correlations and Properties tab.

On the Correlations and Properties page, if the link has been mapped to a data table in the data source, the property information section will display Data Source and Table. You can click the table name to go to the table page.

call\_link\_01 On

Link Information Correlations and Properties

Data Source: demo01 Table: call\_records Save

Basic Information Add Row

Property ID	* Property Name	* Unique ID	Show in Details	Conditional Query	Show in Statistics	Available	* Display Type	* Query Type	Security Level	Actions
L00000014P0001	caller_num	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal ...	S1	
L00000014P0002	callee_num	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal ...	S1	

Source Property Mapping

Source Object Correlated Property : phone\_num\_01-phone\_num \* Link-Correlated Property : caller\_num

Target Property Mapping

Target Object Correlated Property : phone\_num\_01-phone\_num \* Link-Correlated Property : callee\_num

Advanced

Accumulative Statistics Settings

Link Weight Settings

5. Set the basic configurations in the Basic Information page.

Click Add Row to add a property for a link in Basic information.




#### Note:

Link-Correlated Property in Source Property Mapping and Link-Correlated Property in Target Property Mapping are not the same and they are both required to be configured. Therefore, a link must have at least two properties.

The configuration items in Basic Information are described in [Table 7-7: Description of basic configuration items](#).

Table 7-7: Description of basic configuration items

Configuration item	Description
Property ID	The ID of a property. It is automatically generated.

Configuration item	Description
Property Name	The name of the property. If you select <b>Available</b> , the name of the property will be displayed in Analytics Workbench.
Unique ID	Defines the logical primary key of a link table.
Show in Details	If you select this item, the property will be displayed in the Details tab in Analytics Workbench.
Conditional Query	If you select this item, when you perform an analysis on the Graph page in Analytics Workbench, you can query a link based on the link type in Link Type.
Show in Statistics	If you select this item, the property is displayed in Analytics Workbench > Graph > right-side navigation pane > Statistics. Otherwise, it is not displayed.
Available	<p>If you select this item, the property takes effect and is displayed in Analytics Workbench.</p> <p>The <b>Available</b> parameter is automatically selected for a property if any of the following parameters has been selected for the property: Unique ID, Show in Details, Conditional Query, and Show in Statistics. The <b>Available</b> parameter is automatically deselected for a property if all of the preceding parameters are deselected for the property.</p>
Display Type	<p>After you set the display type, the property is displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details tab and the Property tab based on the selected type.</p> <div>  <b>Note:</b> To display a property in the format of Dictionary, you need to configure a dictionary first. </div>
Query Type	The data type that is supported in the query condition of a property. If you select <b>Dictionary</b> for Display Type, you must select <b>Dictionary Option</b> for Query Type.
Security Level	The security level for a property. A user with a lower security level cannot view the property.
Search Item Configuration	Associates this property with a search item so that the link can be searched by this property in Analytics Workbench.

Configuration item	Description
Default Query Condition Settings	Specifies the default condition used for a link query. If other properties are used as conditions for the query, this condition is also included by default.
Authorization Code	After the authorization code function has been enabled, only users with the required authorization code can access this property.
Derived Property	Sets a property as a derived property so that it can be generated automatically based on other properties. You can set the derivative method based on your requirements.
Move Up and Move Down arrows	The Move Up arrow and the Move Down arrow can be used to adjust the order of properties that are displayed in Analytics Workbench.

**6. You can set Link-Related Property in Source Property Mapping and Target Property Mapping.**

These two configurations are related with the `Source Object` and the `Target Object` of the link.

Take `Source Property Mapping` as an example: `Source Object Correlated Property` and `Link-Related Property` must be mapped to the same column in the same table. The `Source Object Correlated Property` parameter is the primary key property of the source object, which is automatically loaded according to the `Source Object` parameter. For the `Link-Related Property` parameter, you must select the link property that is mapped to the same column in the same table as the `Source Object` primary key.

Set `Target Property Mapping` in the same way you set `Source Property Mapping`.

## 7. Optional: Set Advanced, Accumulative Statistics Settings, and Link Weight Settings based on your requirements.

For more information about the configurations of key parameters, see [Table 7-8: Parameter configurations](#).

Table 7-8: Parameter configurations

Category	Configuration item	Description
Advanced	Chronological Time Property	Specifies the link properties based on which chronological analysis is performed. From the drop-down list, select one or more link properties of which the query type is time.
	Time Property for Behavior Analysis	Specifies the link properties based on which behavior analysis is performed. From the drop-down list, select one or more link properties of which the query type is time.
	Linked Times	Specifies the property of which the number of the same values are counted. The total number is displayed as the number of link occurrences. The Linked Times parameter is used as the default setting to filter link types. For example, if there are two lines of $A > C$ calls in the call log, the analysis result displays that the number of $A > C$ calls is two.
	Details Sorting Property	Specifies the property by which the returned behavior details are sorted by default.
Accumulative Statistics Settings	N/A	<p>Used to perform logical statistics for link properties of which the query type is numeric range. The logical statistics operations include top, =, and <math>\geq</math>. This configuration applies to business scenarios where statistics filtering is required for link query results. The Linked Times parameter is used to filter records in link query results.</p> <p>You can add statistical conditions to filter the link properties of which the query type is numeric range.</p>

Category	Configuration item	Description
Link Weight Settings	N/A	You can specify a link property of which the query type is numeric range and calculate the link weight based on the numeric range specified for the link property.

8. After you have modified the parameters, click **Save**. A message is displayed, indicating that the modifications have been saved.

### 7.2.7 Configure event property parameters

After you have created an event, you must configure the event properties. Event properties are critical to an event. You can configure event properties, and correlate the properties to objects on the Property Information tab.

#### Prerequisites

Make sure that you have obtained an account and a password for Graph Analytics and you have been authorized with the required Event Permissions.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click the **Events** tab in the top navigation bar. The Event Information page appears.
3. Click an event in the left-side navigation pane, and click the **Property Information** tab on the right side of the page. The Property Information tab appears.
4. Set the required event parameters.

The required parameters are displayed in the Basic Information and Set Mappings Between Correlated Objects and Properties areas. The event parameters are described in [Table 7-9: Required event property parameters](#).




**Note:**





**To save the property information, you must set all the required parameters.**

Table 7-9: Required event property parameters

Area	Parameter	Description
Basic Information	Property ID	The ID of a property. It is automatically generated.
	Property Name	The property name that is displayed on Analytics Workbench. We recommend that you enter a name that describes the business type.
	Primary Key	Each primary key uniquely identifies an event. A property cannot be deleted after it has been configured as a primary key.

Area	Parameter	Description
	Show in Graph	<div>  <b>Note:</b>  For each event, you must select at least one property to be displayed in the graph. </div> <p>If you select this parameter, the Property Name of this property will be displayed in Graph in Analytics Workbench together with the object node. Otherwise, the property is not displayed. For example, if this parameter is selected by the ID card property of an event, the ID card number will be displayed in Graph with the event. If the name property is also selected, the name and the ID card number will be displayed with the event.</p> <p>If you select this parameter for a specific property, a bubble icon appears next to the option. You can click the bubble icon to set whether to show the Property Name in Graph. For example, if you select this parameter for an ID card number, and set to display the Property Name in Graph, the ID card property displayed in Graph will be ID card number: xxxxxx.</p>
	Show in Properties	If you select this parameter for a property, the property will be displayed in the right-side pane on the Graph page of Analytics Workbench. To view the property, choose Properties > Event Properties.
	Element Identifier	The element identifier of a property. Set this parameter based on the actual property.

Area	Parameter	Description
	<b>Conditional Query</b>	If you select this parameter for a property, the event can be queried based on this property in Link Type on the Graph page of Analytics Workbench when you perform a relationship analysis.
	<b>Show in Statistics</b>	If you select this parameter for a property, the event will be displayed in the right-side pane on the Graph page of Analytics Workbench. To view the property, choose Statistics > Event Distribution.
	<b>Available</b>	<p>If you select this parameter for a property , the property takes effect and can be displayed on the Graph page of Analytics Workbench.</p> <p>This parameter is selected by default and cannot be changed.</p>
	<b>Display Type</b>	The format in which a property is displayed in the right-side pane of the Graph page on Analytics Workbench. Set this parameter as needed.
	<b>Query Type</b>	The data type that is supported in the query condition of a property.
	<b>Security Level</b>	The security level for a property. A user with a lower security level cannot view the property.

Area	Parameter	Description
	Search Item Configuration	<p>Click the More icon  in the Actions column corresponding to a property. Set the following four parameters:</p> <ul style="list-style-type: none"> <li>• <b>Search Item Configuration:</b> Search items are displayed in the drop-down list only after they have been configured in <a href="#">Configure a search item</a>.</li> <li>• <b>Default Query Condition Settings:</b> The default condition used for a link query. If other properties are used as conditions for a query, this condition is also included by default.</li> <li>• <b>Authorization Code:</b> After the authorization code function has been enabled, only authorized users can access this property.</li> <li>• <b>Derived Property:</b> After a property is set as a derived property, it can be generated automatically based on other properties. Configure the method in which the column is generated based on your needs.</li> </ul>
	Default Query Condition Settings	
	Authorization Code	
	Derived Property	
	Delete	When a property is no longer used, you can click the Delete icon  to delete this property.
Set Mappings Between Correlated Objects and Properties	Add Correlated Object	<p>Adds a mapping between an object and the event. One event must have at least two objects mapped to it.</p> <p>Click Add Correlated Object to add a correlated object, and configure the mapping between the event and the primary keys of the object you have added.</p>

5. **Optional:** After you have set the required parameters, you can set the optional parameters as needed.

The optional parameters are included in the Advanced and Display Settings areas. The optional parameters are described in [Table 7-10: Optional event property parameters](#).



**Note:**

**Location Settings are currently not supported.**

Table 7-10: Optional event property parameters

Area	Parameter	Description
Advanced	Behavior Property	Defines the properties based on which a behavior analysis is performed.
	Default Details Sorting Property	Defines the property by which the details are sorted.
	Logical Relation of Authorized Properties	<p>The logical relationship between the authorization codes of properties in each record.</p> <ul style="list-style-type: none"><li>• <b>AND:</b> The current record is visible only to the users who meet all authorization code conditions of the properties in this record.</li><li>• <b>OR:</b> The current record is visible to the users who meet any one authorization code condition of the properties in this record.</li></ul>
Display Settings	Enable Display	Indicates whether to show the event details.
	Group-by Properties	Indicates the property based on which events are aggregated. For example, aggregate Travel Events into a folder based on the Train Number property.

6. After you have completed the configurations, click **Save** in the upper-right corner. A success message is displayed after the modifications have been saved.
- An event is automatically enabled after its properties have been saved.


## 7.2.8 Log on to Analytics Workbench

Analytics Workbench is a data analysis platform of Graph Analytics. After you have configured relevant data in Administration Console, you can perform data analyses in Analytics Workbench.

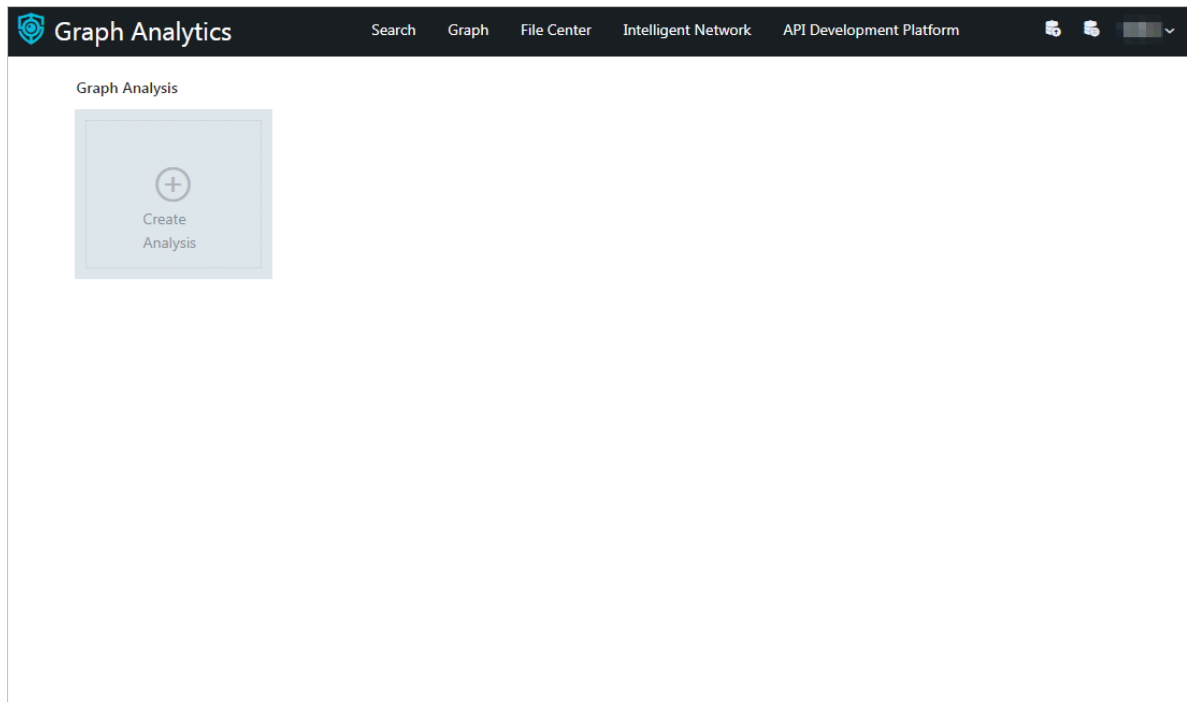
### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username **super**. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. Click the  icon in the upper-left corner, and choose Big Data > Graph Analytics.  
Or, in the left-side Custom Menu, choose Big Data > Graph Analytics.

6. On the Graph Analytics Iplus page, select a Region and a Department, and click Iplus. The homepage of Analytics Workbench appears.



## 7.2.9 Create analyses

After you log on to Analytics Workbench, you must create an analysis and add the objects to be analyzed as nodes before you analyze the nodes.

### Prerequisites

- Make sure that you have obtained an account and a password with the permission to perform graphic operations.
- Make sure that you have created source tables, objects, links, and events.
- Make sure that you have obtained data in the tables that have been mapped to the primary keys of the objects to be analyzed. You can obtain the data by querying the corresponding tables in the database.


### Procedure

1. [Log on to Analytics Workbench](#).
2. Click **Create Analysis**. A **Temporary Analysis** tab page appears.

3. Click Add in the toolbar and then click the blank space, or right-click the blank space and select Add Node. Set the parameters in the Add Node dialog box that appears.

The parameters are described in [Table 7-11: Parameters descriptions for adding a node](#).

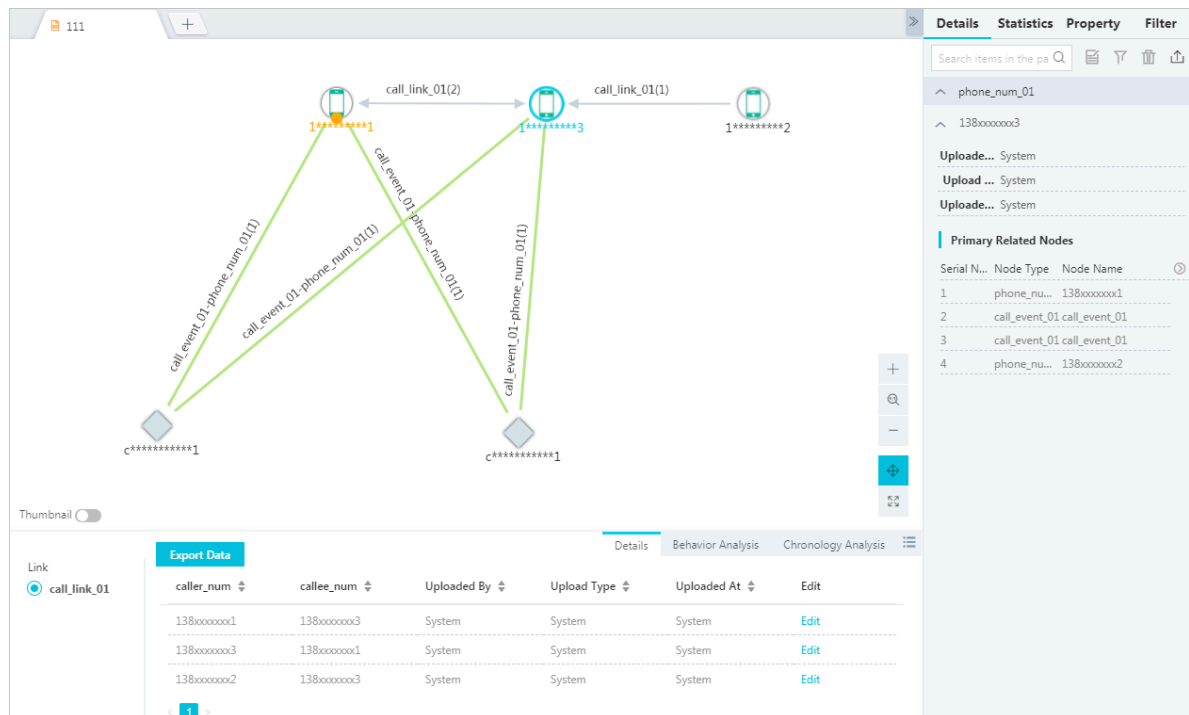
Table 7-11: Parameters descriptions for adding a node

Parameter	Description
Object type	<p>The drop-down list displays all created objects. Select an object as needed.</p> <div> <b>Note:</b> Graph Analytics supports adding compound nodes. A compound node is defined by multiple sub-types. For example, you can specify two sub-types for the object type "person": ID card and passport. The person can be uniquely identified by either the ID card or the passport.</div>
Text area	<p>Enter one or more primary key values.</p> <p>Separate multiple primary key values with commas (,).</p>

4. Click OK.



5. Right-click a node that has been added, and select Quick Extension. The system automatically performs a link analysis based on the configured data sources, objects, links, and events, and displays the analysis results in a graph.



6. Select one or more objects, links, or events. Click Behavior Chronology in the lower-right corner to see the corresponding Details, Behavior Analysis, and Chronology Analysis information.
7. Select one or more objects, links, or events. Click the icon (⏪) in the upper-right corner of the right-side pane to see the corresponding information on the Details, Statistics, Property and Filter tabs.
8. After the analysis has been completed, click Save in the upper-right corner. In the Save Analysis dialog box, enter a File Name and select a folder, and then click OK. A success message is displayed after the file has been saved.
- After you have saved the analysis file, if a collaborative analysis is required, you can share this personal analysis with other members.
9. Click the Share icon in the upper-right corner to specify the members you want to share this analysis with.

## 7.2.10 View analyses

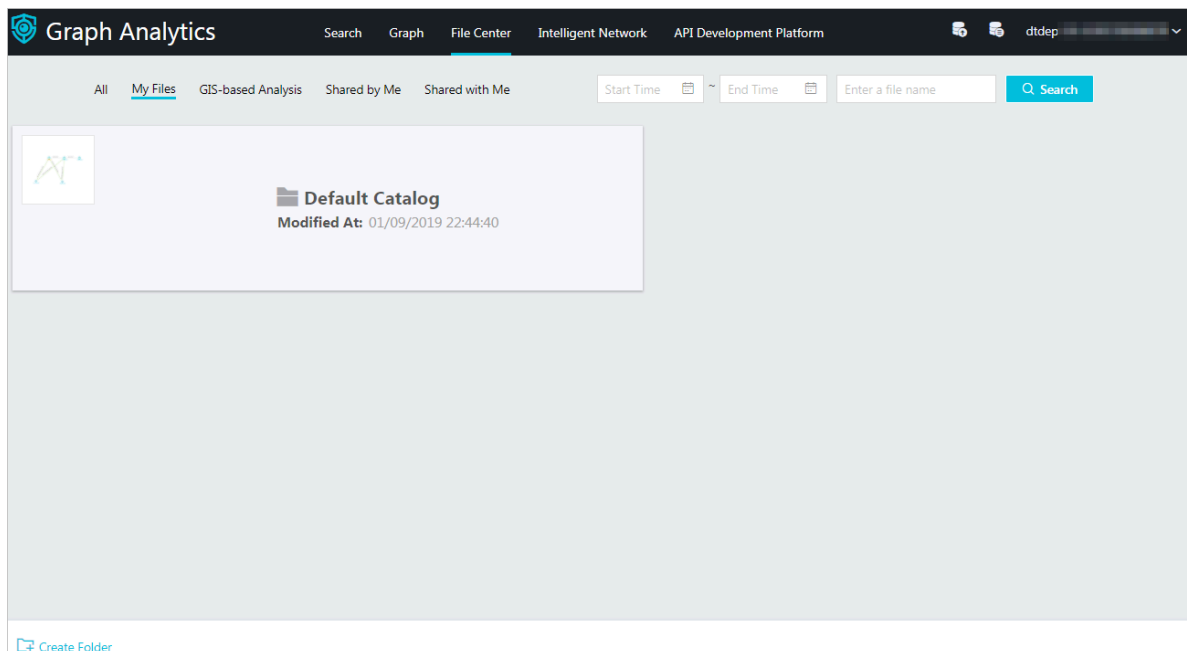
After you close the analysis file or log on to Analytics Workbench again, you can view the existing analyses or shared analyses in File Center.

### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Click File Center in the top navigation bar. The File Center page appears.



3. Click My Files, Shared by Me, or Shared with Me to view the corresponding analysis files.
4. Double-click an analysis to directly open the analysis file on the Graph page.

## 7.3 Source tables

### 7.3.1 Data sources

#### 7.3.1.1 Create data sources

Before you perform a relationship analysis, you must integrate data that you want to analyze, typically databases, into Graph Analytics. These databases will be used

as data sources. In Graph Analytics, every data source is unique and can only be added once.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have obtained the data source address, user name, password, port number, and other information, and the data source is accessible.

### Context

A data source is one of the entries for new objects, links, and events. It is also the only entry for objects, links, and events to map to a data table. Objects, links, and events that are created on the Object Information, Link Information and Event Information pages are logical business objects, links, and events without mappings.



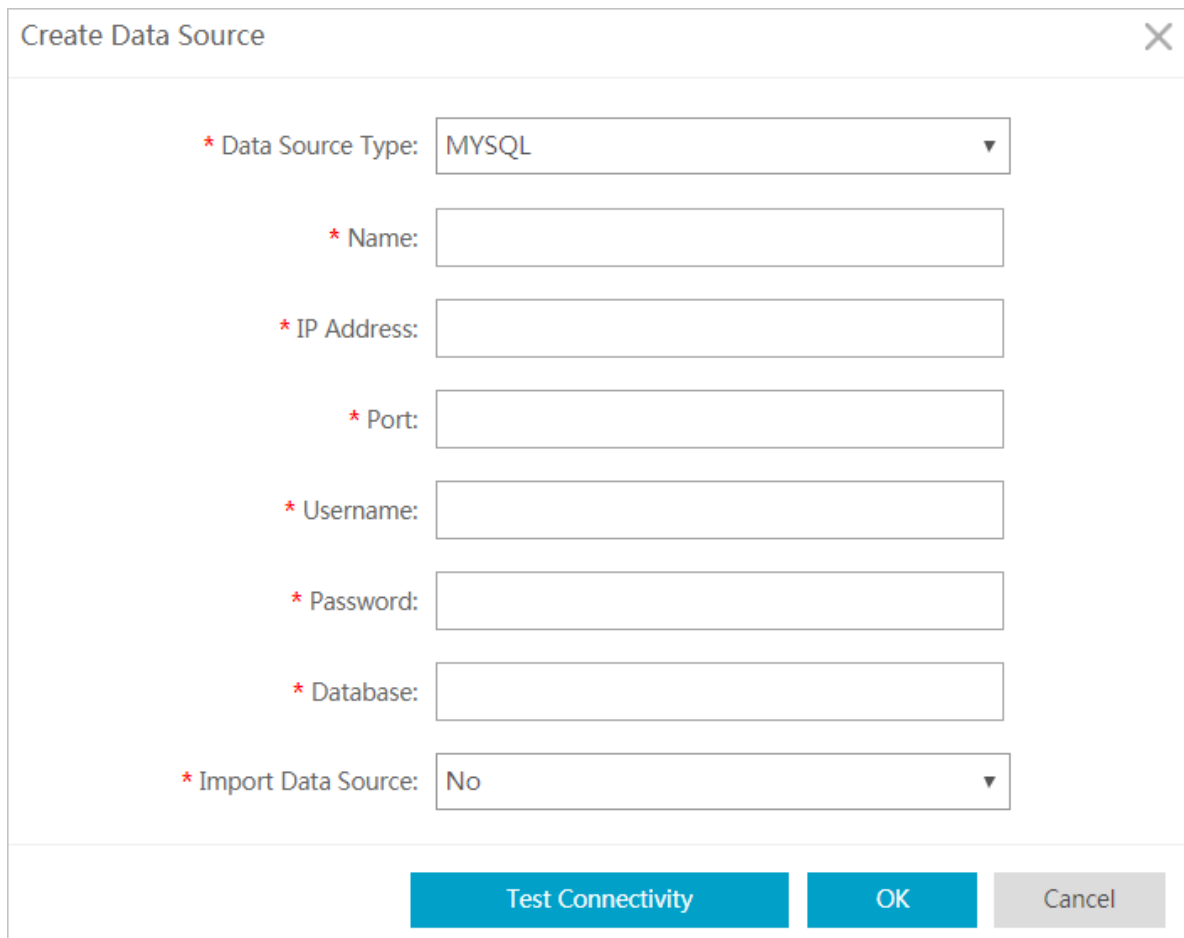
#### Note:

Objects, links, and events created through the data source only take effect after you log on again.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Source Tables to go to the Data Sources page.

3. On the Data Sources page, click the Create Data Source icon in the left-side navigation pane. The Create Data Source dialog box appears.



Create Data Source

\* Data Source Type: MYSQL

\* Name:

\* IP Address:

\* Port:

\* Username:

\* Password:

\* Database:

\* Import Data Source: No

Test Connectivity OK Cancel

4. On the Create Data Source dialog box appears, specify the data source information as needed.

Table 7-12: Data source parameters

Parameter	Configuration method
Data Source Type	Select a data source type as needed. Supported data source types include: MYSQL, ORACLE, RDS, and GREENPLUM.
Name	The name of the data source. It can be user-defined.
IP Address	The IP address or domain name of the data source.
Port	The port number of the data source.
Group	The department or group to which the database belongs.
Username and Password	The username and password used to connect to the data source.

Parameter	Configuration method
Database	The actual name of the data source.
Import Data Source	When the data source type is not set to the ORACLE type, you must specify whether the data source is imported to the Graph Analytics system. You can only import data in Analytics Workbench after you have configured the imported data source.

- After you have configured the preceding parameters, click **Test Connectivity** to check whether the data source can be connected.

If the data source is connected properly, the interface prompts the test to be normal. If the data source cannot be connected, check if the information is correct and the data source itself is normal.

- After the connection test is confirmed as successful, click **OK**.

### 7.3.1.2 View data sources

#### Procedure

- [Log on to Administration Console of Graph Analytics.](#)
- Click **Source Tables** in the top navigation bar. The **Data Sources** page appears.
- Select a data source from the left-side navigation pane. The data source details are displayed on the right side of the page, as shown in [Figure 7-9: View a data source](#).

Figure 7-9: View a data source

**Data Connection Information** [Edit Information](#)

IP Address :	Port : 3306
Username :	Password :
Data Source Type : MYSQL	Database : iplus_
Import Data Source : No	Network Type : Classic Network

Table Name:  Table Descriptions:  Table Group:

Table Name	Table Description	Created Links/Objects	Table Group	Table Ro...	Last Updated At	Actions
cust_login_info_tmp				56351		<a href="#">Add to OLEP</a>
cust_regist_info_tmp				2454		<a href="#">Add to OLEP</a>

### 7.3.1.3 Modify a data source

#### Prerequisites

To modify a data source, you must have the user password that is used to connect to the data source.

#### Context

You can modify all parameters of a data source except Data Source Type.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click Source Tables in the top navigation bar. The Data Sources page appears.
3. Click a data source in the left-side navigation pane, and click Edit Information in the Data Connection Information section on the right side of the page. Or, click the parent directory of the target data source in the left-side navigation pane, and click Modify next to the target data source in the data source list on the right side of the page.
4. In the dialog box that appears, modify the data source parameters as needed, and enter the user password.
5. Click Test Connectivity to verify the modifications.
6. After the verification is successful, click OK.


### 7.3.1.4 Delete data sources

When a data source is no longer used, you can delete the data source.

#### Prerequisites

All tables in the data source are not correlated with any objects, links, or events. If a data table is correlated with an object, link, or event, remove the association first. For more information, see [Remove OLEP tables.](#)

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click Source Tables in the top navigation bar. The Data Sources page appears.
3. Select one or more data sources from the left-side navigation pane, and then click the Delete icon () at the top of the navigation pane.
4. In the Delete Data Sources message box that appears, click OK to delete the specified data sources.

## 7.3.2 OLEP tables

### 7.3.2.1 Create OLEP models for tables

After you add a data source, you must create object, link, event, and property (OLEP) models for the tables in the data source as needed. Before you configure OLEP tables, prepare the tables for which you will create OLEP models, the columns of each table, and the business models to be configured. Referenced tables cannot be deleted.

#### Prerequisites

You have created an accessible data source.

#### Context

OLEP models include the following three types of mappings: table-to-object mappings, table-to-link mappings, and table-to-event mappings. You can create objects, links, and events when you create OLEP models. Afterward, you can view and configure these objects, links, and events on the Object Information, Link Information, and Event Information pages, respectively. You can configure these items to reflect your business semantics. An OLEP table serves as a source for configuring objects, first-degree links, and events. A table can be mapped to multiple objects, links, and events.

#### Procedure

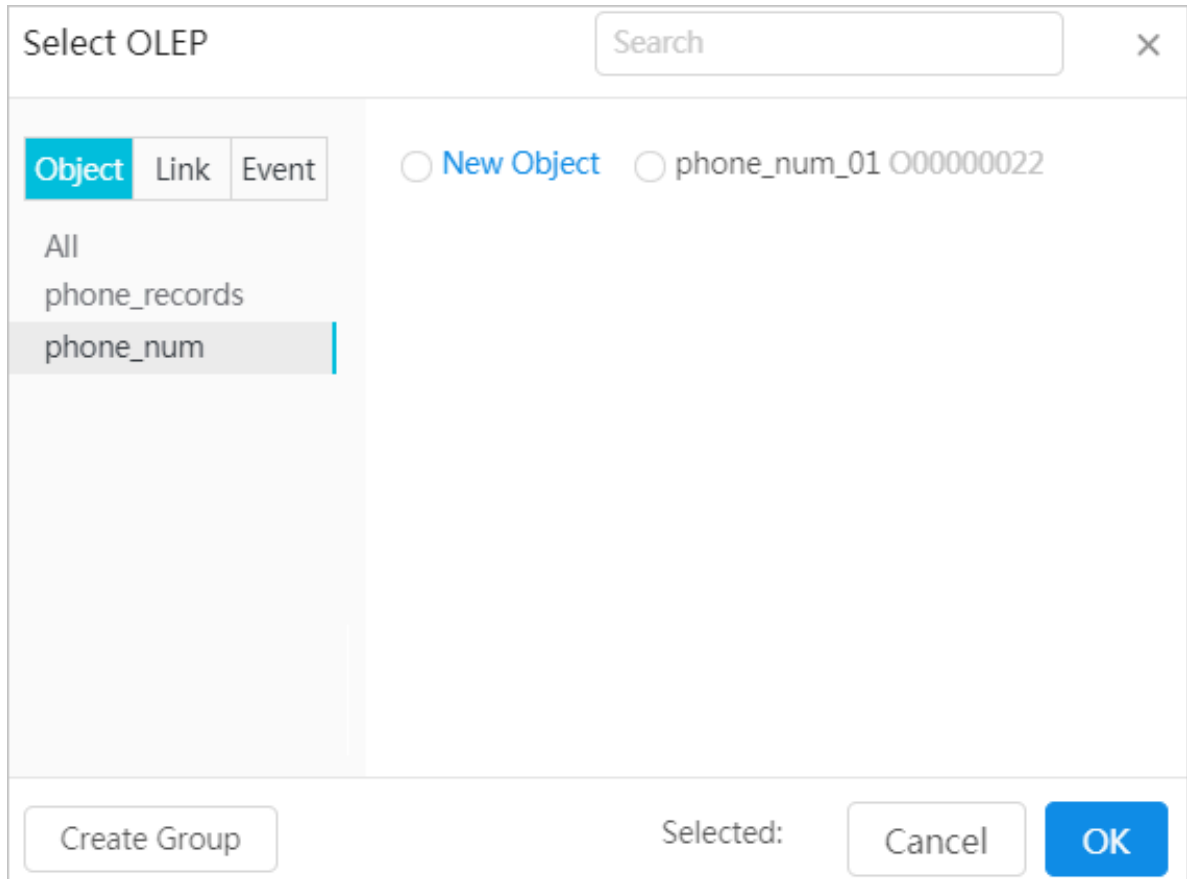
1. [Log on to Administration Console of Graph Analytics.](#)
2. Click **Source Tables** in the top navigation bar. The **Data Sources** page appears.
3. Click a data source in the left-side navigation pane. The data source details are displayed on the right side of the page.





5. Select a table, and then click Add to OLEP in the Actions column. The Select OLEP dialog box appears.

Figure 7-11: Select OLEP dialog box



The Select OLEP dialog box contains the Object, Link, and Event tabs which are used to create mappings to objects, links, and events, respectively. For more information about how to create a mapping to an object, link, or event, see [step 6](#), [step 7](#), and [step 8](#).

If there are no existing object, link, or event groups that can meet your requirements, click Create Group to create a new object, link, or event group.

**6. Map the table to an object.**

- a) **Select New Object or an existing object and then click OK. The Map to Object dialog box appears.**

Figure 7-12: Map to a newly created object

Map to Object
×

\* Object Name: 
Group:

Property ID	Table Column	* Property Name	* Primary Key	Mapping <input checked="" type="checkbox"/>
O00000027P0001	<input type="text" value="identity_card"/>	<input type="text" value="identity_card"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
O00000027P0002	<input type="text" value="name"/>	<input type="text" value="name"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
O00000027P0003	<input type="text" value="phone_num"/>	<input type="text" value="phone_num"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

<
1
>

Cancel
OK

Figure 7-13: Map to an existing object

Map to Object
×

Object Name: 
Group: 
Add Property

Property ID	Table Column	Property Name	Primary Key
O00000022P0001	<input type="text" value="identity_card"/>	<input type="text" value="identity_card"/>	<input type="checkbox"/>
O00000022P0002	<input type="text" value="name"/>	<input type="text" value="name"/>	<input type="checkbox"/>
O00000022P0003	<input type="text" value="phone_num"/>	<input type="text" value="phone_num"/>	<input checked="" type="checkbox"/>

<
1
>

Cancel
OK

**b) Set the parameters as needed.**

- **New object:** Set parameters based on [Table 7-13: Parameters used to map the table to a new object](#).
- **Existing object:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing properties of the object, you can click Add Property to add new properties.

Table 7-13: Parameters used to map the table to a new object

Feature name	Description
Object Name	The user-defined object name. It must be unique.
Group	The object group to which the object belongs. All available object groups are displayed in the drop-down list.
Name	<p>The name of an object property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, property names are displayed instead of the actual table columns that are mapped to the properties.</p>
Mapping	Whether to enable the property mapping.
Primary Key	Sets a property as a primary key. Each primary key uniquely identifies an object. You must set one or more properties as primary keys for each object. You must enable Mapping for primary keys.

**c) Click OK.**

**7. Map the table to a link.**

- a) **Click the Link tab. All first-degree links to which the current table has been mapped are displayed.**

- b) **Select New Link or an existing link and then click OK. The Map to Link dialog box appears.**

Figure 7-14: Map to a newly created link

Map to Link
×

\* Link Name: 
Group:

\* Source Object: 
\* Target Object:

**Basic Information**

Property ID	Table Column	* Property Name	Mapping <input checked="" type="checkbox"/>
L00000016P0001	<input type="text" value="callee_num"/>	<input type="text" value="callee_num"/>	<input checked="" type="checkbox"/>
L00000016P0002	<input type="text" value="caller_num"/>	<input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>

<
1
>

**Source Property Mapping**

SourceObject Property:phone\_num\_01 - phone\_num
\* Link Property:

**Target Property Mapping**

TargetObject Property:phone\_num\_01 - phone\_num
\* Link Property:

Cancel
OK

Figure 7-15: Map to an existing link

Map to Link
×

Link Name: 
Group:

Source Object: 
Target Object: 
Create Property

**Basic Information**

Property ID	Table Column	Property Name
L00000014P0001	<input type="text" value="caller_num"/>	<input type="text" value="caller_num"/>
L00000014P0002	<input type="text" value="callee_num"/>	<input type="text" value="callee_num"/>

<
1
>

**c) Set parameters as needed.**

- **New link:** Set parameters based on [Table 7-14: Parameters used to map the table to a new link](#).
- **Existing link:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing link properties, you can click Add Property to add new link properties.

Table 7-14: Parameters used to map the table to a new link

Feature name	Description
Link Name	The user-defined link name. It must be unique.
Group	The link group to which the link belongs. All available link groups are displayed in the drop-down list.
Source	The source object of the link. You can select an object from the drop-down list. The Source Property Mapping parameter is available only after you set the Source Object parameter.
Target	The target object of the link. You can select an object from the drop-down list. The Target Property Mapping parameter is available only after you set the Target Object parameter.
Name	<p>The name of an event property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, the Property Name values are displayed as the table header in Details on the Graph page.</p>
Mapping	Whether to enable the property mapping.
Link Property in Source Property Mapping	The link property to which a primary key property of the source object is mapped.

Feature name	Description
Link Property in Target Property Mapping	The link property to which a primary key property of the target object is mapped.

d) Click OK.



**8. Map the table to an event.**

- a) **Click the Event tab. All events to which the current table has been mapped are displayed.**

- b) Select New Event or an existing event and then click OK. The Map to Event dialog box appears.**

Figure 7-16: Map to a newly created event

Map to Event

\* Event Name : 
Group:

**Basic Information**

Property ID	Table Column	* Property Name	* Primary Key	Mapping
E00000014P0001	<input type="text" value="callee_num"/>	* <input type="text" value="callee_num"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
E00000014P0002	<input type="text" value="caller_num"/>	* <input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

<
1
>

^ Primary Key Mappings of Correlated Objects
Add Mapping

phone\_num O00000022

phone\_num(O00000022P0003) :

phone\_num O00000022

phone\_num(O00000022P0003) :

Cancel
OK

Figure 7-17: Map to an existing event

Map to Event

Event Name : 
Group: 
Add Property

**Basic Information**

Property ID	Physical Table Field	Property Name	Primary Key
E00000014P0001	<input type="text" value="caller_num"/>	<input type="text" value="callee_num"/>	<input type="checkbox"/>
E00000014P0002	<input type="text" value="callee_num"/>	<input type="text" value="caller_num"/>	<input checked="" type="checkbox"/>

<
1
>

**c) Set parameters as needed.**

- **New event:** Set parameters based on [Table 7-15: Parameters used to map the table to a new event](#).
- **Existing event:** Create a one-to-one mapping between each Table Column and Property Name as needed.

If the number of table columns are more than the number of existing event properties, you can click Add Property to add new event properties.

Table 7-15: Parameters used to map the table to a new event

Feature name	Description
Event Definition Name	The user-defined event name. It must be unique.
Group	The event group to which the event belongs. All available event groups are displayed in the drop-down list.
Name	<p>The name of an event property to which a table column is mapped. By default, the property name is the same as the column name. You can also define the property name as needed.</p> <p>On the Analytics Workbench, the Property Name values are displayed as the table header in Details on the Graph page.</p>
Switch	Whether to enable the property mapping.
Primary Key	Sets a property as a primary key. Each primary key uniquely identifies an event. You must set one or more properties as primary keys for each event. Switch must be turned on for the properties that are set as primary keys.

Feature name	Description
<b>Map Primary Keys to Correlated Objects</b>	<p>Indicates the mappings between the primary keys of correlated objects and the event properties. At least two correlated objects are required. You can click Add Mapping to add more necessary mappings between the primary keys of correlated objects and the event properties.</p> <p>You must enable Mapping for the event properties to which the primary keys of the correlated objects are mapped.</p>

d) Click OK.

- After you have created OLEP models for the table, click the Added to OLEP tab to check the results.

### 7.3.2.2 View an OLEP table

Select a table that you want to view from the left-side navigation pane.

The table information is displayed on the right side of the page, including table basics, and dependent objects and links, and table columns, as shown in [Figure 7-18: View a table](#).

Figure 7-18: View a table

cust\_regist\_info\_demo

Table Information

Edit

Remove

Table Name : cust\_regist\_info\_demo

Table Description :

Table Size :

Table Rows :

Created Objects : device,cust,ip

Created Links : cust\_device,cust\_ip

Created Events :

Created Dictionaries :

Added Columns

Columns Not Added

Column Name :

Column Description :

Search

All

device

cust

ip

cust\_device

cust\_ip

Column Name	Column Description	Property Name	Property ID	Column Type	String Length	Primary Key	Timestamp Format	Actions
ip_prov		--	--	string		No	None	Map
mac		mac	O00000156P0005	string		No	None	Remove
date_time		--	--	time		No	yyyy-MM-dd HH:mm:ss	Map

### 7.3.2.3 Edit OLEP tables

After you have created OLEP models for a table, you can still add new object, link, or event mappings to the table. You can also add a description for the table.

#### Prerequisites

Make sure that OLEP models have been created for a table.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click a table in the left-side navigation pane. The data source details are displayed on the right side of the page.


Add new object, link, or event mappings to tables.

3. On the Added to OLEP tab, select a table, and then click Add Mapping. The Select OLEP dialog box appears.

You can add a new object, link, or event mapping to the table. For more information, see the procedure described in [Create OLEP models for tables](#).

Modify the table description.

**4. You can use one of the following methods to modify the table description.**

Method	Operation
On the Added to OLEP page	<p>a. Click the Edit icon  next to the Table Description column. You can then modify the Table Description.</p> <p>b. Click OK to save the changes. A success message is displayed after the operation is completed.</p>
On the table details page	<p>a. Click a table in the left-side navigation pane. On the right side of the page, click Edit in the Table Information section. You can then modify the Table Description.</p> <p>b. Click OK to save the changes. A success message is displayed after the operation is completed.</p>

**7.3.2.4 Remove OLEP tables**

If a mapping between a table and an object, link, or event is no longer used, you can remove the mapping.

**Prerequisites**

Make sure that you can access the data source that stores the table.

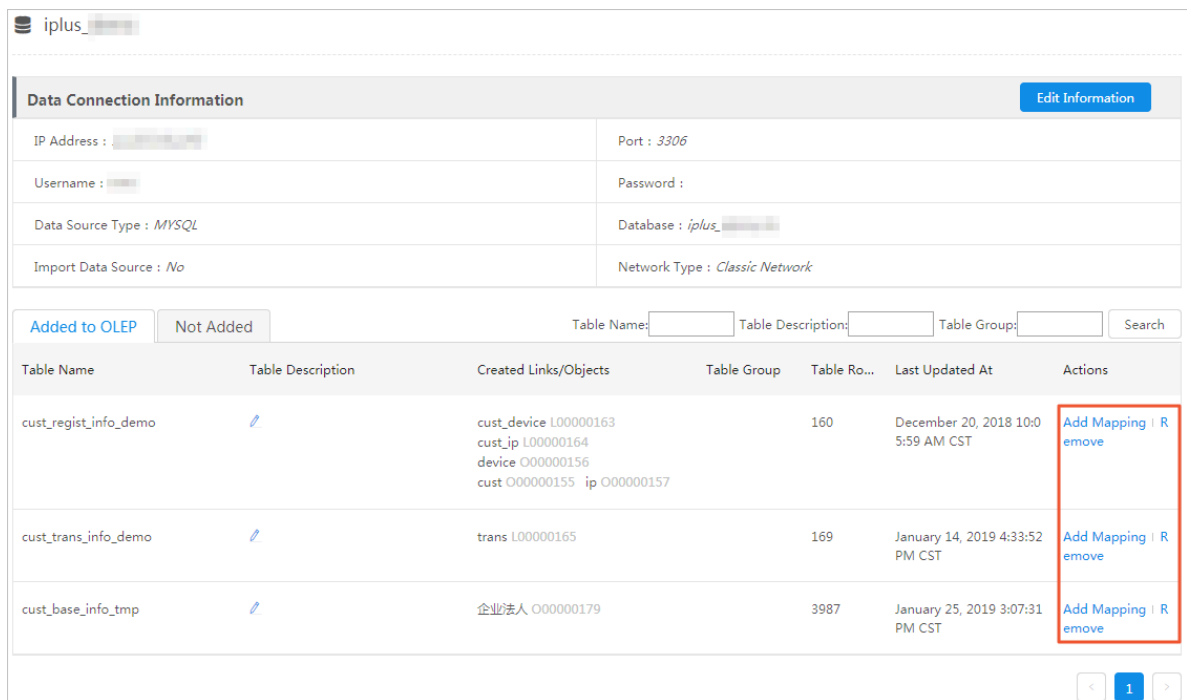
**Context**

You only remove the mappings between the tables and objects, links, and events. The objects, links, events, and tables will not be deleted.

**Procedure**

1. Click the Source Tables tab in the top navigation bar. The Data Sources page appears.
2. Click a table in the left-side navigation pane. The data source details are displayed on the right side of the page.

### 3. On the Added to OLEP tab, click Remove next to a table.



**Data Connection Information** [Edit Information](#)

IP Address :	Port : 3306
Username :	Password :
Data Source Type : MYSQL	Database : iplus_
Import Data Source : No	Network Type : Classic Network

[Added to OLEP](#) [Not Added](#) Table Names: Table Description: Table Group: Search

Table Name	Table Description	Created Links/Objects	Table Group	Table Ro...	Last Updated At	Actions
cust_regist_info_demo		cust_device L00000163 cust_ip L00000164 device O00000156 cust O00000155 ip O00000157		160	December 20, 2018 10:05:59 AM CST	<a href="#">Add Mapping</a>   <a href="#">Remove</a>
cust_trans_info_demo		trans L00000165		169	January 14, 2019 4:33:52 PM CST	<a href="#">Add Mapping</a>   <a href="#">Remove</a>
cust_base_info_tmp		企业法人 O00000179		3987	January 25, 2019 3:07:31 PM CST	<a href="#">Add Mapping</a>   <a href="#">Remove</a>

< 1 >

### 4. In the Select OLEP dialog box, select an Object, Link, or Event mapping. You can select only one mapping at one time.

### 5. Click OK.

If all mappings are removed from a table, the table will be automatically moved from the Added to OLEP tab to the Not Added tab.

## 7.3.3 OLEP table columns

### 7.3.3.1 Add OLEP table columns

If a data table has been mapped to an object, link, or event, and the table still has unoccupied columns (columns that are not correlated with any object, link, or event), you can add these columns to the existing mappings as needed.

#### Prerequisites

A data table has been mapped to an object, link, or event, but the table still has unoccupied columns.

#### Context

Before you configure the OLEP table columns, sort out the columns for which you will create OLEP models and data types of the columns, especially the time columns

.

## Procedure

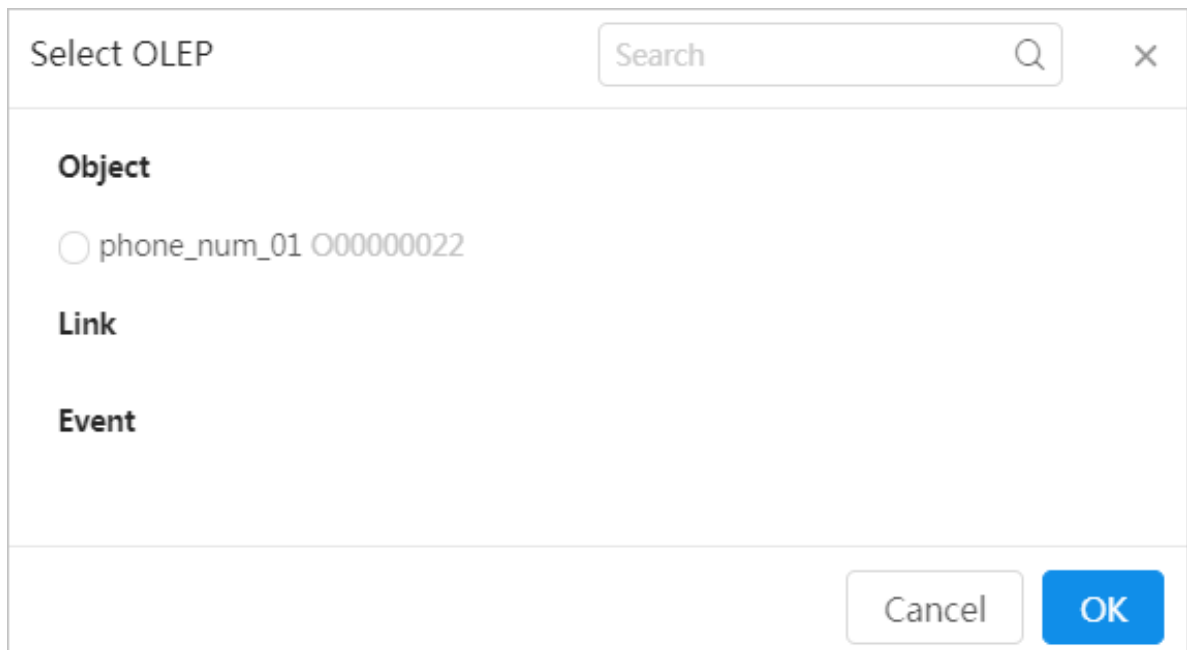
1. [Log on to Administration Console of Graph Analytics.](#)
2. Click **Source Tables** in the top navigation bar.
3. On the **Data Sources** page, click a data source in the left-side navigation pane, and then click the table to which you want to add the columns.
4. Click the **Columns Not Added** tab in the right-side area. In the **Columns Not Added** tab that appears, click **Add** in the **Actions** column.

Columns displayed in the **Columns Not Added** tab are not mapped to any object, link, or event.

5. In the **Select OLEP** dialog box that appears, select the object, link, or event to which the columns map, and then click **OK**.

The **Select OLEP** tab only displays the objects, links, and events that have mapped to the current data table.

The following example describes how to add columns to an object.



Select OLEP

Search

**Object**

☐ phone\_num\_01 000000022

**Link**

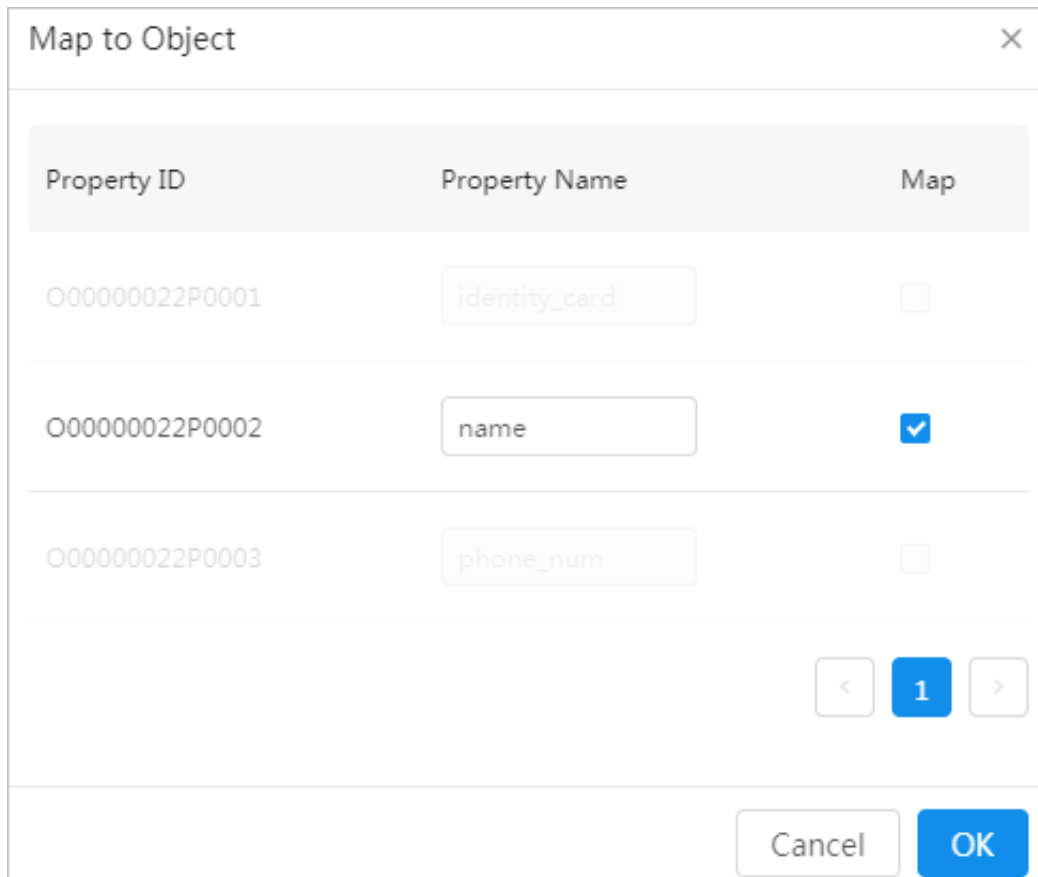
**Event**

Cancel OK



6. In the Map to Object dialog box that appears, select the object properties to be mapped to the columns to be added.

Object properties that have been mapped to the current data table are gray and cannot be operated.



The 'Map to Object' dialog box displays a table with three columns: Property ID, Property Name, and Map. The first row shows 'O00000022P0001' with 'identity\_card' in a disabled text field and an unchecked checkbox. The second row shows 'O00000022P0002' with 'name' in a text field and a checked checkbox. The third row shows 'O00000022P0003' with 'phone\_num' in a disabled text field and an unchecked checkbox. At the bottom right are navigation buttons '<', '1', and '>'. At the bottom center are 'Cancel' and 'OK' buttons.

Property ID	Property Name	Map
O00000022P0001	identity_card	<input type="checkbox"/>
O00000022P0002	name	<input checked="" type="checkbox"/>
O00000022P0003	phone_num	<input type="checkbox"/>

< 1 >

Cancel OK

7. After you have configured the preceding parameters, click OK.

### 7.3.3.2 Edit OLEP table columns

After the data table is mapped to objects, links, or events, you can modify the Column Description and Timestamp Format of added columns.

#### Prerequisites

A data table has been mapped to an object, link, or event.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click Source Tables in the top navigation bar.

3. On the Data Sources page, click a data source in the left-side navigation pane, and then click the table that contains the columns to be modified.

After you select the table, the details of the table will be displayed on the right side of the Data Sources page.

4. Click the Added Columns tab.

By default, the Added Columns tab displays all columns that have been added.

cust\_regist\_info\_demo

Table Information

Edit

Remove

Table Name : cust\_regist\_info\_demo

Table Description :

Table Size :

Table Rows :

Created Objects : device,cust,ip

Created Links : cust\_device,cust\_ip

Created Events :

Created Dictionaries :

Added Columns

Columns Not Added

Column Name :

Column Description :

Search

All

device

cust

ip

cust\_device

cust\_ip

Column Name	Column Description	Property Name	Property ID	Column Type	String Length	Primary Key	Timestamp Format	Actions
ip_prov		ip_prov	O00000157P0004	string		No	None	<div>Edit</div>
mac		mac mac	O00000156P0005 L00000163P0005	string		No	None	<div>Edit</div>
date_time		date_time date_time	L00000163P0002 L00000164P0002	time		No	yyyy-MM-dd HH:mm:ss	<div>Edit</div>
ip		ip ip	O00000157P0003 L00000164P0003	string		No	None	<div>Edit</div>

**5. On the All tab page, click Edit in the Actions column to modify the Column Description and Timestamp Format of a column.**

Added Columns		Columns Not Added		Column Name : <input type="text"/> Column Description : <input type="text"/> Search <input type="text"/>				
All	device	cust	ip	cust_device	cust_ip			
Column Name	Column Description	Property Name	Property ID	Column Type	String Length	Primary Key	Timestamp Format	Actions
ip_prov	<input type="text"/>	ip_prov	O00000157P0004	string		No	None	Save
mac		mac mac	O00000156P0005 L00000163P0005	string		No	None	Edit
date_time		date_time date_time	L00000163P0002 L00000164P0002	time		No	yyyy-MM-dd HH:mm:ss	Edit
ip		ip ip	O00000157P0003 L00000164P0003	string		No	yyyy-MM-dd	Edit
cust_id	cust_id	cust_id cust_id cust_id	O00000155P0001 L00000163P0001 L00000164P0001	string		No	yyyy/mm/dd	Edit

**The Timestamp option in the Timestamp Format drop-down list refers to the UNIX timestamp.**

**6. After you have configured the preceding parameters, click Save.**

### 7.3.3.3 Remove OLEP table columns

After you have created OLEP models for a table, you can remove any unnecessary OLEP mappings of specific columns in the table.

#### Prerequisites

**Referenced columns cannot be deleted.**

#### Context

After a table has been mapped to objects, links, or events, you cannot separately remove the following mappings of a column:

- **Table-to-object mappings:** The mapping between the column and a primary key property of the object cannot be removed.
- **Table-to-link mappings:** The mapping between the column and a correlated property of the link cannot be removed.
- **Table-to-event mappings:** The mapping between the column and a primary key property of the event cannot be removed.

To remove these mappings, remove the corresponding OLEP mapping from the table and then add OLEP mappings to the table again. For more information, see [Remove OLEP tables](#) and [Create OLEP models for tables](#).

## Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click **Source Tables** in the top navigation bar.
3. On the **Data Sources** page, click a data source in the left-side navigation pane, and then click the table that contains the columns to be removed.
4. On the right side of the **Data Sources** page, click the **Added Columns** tab.

By default, the **Added Columns** tab displays all columns that have been added.

5. On the **Added Columns** tab, click the tab of an object, link, or event that has been mapped to a data table.

On the selected object, link, or event page, the table columns that have been mapped are highlighted.

<

6. Select a column, and then click **Remove** to remove the object, link, or event mapping.

If all mappings of a column are removed, including object, link, and event mappings, the column will be automatically moved to the **Columns Not Added** tab.

## 7.4 Dictionaries

### 7.4.1 Create a dictionary

Before you create a dictionary, sort out the columns to be converted from the system data. If a dictionary has been referenced, it cannot be deleted.

#### Prerequisites

Before you create a dictionary, complete the following tasks:

- Make sure that you have created a data source. For more information, see [Create data sources](#).
- Make sure that you have configured mappings between tables and objects, links, or events. For more information, see [Create OLEP models for tables](#).
- Map table columns to OLEP. For more information, see [Add OLEP table columns](#).

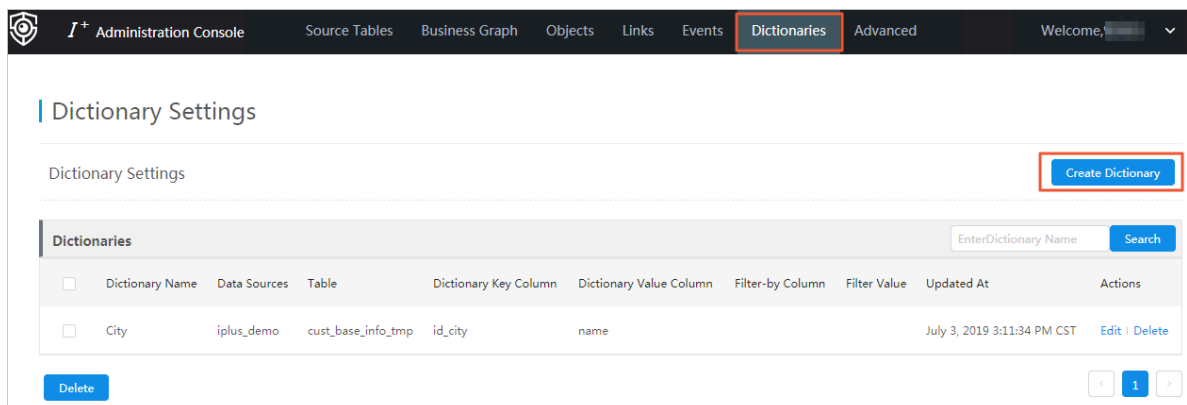
#### Context

A dictionary is used to map a specific column in the table to a value column and display the value column name in Analytics Workbench instead of the original column name.

For example, in the Graph area of Analytics Workbench, column A (company code) is displayed. After column A is escaped to column B (company name), the company name corresponding to the code will be displayed in the Graph area.

#### Procedures

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Dictionaries.



3. On the Dictionary Settings page, click **Create Dictionary** in the upper-right corner of the page.

Dictionary Settings

×

\* Dictionary Name:

Enter a dictionary name.

\* Data Sources:

▼

\* Table:

▼

\* Key Column:

▼

?

\* Dictionary Value ..

▼

?

Filter-by Column:

▼

?

Filter Operator:

▼

Filter Value:

Enter a filter value.

?

Cancel

OK

4. In the Dictionary Settings dialog box that appears, specify the parameters as needed.

These parameters are described as follows:

Table 7-16: Parameter configurations for adding a dictionary

Parameter	Description
Dictionary Name	The name of the dictionary. The user can customize the name as needed.
Data Sources	The data sources to be referenced.
Table	The table in the data source to be referenced.

Parameter	Description
Key Column	The column that stores the dictionary code in the selected table.
Dictionary Value Column	The value column corresponding to the converted dictionary.
Filter-by Column	These three parameters are not required and are used to filter dictionary tables based on different conditions. If you specify any of the three parameters, the other two parameters are required.
Filter Operator	
Filter Value	

5. After you have configured the preceding parameters, click OK.

## 7.4.2 Modify a dictionary

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click **Dictionaries**. The **Dictionary Settings** page appears. All existing dictionaries are displayed.

3. Select a dictionary and then click Edit, as shown in [Figure 7-19: Modify a dictionary](#).

Figure 7-19: Modify a dictionary

Dictionary Settings

\* Dictionary Name: City

\* Data Sources: iplus\_demo

\* Table: cust\_base\_info\_tmp

\* Key Column: id\_city

\* Dictionary Value .. name

Filter-by Column:

Filter Operator:

Filter Value: Enter a filter value.

Cancel OK

4. In the dialog box that appears, modify the parameters as needed. For more information about dictionary parameters, see [Parameter description](#).
5. Click OK.

### 7.4.3 Delete a dictionary

**You cannot delete dictionaries that have been referenced.**

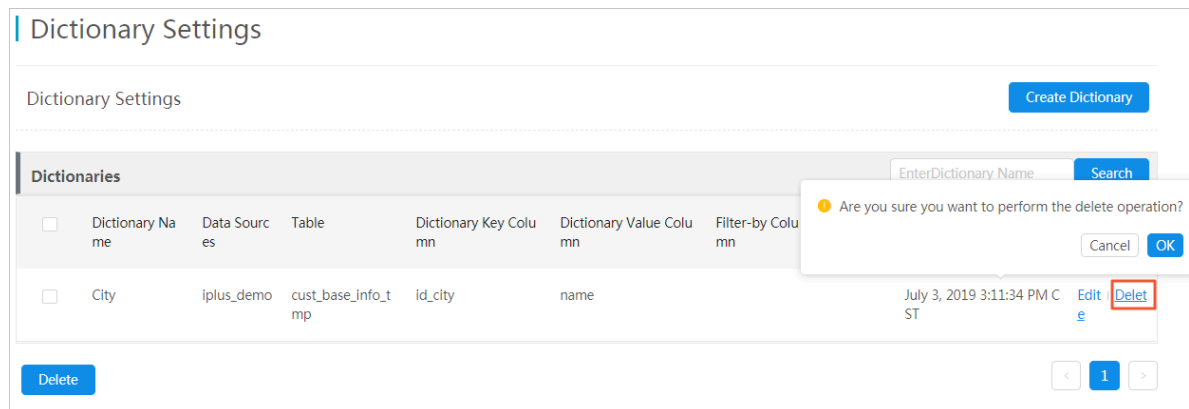
Delete a dictionary

1. [Log on to Administration Console of Graph Analytics](#).



2. In the top navigation bar, click Dictionaries. The Dictionary Settings page appears. All existing dictionaries are displayed.
3. Select a dictionary, and then click Delete. A confirm message appears, as shown in [Figure 7-20: Delete a dictionary](#).

Figure 7-20: Delete a dictionary

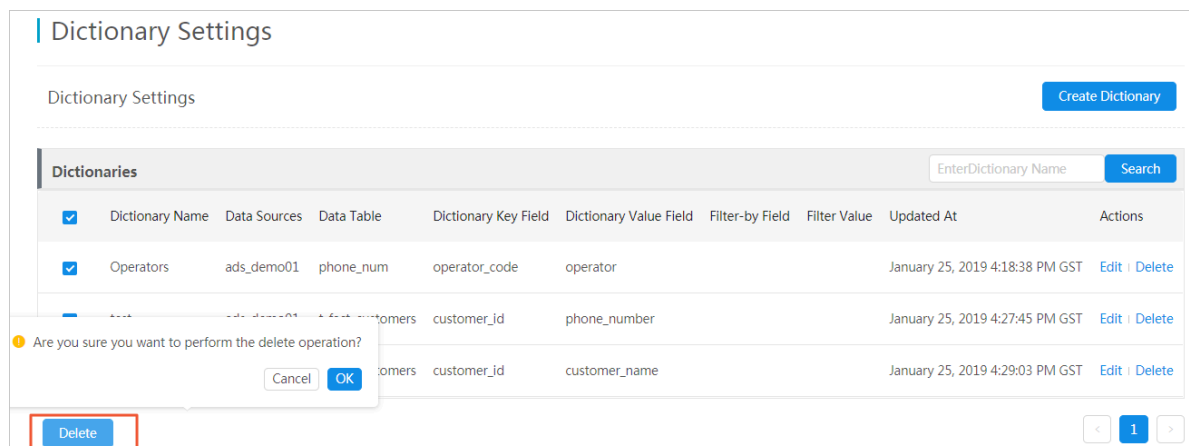


4. Click OK.

Delete multiple dictionaries at one time

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Dictionaries. The Dictionary Settings page appears. All existing dictionaries are displayed.
3. Select one or more dictionaries, and then click Delete in the lower-left corner. A confirm message appears, as shown in [Figure 7-21: Delete multiple dictionaries at one time](#).

Figure 7-21: Delete multiple dictionaries at one time



4. Click OK.

## 7.5 Object information

### 7.5.1 Object groups

#### 7.5.1.1 Create an object group

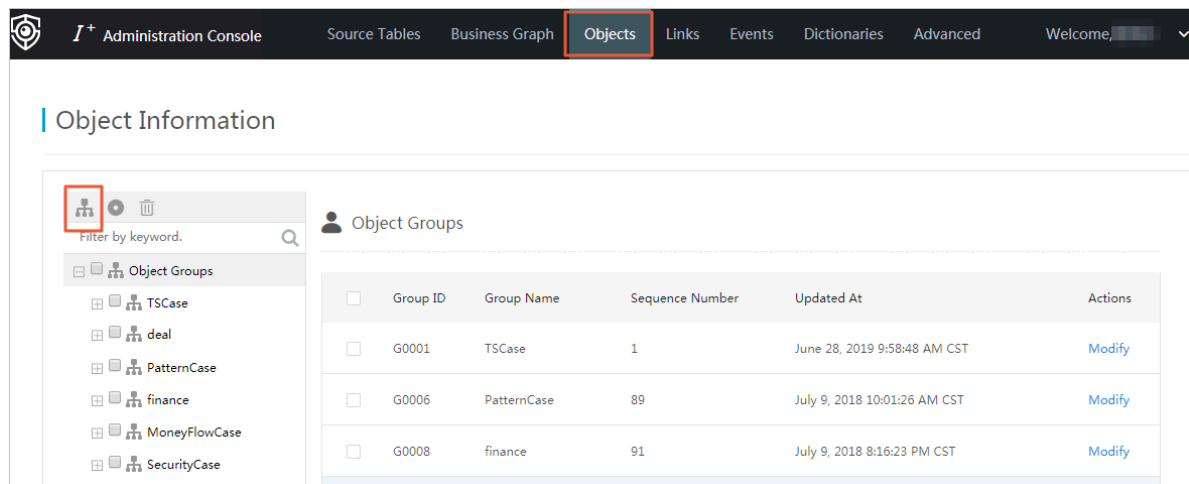
You can use object groups to classify objects, so that you can search for and manage objects with ease. Any object must be and can only be grouped into one object group. You need to create a proper object group before you create an object.

#### Prerequisites

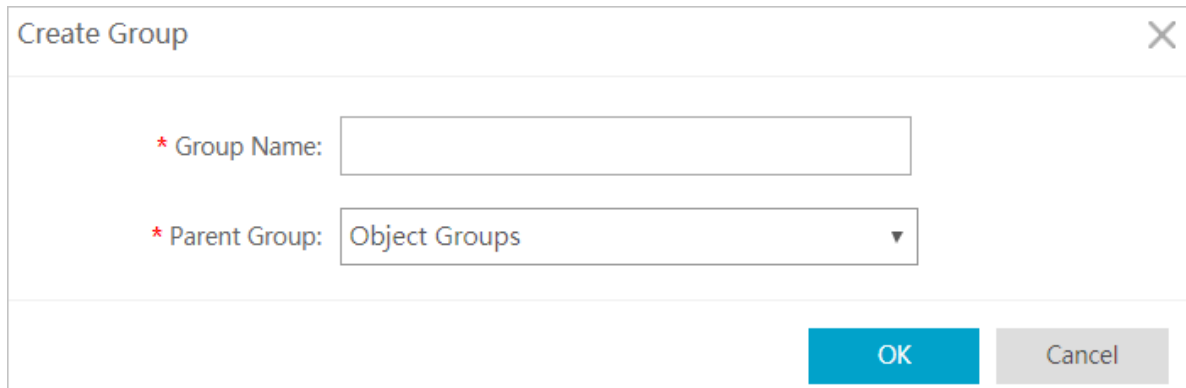
Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click **Objects**.
3. Click the **Create Group** icon  to create a new group.



4. In the Create Group dialog box that appears, specify the Group Name and the Parent Group.



The 'Create Group' dialog box contains two required fields: 'Group Name' (a text input field) and 'Parent Group' (a dropdown menu with 'Object Groups' selected). At the bottom right are 'OK' and 'Cancel' buttons.

5. Click OK.

### 7.5.1.2 View object groups and objects

In Graph Analytics, you can view all object groups in the current environment. You can understand the existing object groups and the object information under each group at any time.

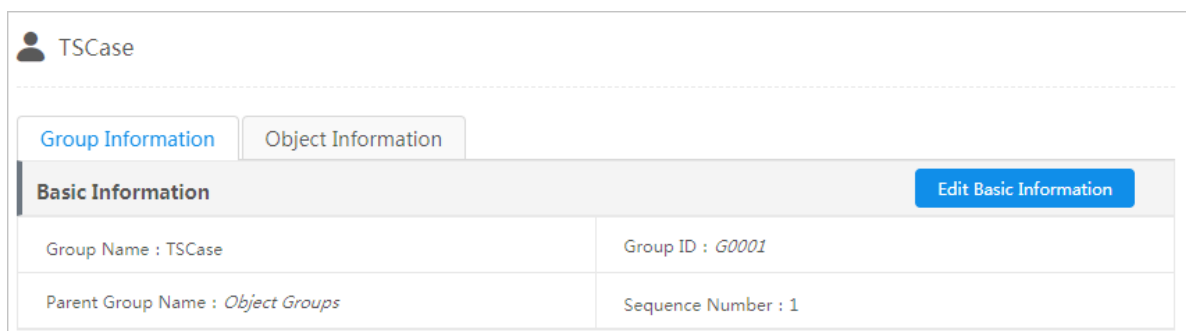
#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Objects.
3. On the Object Information page, click an object group in the left-side navigation pane. The Group Information and Object Information of the group are displayed on the right side of the page.

The left-side navigation pane displays all the object groups in the current environment. You can view the groups one by one.



The 'Object Information' page for the 'TSCase' group shows two tabs: 'Group Information' (active) and 'Object Information'. Under 'Group Information', there is a 'Basic Information' section with an 'Edit Basic Information' button. The information is displayed in a table:

Group Name : TSCase	Group ID : G0001
Parent Group Name : Object Groups	Sequence Number : 1

### 7.5.1.3 Modify object groups and objects

In Graph Analytics, you can modify the name and order number of an object or object group. You can adjust the basic information of an object or object group at any time as needed.

#### Prerequisites

- You have created an object group. For more information about how to create an object group, see [Create an object group](#).
- You have created an object that belongs to this object group. For more information about how to create an object, see [Create an object](#).
- To disable objects, you must first delete the dependency information of these objects, including the mappings between the objects and data tables.

#### Context

This topic describes the following operations:

- Modify the basic information about an object group
- Modify the basic information about an object
- Disable and enable an object

After you have created a complete object (configured the object properties), the object is enabled automatically. You can disable an object if it is no longer used for a certain period of time and enable it again when necessary. You cannot use an object in Analytics Workbench after the object is disabled.

#### Modify the basic information about an object

You can modify the basic information of an object group by using the following two methods.

Method	Procedure
Method one	<ol style="list-style-type: none"><li>1. <a href="#">Log on to Administration Console of Graph Analytics</a>.</li><li>2. In the top navigation bar, click Objects.</li><li>3. In the Object Groups area, select an object group and click Modify in the Actions column.</li><li>4. In the Edit Information dialog box that appears, reset the Group Name and Sequence Number, as shown in <a href="#">Figure 7-22: Edit information</a>.</li><li>5. Click OK.</li></ol>

Method	Procedure
<b>Method two</b>	<ol style="list-style-type: none"> <li>1. <i>Log on to Administration Console of Graph Analytics.</i></li> <li>2. <b>In the top navigation bar, click Objects.</b></li> <li>3. <b>In the left-side navigation pane, click the object group to be modified.</b></li> <li>4. <b>On the Group Information tab, click Edit Basic Information. Reset the Group Name and the Sequence Number, as shown in <i>Figure 7-23: Edit basic information.</i></b></li> <li>5. <b>Click Save.</b></li> </ol>

Figure 7-22: Edit information

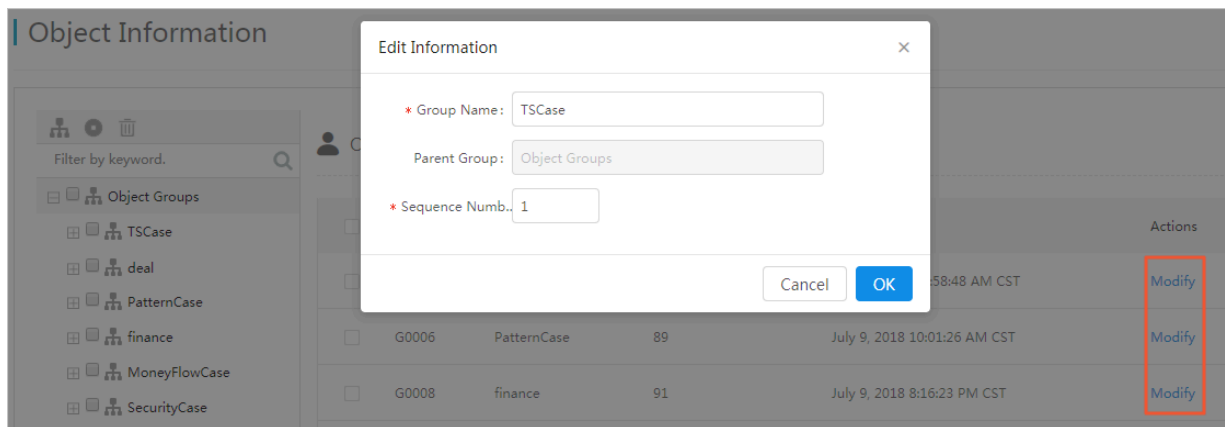
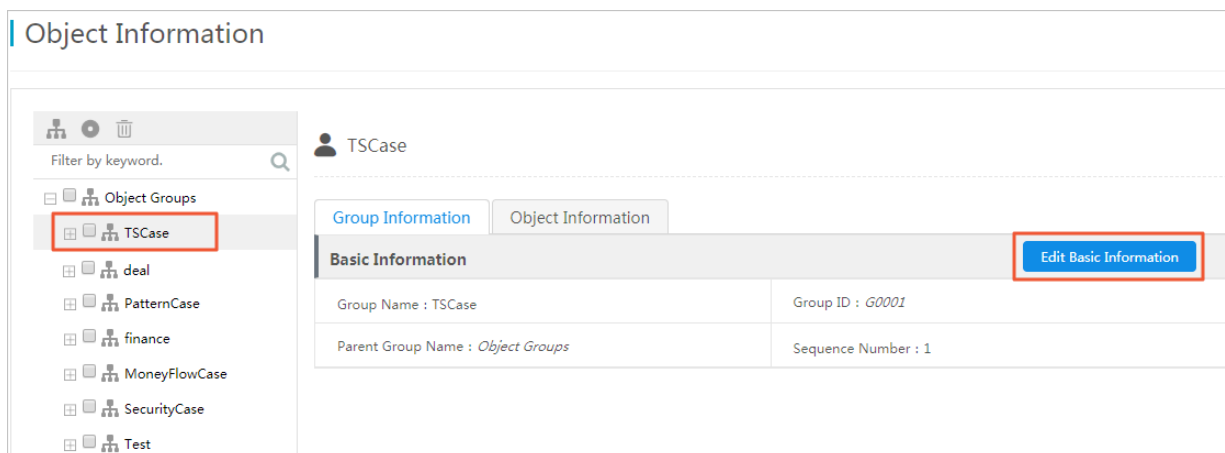


Figure 7-23: Edit basic information









### Modify basic object information

1. *Log on to Administration Console of Graph Analytics.*
2. **In the top navigation bar, click Objects.**

3. In the left-side navigation pane, click the object group to which the object belongs, and then click the Object Information tab on the right side of the page.
4. On the Object Information tab, select an object and then click Edit in the Actions column.
5. In the Edit Information dialog box that appears, reset the Object Name and the Sequence Number.
6. Click OK.

Disable and enable an object

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Objects.
3. In the left-side navigation pane, click the object group to which the object belongs, and then click the Object Information tab on the right side of the page.
4. You can use the following method to enable or disable an object on the Object Information tab.

Method	Procedure
Disable an object	<p>Disable an object individually: Click the  icon next to the object to be disabled.</p> <p>Disable multiple objects: Select the objects to be disabled and then click Off at the bottom of the page.</p> <p>After the object is disabled, the Status changes from a highlighted  icon to a gray  icon.</p>
Enable an object	<p>Enable an object individually: Click the  icon next to the object to be enabled.</p> <p>Enable multiple objects: Select the objects to be enabled and then click On.</p> <p>After the object is enabled, the Status changes from a gray  icon to a highlighted  icon.</p>



### 7.5.1.4 Delete object groups and objects

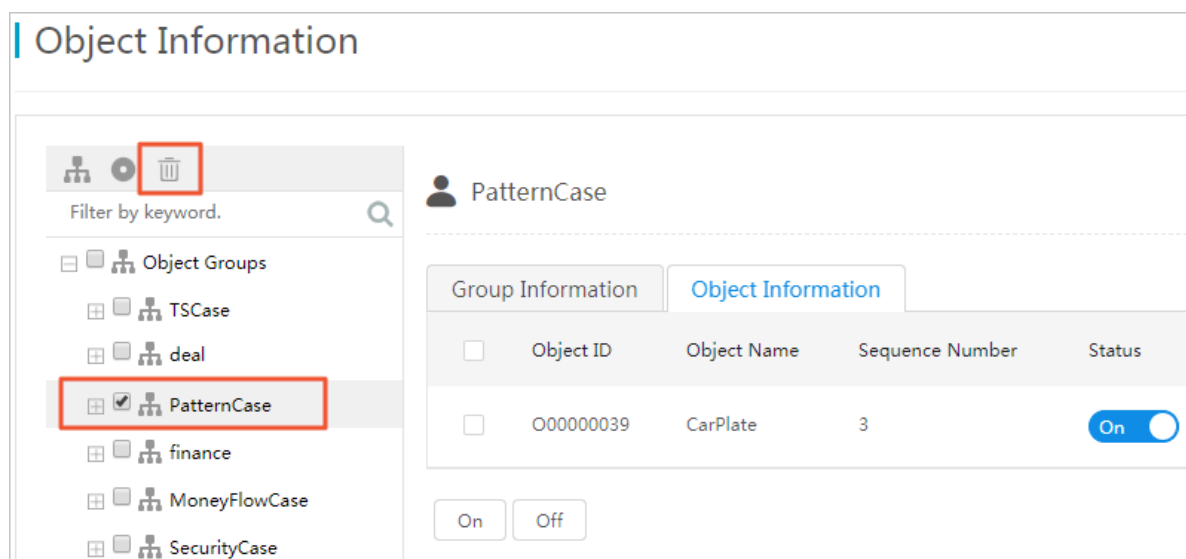
When some objects or object groups are no longer used, you can delete these objects and object groups.

#### Prerequisites

- Before you delete an object, you must delete the dependency information of the object, for example, the mapping between the object and the physical table.
- Before you delete an object group, you must delete all the objects in the group. Currently, you can only delete empty object groups.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Objects.
3. Optional: If there are still objects in the object group, you must delete these objects first.
  - a. In the left-side navigation pane, select all objects in the object group to be deleted, and click the  icon in the upper-left corner.
  - b. In the Delete Object Information dialog box that appears, click OK to clear the object group.
4. In the left-side navigation pane, select the object group to be deleted and then click the  icon.



5. In the Delete Object Information dialog box that appears, click OK.

## 7.5.2 Objects

### 7.5.2.1 Create an object

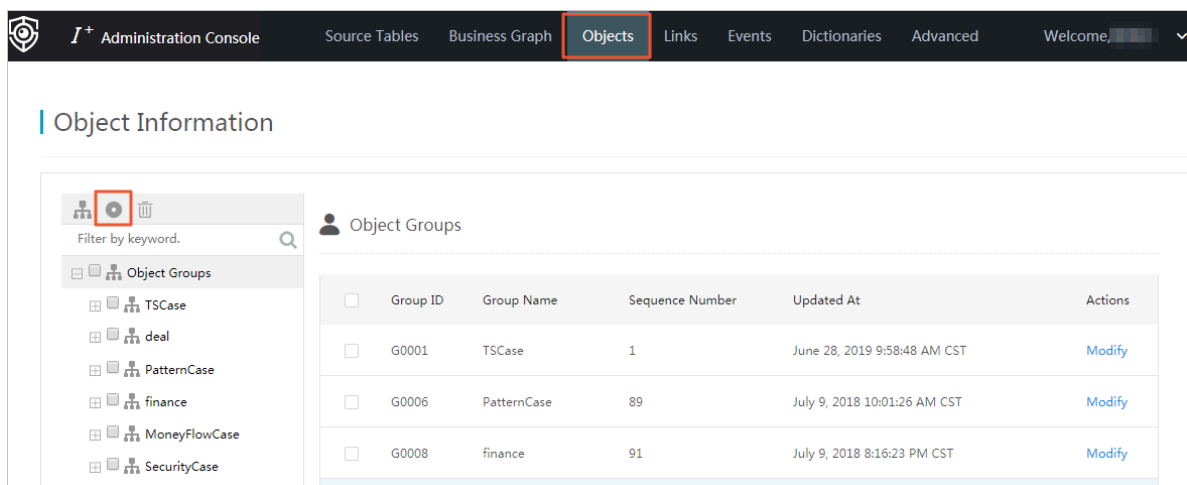
In Graph Analytics, objects are mapped to entities in the real world. Before you perform a relationship analysis on an entity, you must create an object corresponding to the entity based on the data you have obtained. A complete object contains the basic information, property information, and relevant parameters. This topic describes how to configure the basic information of an object.

#### Prerequisites

Before you create an object, make sure that you have created an object group. For more information, see [Create an object group](#).

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Objects.



3. On the Object Information page, click the Create Object icon.
4. Configure the object information in the Create Object dialog box that appears.

The parameter configurations are described in [Table 7-17: Parameters of created objects](#).

Table 7-17: Parameters of created objects

Parameter	Description
Object Name	The user-defined object name. It must be unique.
Object Description	Enter the description of the object, so the users can understand the object with ease.



Parameter	Description
Group	The selected object group in the left-side navigation pane is used by default, but you can select another group as needed.
Add in Graph	Specifies whether you are allowed to manually add the object node to the graph page.
Object Icon Display Position	Valid values are as follows: <ul style="list-style-type: none"><li>• Use Icon in Graph, Not Show Image on Right-side Pane</li><li>• Use Icon in Graph, Show Image on Right-side Pane</li><li>• Use Image in Graph, Not Show Image on Right-side Pane</li><li>• Use Image in Graph, Show Image on Right-side Pane</li></ul>
Allow Table Mapping	Yes is selected by default.
Object Icon	Sets the icon of the object. You can select an icon in Icon Library or enter a URL to reference an external icon.

5. Click OK.

## What's next

1. After you have created an object, you must configure the properties and business parameters based on your business requirements. For more information, see [Configure object properties and business parameters](#).
2. After you have configured the properties and business parameters of an object, you must log on to Analytics Workbench again to use the new object.

### 7.5.2.2 Configure object properties and business parameters

After you add an object, you need to configure the business parameters of the object based on your requirements so that you can view and analyze the object in Analytics Workbench.

## Prerequisites

- Make sure that you have created a data source. For more information, see [Create data sources](#).
- Make sure that you have configured mappings between tables and objects, links, or events. For more information, see [Create OLEP models for tables](#).

## Procedure

1. [Log on to Administration Console of Graph Analytics](#).

2. Click Objects on the top of the page.
3. In the left-side navigation pane of the Object Information page, click the name of the object to be configured and then click the Property Information tab on the right side.

If this object has been mapped to a physical table in the data source, the property information section displays the Data Source and Table information.

SV\_Account On

Object Information Property Information

Data Source: skyview Table: demo\_tranx\_id Save

Basic Information Add Row

Property ID	* Property Name	* Primary Key	* Show in Graph	Show in Properties	Conditional Query	Show in Statistics	Available	* Display Type	* Query Type	* Security Level	Actions
O00000072P0001	TRANX_ID	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	
O00000072P0002	BRANCH_NAM	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	
O00000072P0003	NAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal	S1	

Advanced

Add Image To : Logical Relation of Authorized Properties :

Location Settings Add Location

Trajectory Settings Add Trajectory

4. Set the basic configurations in the Basic Information page.


Click Add Row to add a property for a link in Basic information.

The configuration items in Basic Information are described in [Table 7-18: Description of basic configuration items](#).

Table 7-18: Description of basic configuration items

Configuration item	Description
Property ID	The ID of a property. It is automatically generated.
Property Name	The property name that is displayed on Analytics Workbench. Enter a name that describes your business.

Configuration item	Description
Primary Key	<p>You must select at least one primary key. After you complete the configuration, it cannot be modified or deleted.</p> <p>When you add a node in Analytics Workbench, you need to enter the physical table column mapped by the primary key property. For example, if the ID card number is the primary key, you need to enter the ID card number when you add an ID card node in Analytics Workbench.</p>
Show in Graph	<p>If you select this parameter, the Property Name of this property will be displayed in Graph in Analytics Workbench together with the object node. Otherwise, the property is not displayed. For example, if you select the ID card number, this number will be displayed in Graph together with the ID card object. If you select the name property at the same time, the name and the ID card number will be displayed together with the ID card object.</p> <p>If you select this parameter for a specific property, a bubble icon appears next to the option. You can click the bubble icon to set whether to show the Property Name in Graph. For example, if you select this parameter for the ID card number, and set to display the property name in Graph, the ID card property displayed in Graph will be ID card number: xxxxxx.</p>
Show in Properties	<p>If you select this parameter for a property, the property will be displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details tab and the Property tab. Otherwise, the property is not displayed.</p>
Conditional Query:	<p>If you select this parameter for a property, you can query the object based on this property in Target Object when you perform an analysis on the Graph page.</p>
Show in Statistics	<p>If you select this option for a property, the property will be displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; Statistics. Otherwise, the property is not displayed.</p>

Configuration item	Description
Available	<p>If you select this parameter for a property, the property takes effect and can be displayed in Analytics Workbench. This parameter must be selected for primary key properties.</p> <p>If any of the following parameters has been selected for the property: Primary Key, Show in Graph, Show in Properties, Conditional Query and Show in Statistics, the Available parameter is automatically selected for a property. The Available parameter is automatically deselected if you deselect all the preceding parameters.</p>
Display Type	<p>After you set the display type, the property is displayed in Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details tab and the Property tab based on the selected type.</p> <div>  <b>Note:</b> To display a property in the format of Dictionary, you need to configure a dictionary first.         </div>
Query Type	The data type that is supported in the query condition of a property. For Display Type, if you select Dictionary, you must select Dictionary Option for Query Type.
Security Level	The security level for a property. A user with a lower security level cannot view the property.
Search Item Configuration	Associates this property with a search item so that the object can be searched by this property in Analytics Workbench. For more information about search item settings, see <a href="#">Configure a search item</a> .
Default Query Condition Settings	Defines the default condition used for an object query. If other properties are used as conditions for a query, this condition is also included by default.
Authorization Code	After the authorization code function has been enabled, only users with the required authorization code can access this property.
Derived Property	Sets a property as a derived property so that it can be generated automatically based on other properties. You can set the derivative method based on your requirements.

5. **Optional:** If you need to add multiple properties, you can refer to the preceding steps to add more properties.
6. **Optional:** Set the configurations in Advanced.

The configuration information of Advanced is described in [Table 7-19: Description of advanced configuration items](#).

Table 7-19: Description of advanced configuration items

Configuration item	Description
Add Image To	Specifies the avatar of the object that is displayed in Graph. Select a property of the object, and then set the URL of the image and the suffix of the image. Add Image To contains the prefix, the property, and the suffix. The prefix is the URL of the image, and the suffix is the image format.
Logical Relation of Authorized Properties	<p>The logical relationship between the authorization codes of properties in each record:</p> <ul style="list-style-type: none"><li>• <b>AND:</b> The current record is visible only to the users who meet all authorization code conditions of the properties in this record.</li><li>• <b>OR:</b> The current record is visible to the users who meet any one authorization code condition of the properties in this record.</li></ul>

7. Click Save.

### 7.5.2.3 Enable and disable an object

After you have created a complete object (configured the object properties), the object is enabled automatically. You can disable an object if it is no longer used for a certain period of time and enable it again when necessary.

#### Prerequisites

To disable objects, you must first delete the dependency information of these objects, including the mappings between the objects and data tables and the objects referenced by links.

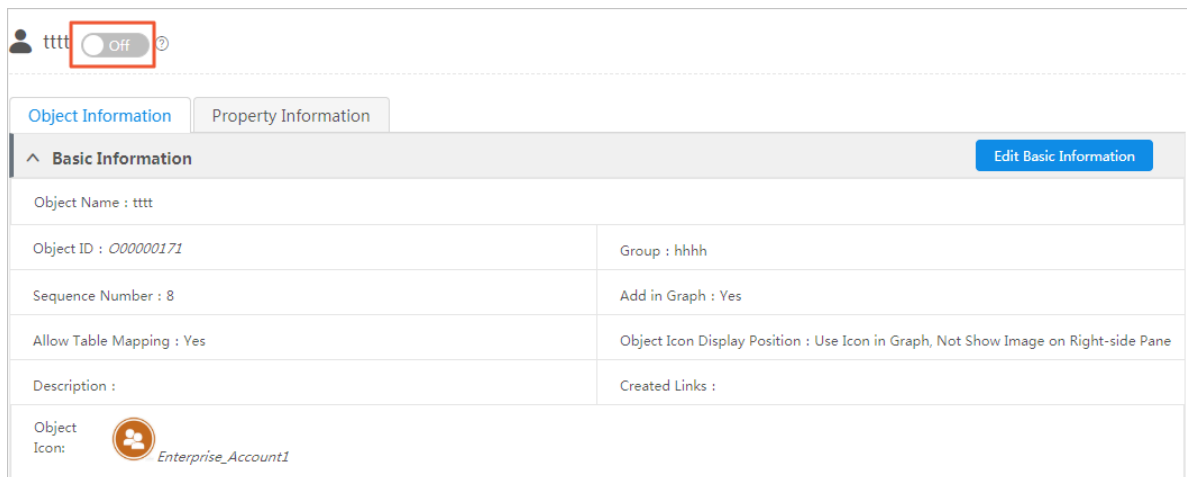
#### Context

You cannot use an object in Analyitcs Workbench after the object is disabled.


#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Objects.
3. In the left-side navigation pane, click the object group to which the object belongs, and then click the object to be enabled or disabled.







The detailed information of the object is displayed on the right side of the page.



The screenshot shows the 'Basic Information' tab for an object named 'tttt'. The status is 'Off'. The interface includes tabs for 'Object Information' and 'Property Information', and a table with details like Object ID, Sequence Number, and Group.

Basic Information		Edit Basic Information
Object Name : tttt		
Object ID : 000000171	Group : hhhh	
Sequence Number : 8	Add in Graph : Yes	
Allow Table Mapping : Yes	Object Icon Display Position : Use Icon in Graph, Not Show Image on Right-side Pane	
Description :	Created Links :	
Object Icon:  Enterprise_Account1		

4. Enable or disable an object as follows.

Operation	Procedure
Disable an object	<p>If the current object is enabled, you can click the  icon to disable the object.</p> <p>After the object is disabled, the Status changes from a highlighted  icon to a gray  icon.</p>
Enable an object	<p>If the current object is disabled, you can click the  icon to enable the object.</p> <p>After the object is enabled, the Status changes from a gray  icon to a highlighted  icon.</p>

#### 7.5.2.4 Modify an object

In Graph Analytics, you can modify the basic information of an object at any time, including the object name, object group, and object icon.


##### Prerequisites

You have created an object. For more information about how to create an object, see [Create an object](#).

## Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click **Objects**.
3. In the left-side navigation pane, click the object group to which the object belongs, and then click the object to be modified.
4. In right-side area, click **Edit Basic Information**.

The screenshot shows the 'Basic Information' tab for the 'SV\_Account' object. The 'Edit Basic Information' button is highlighted with a red box. The form contains the following fields:

Object Name : SV_Account	
Object ID : 000000072	Group : finance
Sequence Number : 1	Add in Graph : Yes
Allow Table Mapping : Yes	Object Icon Display Position : Use Image in Graph, Not Show Image on Right-side Pane
Description : SV_Account	Created Links : SV_Transfer_Link
Object Icon:  BankCard	

5. Modify the object information as needed.

All parameters can be modified except the Object ID.

6. After you have configured the preceding parameters, click **Save**.

### 7.5.2.5 Delete an object

You can delete the objects that are no longer used.

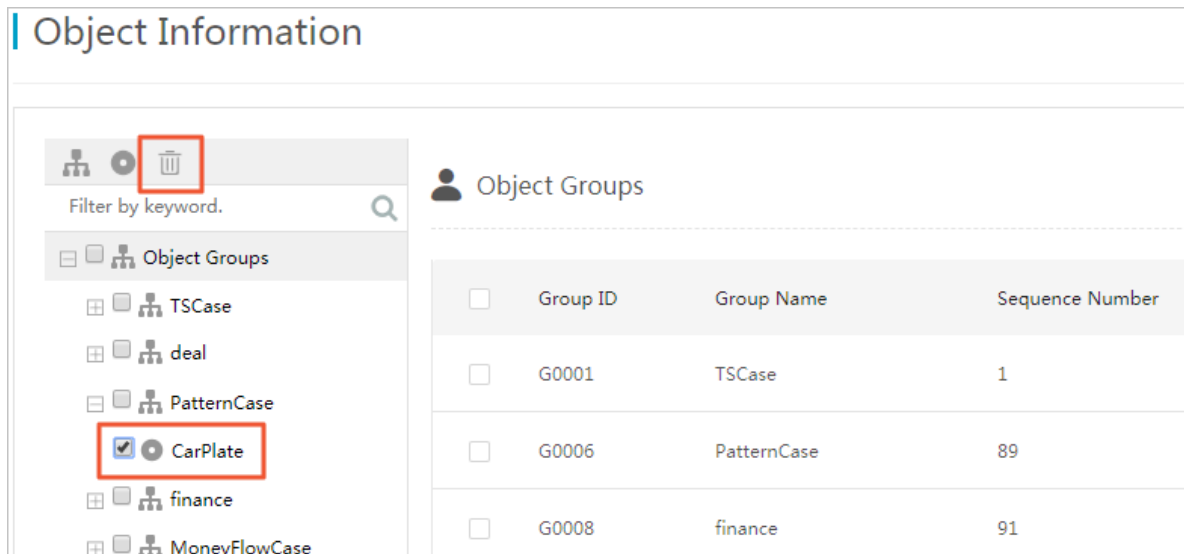
## Prerequisites


You have deleted the mappings between the object and the data table. You have deleted links and events that are referenced by the object.

## Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click **Objects**.

3. In the left-side navigation pane, click the object group to which the object belongs, and select one or more objects to be deleted. You can open multiple object groups and select objects from different groups.



4. Select the objects to be deleted, and click the  icon.
5. In the Delete Object Information dialog box that appears, click OK.

## 7.6 Link information

### 7.6.1 Link groups and links

#### 7.6.1.1 Create a link group

You can use link groups to classify links, so that you can search for and manage links with ease. Any link must be and can only be grouped into one link group. You need to create a proper link group before you create a link.

#### Prerequisites

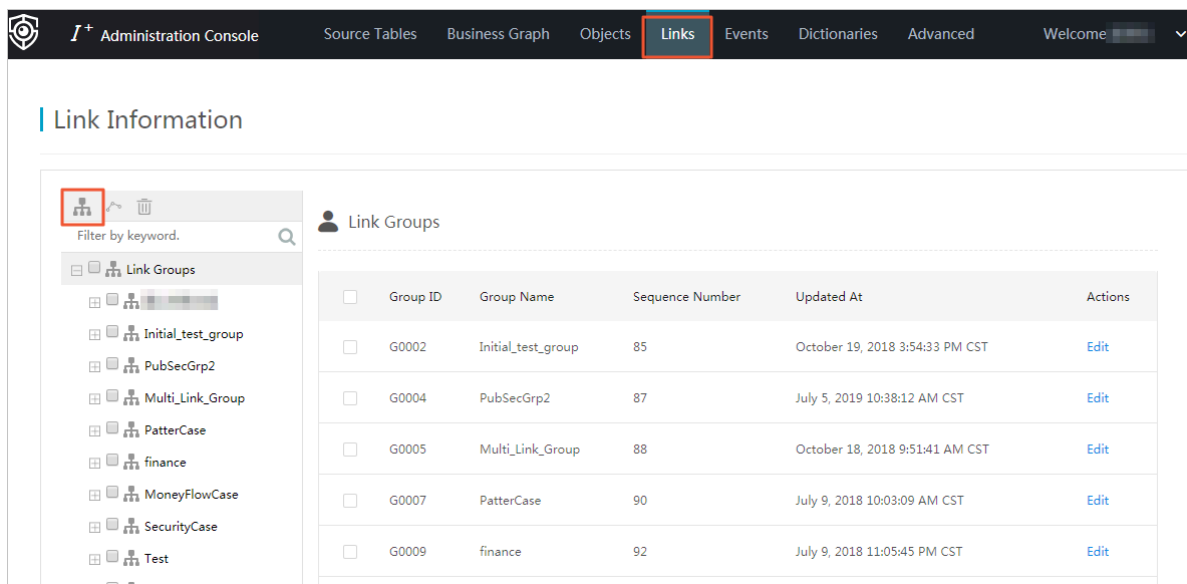
Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)



## 2. In the top navigation bar, click Links.



## 3. On the Link Information page, click the Create Group icon .

## 4. In the Create Group dialog box that appears, specify the Group Name and the Parent Group.

Create Group

\* Group Name:

\* Parent Group:

Link Groups

OK

Cancel

## 5. Click OK.

### 7.6.1.2 View links and link groups

In Graph Analytics, you can view all link groups in the current environment. You can understand the existing link groups and the link information under each group at any time.

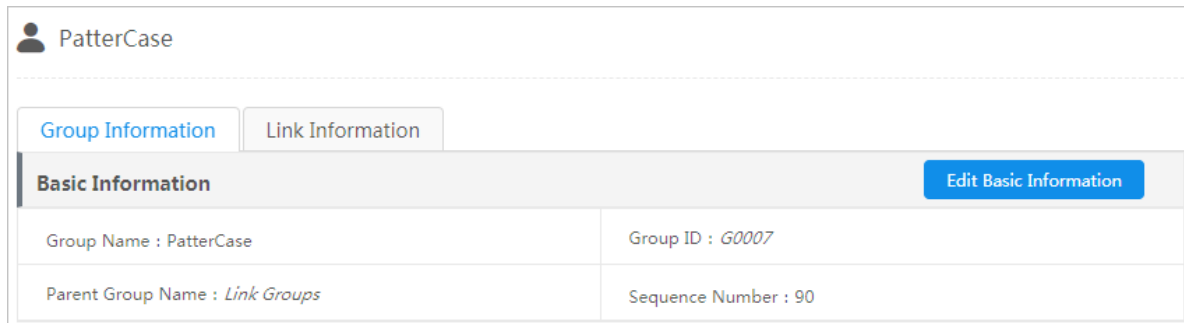
#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Links.

3. On the Link information page, click a link group in the left-side navigation pane. The Group Information and Link Information of the group are displayed on the right side of the page.

The left-side navigation pane displays all the link groups in the current environment. You can view the groups one by one.



Basic Information	
Group Name : PatterCase	Group ID : G0007
Parent Group Name : Link Groups	Sequence Number : 90

### 7.6.1.3 Modify a link or link group

In Graph Analytics, you can modify the name and order number of a link or link group. You can adjust the basic information of a link or link group at any time as needed. In Graph Analytics, you can enable or disable a link.

#### Prerequisites

- You have created a link group. For more information about how to create a link group, see [Create a link group](#).
- You have created a link that belongs to this link group. For more information about how to create a link, see [Create a first-degree link](#), [Create a second-degree link](#), or [Create a multi-degree link](#).
- To disable links, you must first delete the dependency information of these links, including the mappings between the links and data tables.

#### Context

This topic describes the following operations:

- Modify the basic information about a link group
- Modify the basic information about a link
- Enable and disable a link

After you have created a complete link (configured the link properties), the link is enabled automatically. You can disable a link if it is no longer used for a certain period of time and enable it again when necessary. You cannot use a link in Analytics Workbench after the link is disabled.

Modify the basic information about a link group

**You can modify the basic information of an object group by using the following two methods.**

Method	Procedure
Method one	<ol style="list-style-type: none"><li>1. <i>Log on to Administration Console of Graph Analytics.</i></li><li>2. <b>In the top navigation bar, click Links.</b></li><li>3. <b>In the Link Groups area that appears, select a link and click Edit in the Actions column.</b></li><li>4. <b>In the Edit Information dialog box that appears, specify the Group Name and Sequence Number, as shown in <i>Figure 7-24: Edit information.</i></b></li><li>5. <b>After you have configured these parameters, click OK.</b></li></ol>

Method	Procedure
<b>Method two</b>	<ol style="list-style-type: none"> <li>1. <i>Log on to Administration Console of Graph Analytics.</i></li> <li>2. <b>In the top navigation bar, click Links.</b></li> <li>3. <b>In the left-side navigation pane, click the link group to be modified.</b></li> <li>4. <b>On the right-side Group Information tab, click Edit Basic Information. Specify the Group Name and the Sequence Number, as shown in <i>Figure 7-25: Edit the basic information.</i></b></li> <li>5. <b>After you have configured these parameters, click Save.</b></li> </ol>

Figure 7-24: Edit information

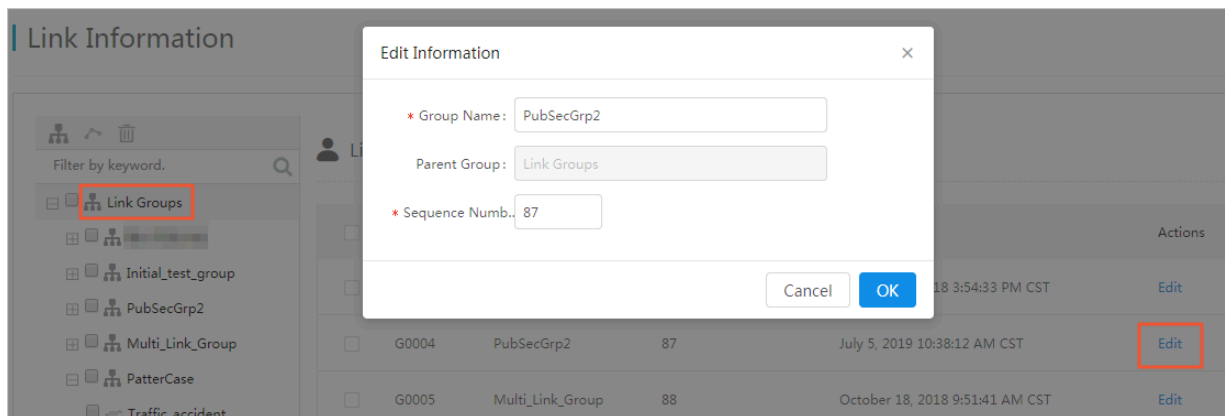
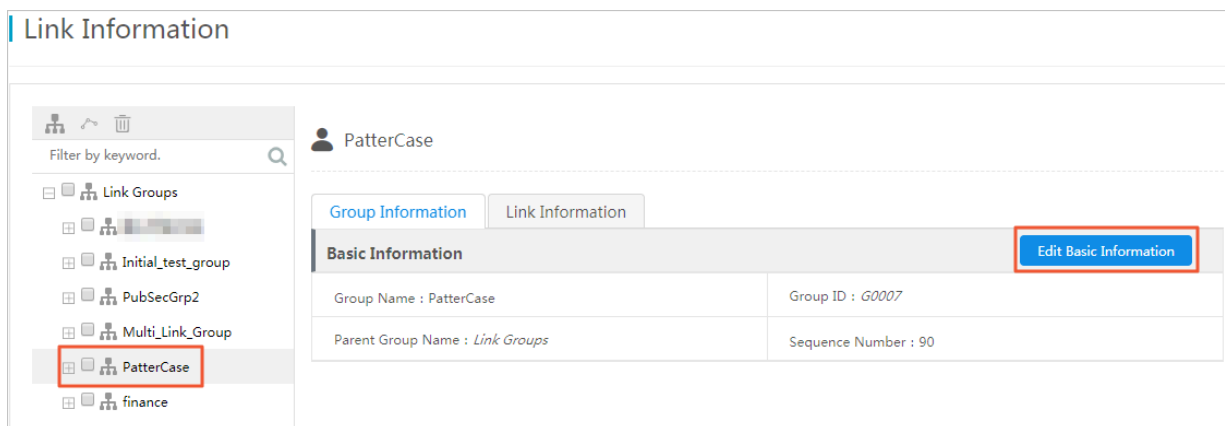


Figure 7-25: Edit the basic information



Modify the basic information about a link

1. *Log on to Administration Console of Graph Analytics.*
2. **In the top navigation bar, click Links.**
3. **In the left-side navigation pane, click the link group to which the link belongs, and then click the Link Information tab.**

4. On the Link Information tab, select a link and then click **Modify** in the Actions column.

5. In the Edit Information dialog box that appears, specify the Link Name and the Sequence Number.
6. Click OK.

Enable and disable a link

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click **Links**.
3. In the left-side navigation pane, click the link group to which the link belongs, and then click the **Link Information** tab on the right side of the page.
4. You can use the following methods to enable or disable a link on the **Link Information** tab.

Method	Procedure
Disable a link	<p><b>Disable a link individually:</b> Click the On icon next to the link to be disabled.</p> <p><b>Disable multiple links:</b> Select the links to be disabled and then click Off at the bottom of the page.</p> <p>After the link is disabled, the Status changes from a highlighted On icon to a gray Off icon.</p>

Method	Procedure
Enable a link	<p>Enable a link individually: Click the Off icon next to the link to be enabled.</p> <p>Enable multiple links: Select the links to be enabled and then click On.</p> <p>After the link is enabled, the Status changes from a gray Off icon to a highlighted On icon.</p>


#### 7.6.1.4 Delete a link or link group


You can delete the links and link groups that are no longer used.

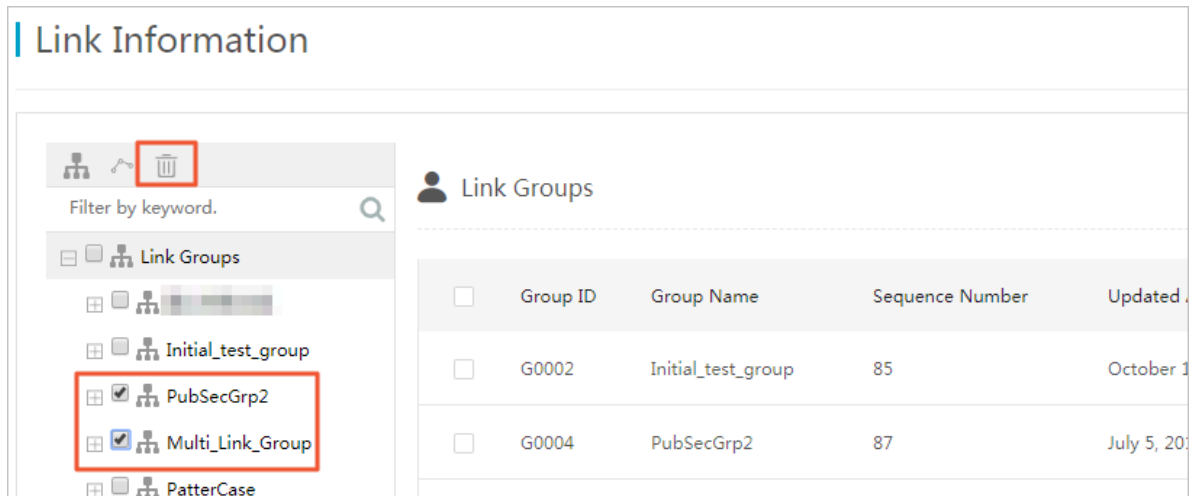
##### Prerequisites

- Before you delete a link, you must delete the dependency information of the link , for example, the mapping between the link and the physical table.
- Before you delete a link group, you must delete all the links in the group. You can only delete empty link groups.

##### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Links.
3. Optional: If there are links in the object group, you must delete these links first.
  - a. In the left-side navigation pane, select all links in the link group to be deleted, and click the  icon in the upper-left corner.
  - b. In the Delete Link Information dialog box that appears, click OK.

4. In the left-side navigation pane, select the link groups to be deleted, and then click the  icon in the upper-left corner.



5. In the Delete Link Information dialog box that appears, click OK.

## 7.6.2 First-degree links

### 7.6.2.1 Create a first-degree link

A first-degree link is the direct link between two objects. It is the basis of second-degree links and multiple-degree links. You must create a first-degree link between objects before you perform a link analysis in Graph Analytics. A complete link contains the basic information, property information, and business parameters. This topic describes how to configure the basic information of a link.

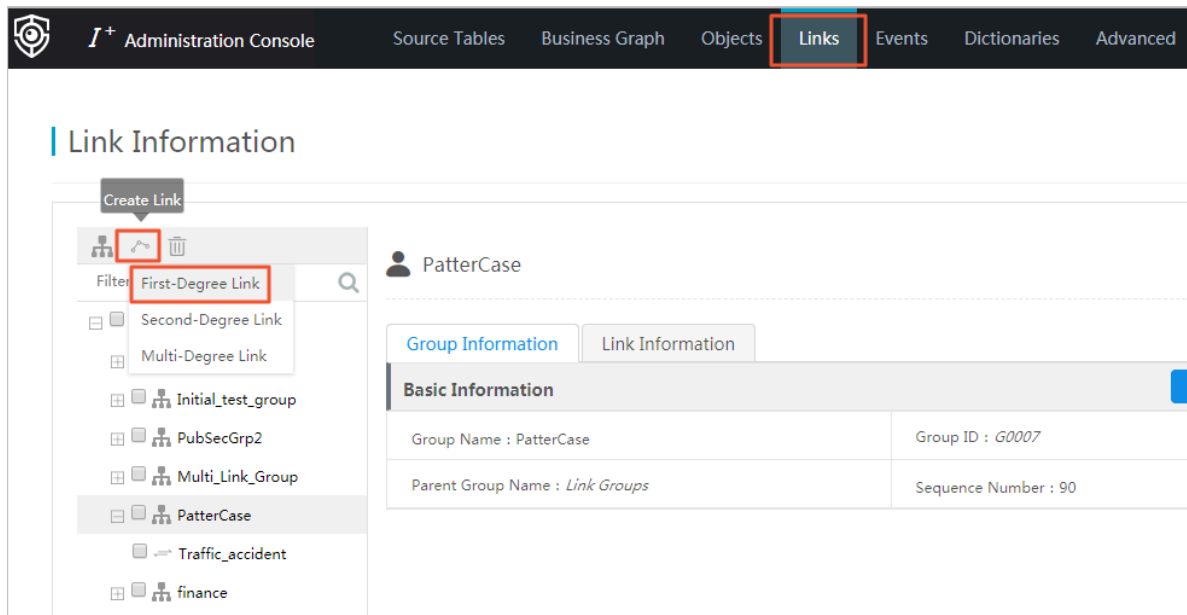
#### Prerequisites

- You have created and enabled a source object and a target object for the first-degree link. For more information about how to create an object, see [Create an object](#).
- You have create a link group for the first-degree link. For more information about how to create a link group, see [Create a link group](#).

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Links.

3. The Link Information page appears. Click the Create Link icon, and choose First-Degree Link.



4. In the Create Link dialog box that appears, specify the parameters.

The parameters are described in [Table 7-20: Parameters used to create a first-degree link](#).

Table 7-20: Parameters used to create a first-degree link

Parameter	Description
Link Name	The user-defined link name. It must be unique.
Group	The selected link group in the left-side navigation pane is used by default, but you can select another link group as needed.



Parameter	Description
Show in Graph	Specifies whether a link can be displayed on the Graph page. If the value is set to No, this link is unavailable on the Graph page.
Directionality	Specifies whether the link is directional. For example, if the source object A calls the target object B, a link is established. If this link is set as directional, it will be displayed on the Graph page. The link direction is from A to B (A > B).
Source Object	Select the source object of the first-degree link in the drop-down list.
Target Object	Select the target object of the first-degree link in the drop-down list.
Description	Enter the description of the first-degree link, so the users can understand it with ease.

5. Click OK.

#### What's next

1. After you have created a first-degree link, you must specify the properties and configure the business parameters based on your requirements. For more information, see [Configure link properties and business parameters](#).
2. To use the new link, you must log on to Analytics Workbench again.

#### 7.6.2.2 Configure link properties and business parameters

After you add a first-degree link, you need to configure the properties and business parameters of the link based on your business requirements, so that you can view and apply this link in Analytics Workbench. This topic describes how to configure the properties and business parameters of a first-degree link.

#### Prerequisites

You have created a first-degree link. For more information about how to create a link, see [Create a first-degree link](#).

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click Links on the top of the page.
3. In the left-side navigation pane, click the link group that contains the first-degree link to be configured, and then click the link.

#### 4. In the right-side area, click the Correlations and Properties tab.

On the Correlations and Properties page, if the link has been mapped to a data table in the data source, the property information section will display Data Source and Table. You can click the table name to go to the table page.

call\_link\_01 On

Link Information Correlations and Properties

Data Source: demo01 Table: call\_records Save

Basic Information Add Row

Property ID	* Property Name	* Unique ID	Show in Details	Conditional Query	Show in Statistics	Available	* Display Type	* Query Type	Security Level	Actions
L00000014P0001	caller_num	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal ...	S1	
L00000014P0002	callee_num	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Character	String Equal ...	S1	

Source Property Mapping

Source Object Correlated Property : phone\_num\_01-phone\_num \* Link-Correlated Property : caller\_num

Target Property Mapping

Target Object Correlated Property : phone\_num\_01-phone\_num \* Link-Correlated Property : callee\_num

Advanced

Accumulative Statistics Settings

Link Weight Settings

#### 5. Set the basic configurations in the Basic Information page.

Click Add Row to add a property for a link in Basic information.




##### Note:

Link-Correlated Property in Source Property Mapping and Link-Correlated Property in Target Property Mapping are not the same and they are both required to be configured. Therefore, a link must have at least two properties.

The configuration items in Basic Information are described in [Table 7-21: Description of basic configuration items](#).

Table 7-21: Description of basic configuration items

Configuration item	Description
Property ID	The ID of a property. It is automatically generated.

Configuration item	Description
Property Name	The name of the property. If you select <b>Available</b> , the name of the property will be displayed in <b>Analytics Workbench</b> .
Unique ID	Defines the logical primary key of a link table.
Show in Details	If you select this item, the property will be displayed in the <b>Details</b> tab in <b>Analytics Workbench</b> .
Conditional Query	If you select this item, when you perform an analysis on the <b>Graph</b> page in <b>Analytics Workbench</b> , you can query a link based on the link type in <b>Link Type</b> .
Show in Statistics	If you select this item, the property is displayed in <b>Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; Statistics</b> . Otherwise, it is not displayed.
Available	<p>If you select this item, the property takes effect and is displayed in <b>Analytics Workbench</b>.</p> <p>The <b>Available</b> parameter is automatically selected for a property if any of the following parameters has been selected for the property: <b>Unique ID</b>, <b>Show in Details</b>, <b>Conditional Query</b>, and <b>Show in Statistics</b>. The <b>Available</b> parameter is automatically deselected for a property if all of the preceding parameters are deselected for the property.</p>
Display Type	<p>After you set the display type, the property is displayed in <b>Analytics Workbench &gt; Graph &gt; right-side navigation pane &gt; the Details</b> tab and the <b>Property</b> tab based on the selected type.</p> <div>  <b>Note:</b> To display a property in the format of <b>Dictionary</b>, you need to configure a dictionary first. </div>
Query Type	The data type that is supported in the query condition of a property. If you select <b>Dictionary</b> for <b>Display Type</b> , you must select <b>Dictionary Option</b> for <b>Query Type</b> .
Security Level	The security level for a property. A user with a lower security level cannot view the property.
Search Item Configuration	Associates this property with a search item so that the link can be searched by this property in <b>Analytics Workbench</b> .

Configuration item	Description
Default Query Condition Settings	Specifies the default condition used for a link query. If other properties are used as conditions for the query, this condition is also included by default.
Authorization Code	After the authorization code function has been enabled, only users with the required authorization code can access this property.
Derived Property	Sets a property as a derived property so that it can be generated automatically based on other properties. You can set the derivative method based on your requirements.
Move Up and Move Down arrows	The Move Up arrow and the Move Down arrow can be used to adjust the order of properties that are displayed in Analytics Workbench.

**6. You can set Link-Correlated Property in Source Property Mapping and Target Property Mapping.**

These two configurations are related with the `Source Object` and the `Target Object` of the link.

Take `Source Property Mapping` as an example: `Source Object Correlated Property` and `Link-Correlated Property` must be mapped to the same column in the same table. The `Source Object Correlated Property` parameter is the primary key property of the source object, which is automatically loaded according to the `Source Object` parameter. For the `Link-Correlated Property` parameter, you must select the link property that is mapped to the same column in the same table as the `Source Object` primary key.

Set `Target Property Mapping` in the same way you set `Source Property Mapping`.

## 7. Optional: Set Advanced, Accumulative Statistics Settings, and Link Weight Settings based on your requirements.

For more information about the configurations of key parameters, see [Table 7-22: Parameter configurations](#).

Table 7-22: Parameter configurations

Category	Configuration item	Description
Advanced	Chronological Time Property	Specifies the link properties based on which chronological analysis is performed. From the drop-down list, select one or more link properties of which the query type is time.
	Time Property for Behavior Analysis	Specifies the link properties based on which behavior analysis is performed. From the drop-down list, select one or more link properties of which the query type is time.
	Linked Times	Specifies the property of which the number of the same values are counted. The total number is displayed as the number of link occurrences. The Linked Times parameter is used as the default setting to filter link types. For example, if there are two lines of $A > C$ calls in the call log, the analysis result displays that the number of $A > C$ calls is two.
	Details Sorting Property	Specifies the property by which the returned behavior details are sorted by default.
Accumulative Statistics Settings	N/A	<p>Used to perform logical statistics for link properties of which the query type is numeric range. The logical statistics operations include top, =, and <math>\geq</math>. This configuration applies to business scenarios where statistics filtering is required for link query results. The Linked Times parameter is used to filter records in link query results.</p> <p>You can add statistical conditions to filter the link properties of which the query type is numeric range.</p>

Category	Configuration item	Description
Link Weight Settings	N/A	You can specify a link property of which the query type is numeric range and calculate the link weight based on the numeric range specified for the link property.

8. After you have modified the parameters, click **Save**. A message is displayed, indicating that the modifications have been saved.

### 7.6.3 Create a second-degree link

A second-degree link refers to the relationship between two objects established by using an intermediary object. Compared with a first-degree link, a second-degree link can mine a more complex relationship network among objects.

#### Prerequisites

- Make sure that you have created related first-degree links for the second-degree link. For more information about how to create a first-degree link, see [Create a first-degree link](#).
- Make sure that you have created a link group for the second-degree link. For more information about how to create a link group, see [Create a link group](#).

#### Context

A second-degree link is created based on first-degree links and can be split into two first-degree links. For example, object A has a first-degree link with object C, and object B also has a first-degree link with object C. In this case, you can build a second-degree link between object A and object B. Typically, people taking the same train or plane or staying in the same hotel have a second-degree link and can be analyzed by using the second-degree link model.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click **Links**.
3. On the Link Information page that appears, click the **Create Link** icon and select **Second-Degree Link**.

#### 4. Configure the parameters in the Link Definition page.

The parameters are described in [Table 7-23: Description of link definition parameters](#).

Call\_second\_link\_01 ☐ Off ?

1 Link Definition 2 Link Element Definition 3 Link Computing Configuration

\* Link Name: Call\_second\_link\_01  
(The link name must be 1 to 16 characters in length and can contain letters, number, and Chinese characters.)

Link ID:

\* Group: Call\_second\_links

\* Show in Graph: Yes

\* Source Object: phone\_num\_01

\* Target Object: phone\_num\_01

\* First-Degree Link: call\_link\_01

Description: Enter a description no more than 20 characters in length

Previous Next Submit

Table 7-23: Description of link definition parameters

Parameter	Description
Link Name	The user-defined link name. It must be unique.
Group	The selected link group in the left-side navigation pane is used by default, but you can select another link group as needed.
Show in Graph	Specifies whether a link can be displayed on the Graph page. If the value is set to No, this link definition is unavailable on the Graph page.
Source Object	Select the source object of the second-degree link from the drop-down list.
Target Object	Select the target object of the second-degree link from the drop-down list.
First-Degree Link	Select the first-degree links between the source object and the target object to create a second-degree link.
Description	Enter the description of the second-degree link, so the users can understand the link with ease.

5. After you have configured the preceding parameters, click Next to set the parameters in Link Element Definition.

The parameters are described in [Table 7-24: Description of link element parameters](#).

Call\_second\_link\_01 ☐ Off ⓘ

1 Link Definition 2 Link Element Definition 3 Link Computing Configuration

**Define Basic Query Conditions** [Select Query Properties](#)

Basic Query Condition ID	Query Property	Query Method	Default Query Value	Actions
undefinedC0001	caller_num	Equal Value	<input type="text"/>	Delete   ⬆ ⬇ ⬆
undefinedC0002	callee_num	Equal Value	<input type="text"/>	Delete   ⬆ ⬇ ⬆

**Define Base Link** [Select Link Properties](#) [Select Object Properties](#)

Base Link ID	Base Link Name	Referenced First-Degree Link Property	Correlation Rule	Default Value	Actions
undefinedC0003	<input type="text" value="caller_numSame"/>	caller_num	Equal Value	<input type="text"/>	Delete

[Previous](#) [Next](#) [Submit](#)

Table 7-24: Description of link element parameters

Section	Parameter	Description
Define Basic Query Conditions	Select Query Properties	Defines the basic query conditions that can be used to query a link. You can view the conditions in the Basic Properties section in the Link Type settings when you configure a link extension.
Define Basic Link	Select Link Properties and Select Object Properties	<p>Defines the table columns that are needed to perform a query. You can define the alias of a column in Base Link Name. The base link can be referenced in Link Configuration in the next step.</p> <p>Configure the Correlation Rule. The Default Value will be used when you perform a query in Analytics Workbench.</p>



6. After you have configured these parameters, click Next and then set the parameters in Link Computing Configuration.

The parameters are described in [Table 7-25: Link computing configuration parameters](#).

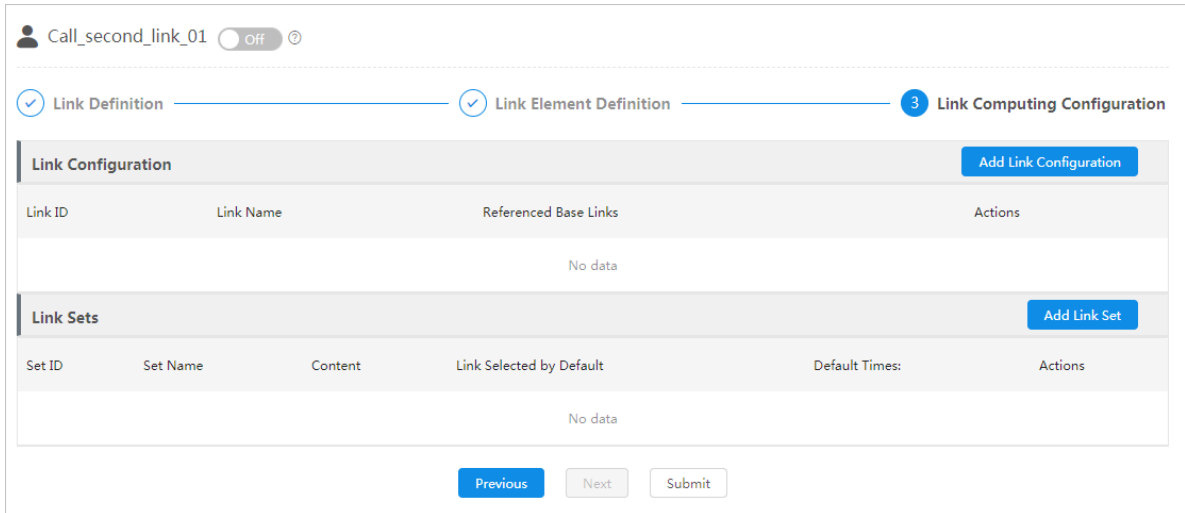


Table 7-25: Link computing configuration parameters

Section	Parameter	Description
Link Configuration	Add Link Configuration	<p>Combines multiple base links to form a multi-condition link.</p> <ol style="list-style-type: none"> <li>Click Add Link Configuration, specify the Link Name and then select links.</li> <li>Click OK to add a multi-condition link.</li> <li>You can repeat the preceding steps to create more multi-condition links as needed.</li> </ol>

Section	Parameter	Description
Link Sets	Add Link Set	<p>A link set is a group of specific multi-condition links configured in Link Configuration. Such link sets define the advanced query conditions for link queries in link analysis.</p> <p>a. Click Add Link Set and set the parameters.</p> <p>Key parameters for adding a link set are described as follows:</p> <ul style="list-style-type: none"> <li>• <b>Default Times:</b> Specifies the minimum number of occurrences of a link set that can be counted as a query match. The default value is 2.</li> <li>• <b>Base Links:</b> The links displayed in the Select Base Links area are all multi-condition links that have been configured in Link Configuration. Each link can only be contained in one link set.</li> <li>• <b>Link Selected by Default:</b> The default query condition that is displayed on Analytics Workbench. When a link set that contains multiple multi-condition links is used for a link query, each analysis is performed based on one of the links. By default, the Link Selected by Default is used.</li> </ul> <p>b. Click OK.</p> <p>c. You can repeat the preceding steps to create more link sets as needed.</p>

Figure 7-26: Add a link configuration

Figure 7-27: Add a link set

7. After you have configured the preceding parameters, click **Submit** to create a second-degree link.

### 7.6.4 Create a multi-degree link

A multi-degree link refers to the relationship between two objects established by using multiple intermediary objects. Compared with a first-degree link and a second-degree link, a multi-degree link can mine a more complex relationship network among objects.

#### Prerequisites

- Make sure that you have created related first-degree links for the multi-degree link. For more information about how to create a first-degree link, see [Create a first-degree link](#).
- Make sure that you have created related second-degree links for the multi-degree link. For more information about how to create a second-degree link, see [Create a second-degree link](#).
- Make sure that you have created a link group for the multi-degree link. For more information about how to create a link group, see [Create a link group](#).

#### Context

A multi-degree link uses multiple first-degree links and second-degree links to query the relationship between two objects.

For example, if mobile phone A and mobile phone B do not have call or text message records and only have email correspondence records, there are only indirect links between these two mobile phones. The analysis of this case is as follows: There is no direct relationship between mobile phone A and mobile phone B, so they cannot be directly linked. When mobile phone A sends an email to mobile phone B, an indirect link is established. This indirect link involves three first-level links: the link between mobile phone A and mailbox A, the link between mailbox A and mailbox B, and the link between mobile phone B and mailbox B. You can query mobile phone B through mobile phone A by using these three links. Similar cases include QQ and WeChat for mobile phones.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click **Links**.

3. In the Link Information page that appears, click the Create Link icon and select Multi-Degree Link.
4. Configure the parameters in the Link Definition page.

The parameters are described in [Table 7-26: Description of link definition parameters](#).

1 Link Definition 2 Link Configuration

\* Link Name:   
(The link name must be 1 to 16 characters in length and can contain letters, number, and Chinese characters.)

Link ID:

\* Group:

\* Show in Graph:

\* Source Object:

\* Target Object:

Description:

Previous Next Submit

Table 7-26: Description of link definition parameters

Parameter	Description
Link Name	The user-defined link name. It must be unique.
Group	The selected link group in the left-side navigation pane is used by default, but you can select another link group as needed.
Show in Graph	Indicates whether a link can be displayed on the Graph page. If the value is set to No, this link is unavailable on the Graph page.
Source Object	Select the source object of the multi-degree link from the drop-down list.
Target Object	Select the target object of the multi-degree link from the drop-down list.
Description	Enter the description of the multi-level link, so the users can understand the link with ease.

5. Click Next.

6. On the Link Configuration page, click **Select Links** in the upper-right corner, and then select the related first-degree links and second-degree links.

Select Links:

- ▼ ☐ Link Groups
  - ▶ ☐ [Redacted]
- ▼ ☐ call\_links
  - ☐ call\_link\_01 Source Object: phone\_num\_01 Target Object: phone\_num\_0
- ▼ ☐ Call\_second\_links
  - ☐ Call\_second\_link\_01 Source Object: phone\_num\_01 Target Object: phone
- ▼ ☐ Call\_multi\_links

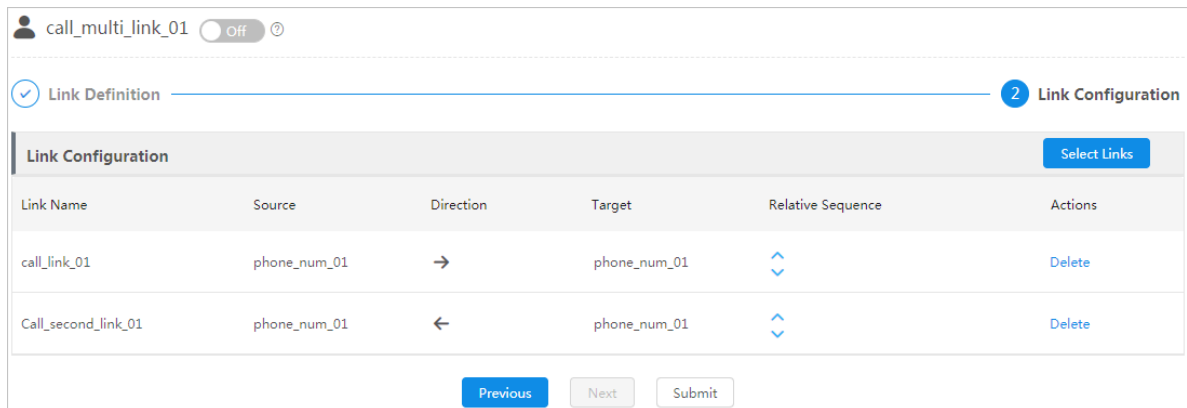
Cancel OK

7. After you have configured the preceding parameters, click **OK** to add a base link.

Typically, you may need to add a base link twice with opposite query directions . For example, if you send an email by using your mobile phone, the query direction is from the mobile phone to the email box. If you receive an email on your mobile phone, the query direction is from the email box to your mobile phone. Therefore, you can add this first-degree link between the mobile phone and the email box twice with opposite query directions.

8. On the Link Configuration page, click the arrow in the Query Direction column to adjust the query direction.

The source object and the target object must be correlated: source object > intermediate object > ... > intermediate object > target object.



Link Name	Source	Direction	Target	Relative Sequence	Actions
call_link_01	phone_num_01	→	phone_num_01	⬆️⬆️	Delete
Call_second_link_01	phone_num_01	←	phone_num_01	⬆️⬆️	Delete

9. After you have configured the preceding parameters, click Submit to create a multi-degree link.

## 7.7 Event information

### 7.7.1 Event groups


#### 7.7.1.1 Create an event group

You can use event groups to classify events, so that you can easily find and manage events. Any event must be and can only be grouped into one event group. You need to create a proper event group before you create an event.

#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. At the top of the left-side navigation pane, click the Create Group icon . The Create Group dialog box appears.

4. Enter a Group Name as needed, and then select Event Groups from the Parent Group drop-down list.
5. Click OK.

### 7.7.1.2 View an event group

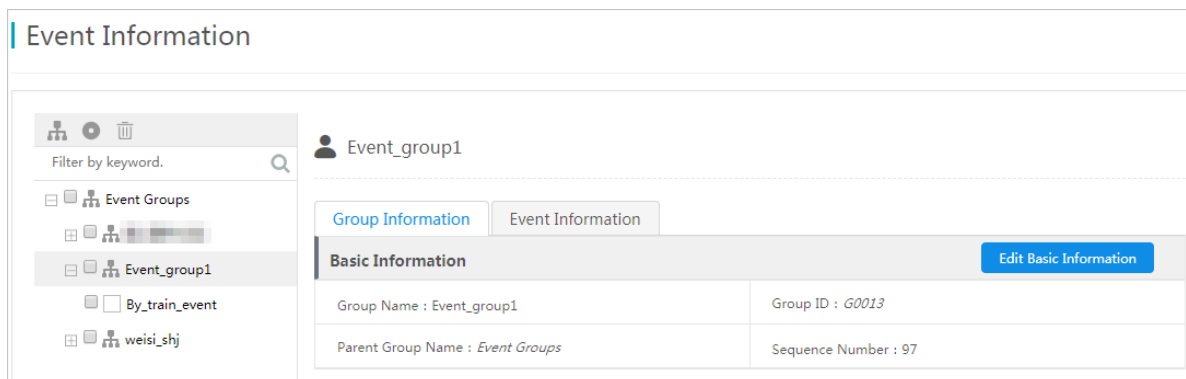
You can view the basic information of an event group and the events in the group in Administration Console.

#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. From the left-side navigation pane, select an event group such as By Train. The detailed information about this event group is displayed on the right side of the page.



You can view event group information on the Group Information and Event Information tabs. The Group Information tab displays the basic information about the event group. The Event Information tab displays all events in the group.

### 7.7.1.3 Modify an event group

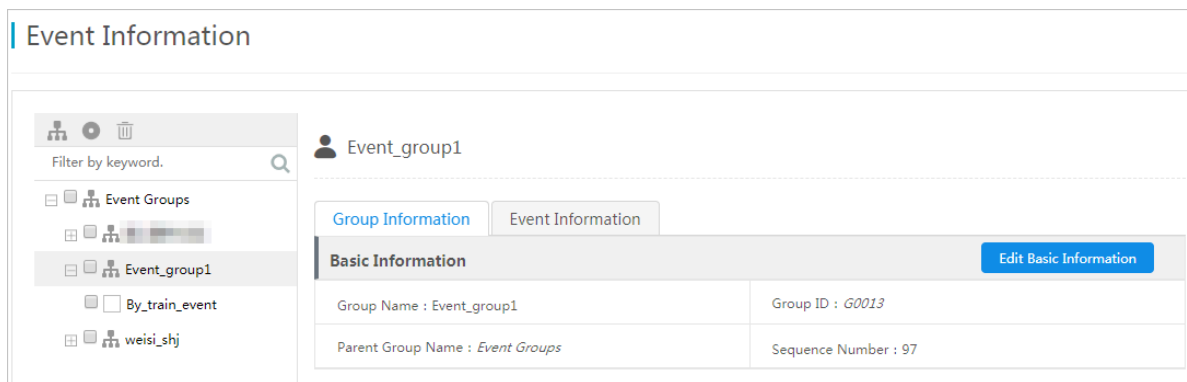
You can modify the basic information of an event group and the events in the group in Administration Console.

#### Prerequisites

**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. From the left-side navigation pane, select an event group such as By Train. The detailed information about this event group is displayed on the right side of the page.



4. Click Edit Basic Information on the Group Information page. You can then modify the Group Name and Sequence Number of the group.
5. Click Save.

#### 7.7.1.4 Delete an event group

If an event group is no longer used, you can first delete all events in the group and then delete the event group.


### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have deleted all events in the event group. For more information about how to delete an event, see [Delete an event](#).

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.



3. Select the event groups that you want to delete from the left-side navigation pane, and then click the Delete icon  on the top of the pane.
4. In the message box that appears, click OK.

## 7.7.2 Events


### 7.7.2.1 Create an event

Events are used to analyze the behaviors of objects. Event information is used to define the event models in Graph Analytics. A complete event contains basic event information, properties, and relevant parameters. This topic describes how to configure the basic information of an event.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created an event group for the event. For more information about how to create an event group, see [Create an event group](#).

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. On the top of the left-side navigation pane, click the Create Event icon . Set the parameters in the Create Event dialog box.

For more information about the parameters and descriptions, see [Table 7-27: Parameters of created events](#).

Table 7-27: Parameters of created events

Parameter	Configuration method
Event Name	The name of the event. Set this parameter as needed . Each event name must be from 1 to 16 characters in length, and can contain letters, digits, and Chinese characters.
Event Description	The description of the event. Set this parameter as needed to help users understand the event.
Group	The event group that the event belongs to. Set this parameter as needed.

Parameter	Configuration method
Event Icon	The event icon. You can select an icon in Icon Library, or enter an accessible URL to reference an external icon.

4. After you set the parameters, click OK. A message is displayed, indicating that the event has been created.

#### What's next

1. After you have created an event, you must configure the properties and business parameters based on your business requirements. For more information, see [Configure event property parameters](#).
2. After you have configured the properties and business parameters of an event, you must log on to Analytics Workbench again to use the new event.

#### 7.7.2.2 Configure event property parameters

After you have created an event, you must configure the event properties. Event properties are critical to an event. You can configure event properties, and correlate the properties to objects on the Property Information tab.

#### Prerequisites

Make sure that you have obtained an account and a password for Graph Analytics and you have been authorized with the required Event Permissions.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. Click an event in the left-side navigation pane, and click the Property Information tab on the right side of the page. The Property Information tab appears.
4. Set the required event parameters.

The required parameters are displayed in the Basic Information and Set Mappings Between Correlated Objects and Properties areas. The event parameters are described in [Table 7-28: Required event property parameters](#).




**Note:**



**To save the property information, you must set all the required parameters.**

Table 7-28: Required event property parameters

Area	Parameter	Description
Basic Information	Property ID	The ID of a property. It is automatically generated.
	Property Name	The property name that is displayed on Analytics Workbench. We recommend that you enter a name that describes the business type.
	Primary Key	Each primary key uniquely identifies an event. A property cannot be deleted after it has been configured as a primary key.

Area	Parameter	Description
	Show in Graph	 <b>Note:</b> For each event, you must select at least one property to be displayed in the graph.  If you select this parameter, the Property Name of this property will be displayed in Graph in Analytics Workbench together with the object node. Otherwise, the property is not displayed. For example, if this parameter is selected by the ID card property of an event, the ID card number will be displayed in Graph with the event. If the name property is also selected, the name and the ID card number will be displayed with the event.  If you select this parameter for a specific property, a bubble icon appears next to the option. You can click the bubble icon to set whether to show the Property Name in Graph. For example, if you select this parameter for an ID card number, and set to display the Property Name in Graph, the ID card property displayed in Graph will be ID card number: xxxxxx.
	Show in Properties	If you select this parameter for a property, the property will be displayed in the right-side pane on the Graph page of Analytics Workbench. To view the property, choose Properties > Event Properties.
	Element Identifier	The element identifier of a property. Set this parameter based on the actual property.

Area	Parameter	Description
	<b>Conditional Query</b>	If you select this parameter for a property, the event can be queried based on this property in Link Type on the Graph page of Analytics Workbench when you perform a relationship analysis.
	<b>Show in Statistics</b>	If you select this parameter for a property, the event will be displayed in the right-side pane on the Graph page of Analytics Workbench. To view the property, choose Statistics > Event Distribution.
	<b>Available</b>	<p>If you select this parameter for a property , the property takes effect and can be displayed on the Graph page of Analytics Workbench.</p> <p>This parameter is selected by default and cannot be changed.</p>
	<b>Display Type</b>	The format in which a property is displayed in the right-side pane of the Graph page on Analytics Workbench. Set this parameter as needed.
	<b>Query Type</b>	The data type that is supported in the query condition of a property.
	<b>Security Level</b>	The security level for a property. A user with a lower security level cannot view the property.

Area	Parameter	Description
	Search Item Configuration	<p>Click the More icon  in the Actions column corresponding to a property. Set the following four parameters:</p> <ul style="list-style-type: none"> <li>• <b>Search Item Configuration:</b> Search items are displayed in the drop-down list only after they have been configured in <a href="#">Configure a search item</a>.</li> <li>• <b>Default Query Condition Settings:</b> The default condition used for a link query. If other properties are used as conditions for a query, this condition is also included by default.</li> <li>• <b>Authorization Code:</b> After the authorization code function has been enabled, only authorized users can access this property.</li> <li>• <b>Derived Property:</b> After a property is set as a derived property, it can be generated automatically based on other properties. Configure the method in which the column is generated based on your needs.</li> </ul>
	Default Query Condition Settings	
	Authorization Code	
	Derived Property	
	Delete	When a property is no longer used, you can click the Delete icon  to delete this property.
Set Mappings Between Correlated Objects and Properties	Add Correlated Object	<p>Adds a mapping between an object and the event. One event must have at least two objects mapped to it.</p> <p>Click Add Correlated Object to add a correlated object, and configure the mapping between the event and the primary keys of the object you have added.</p>

5. **Optional:** After you have set the required parameters, you can set the optional parameters as needed.

The optional parameters are included in the Advanced and Display Settings areas. The optional parameters are described in [Table 7-29: Optional event property parameters](#).



**Note:**

**Location Settings are currently not supported.**

Table 7-29: Optional event property parameters

Area	Parameter	Description
Advanced	Behavior Property	Defines the properties based on which a behavior analysis is performed.
	Default Details Sorting Property	Defines the property by which the details are sorted.
	Logical Relation of Authorized Properties	<p>The logical relationship between the authorization codes of properties in each record.</p> <ul style="list-style-type: none"><li>• <b>AND:</b> The current record is visible only to the users who meet all authorization code conditions of the properties in this record.</li><li>• <b>OR:</b> The current record is visible to the users who meet any one authorization code condition of the properties in this record.</li></ul>
Display Settings	Enable Display	Indicates whether to show the event details.
	Group-by Properties	Indicates the property based on which events are aggregated. For example, aggregate Travel Events into a folder based on the Train Number property.

6. After you have completed the configurations, click **Save** in the upper-right corner. A success message is displayed after the modifications have been saved.
- An event is automatically enabled after its properties have been saved.

### 7.7.2.3 Enable and disable an event

If you do not need to use a specific event during a specific period, you can disable this event, and this event can be re-enabled.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- To disable some events, you must first delete the dependency information of these events, including the mappings between the events and data tables and the objects referenced by the events.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. You can enable or disable an event by using one of the following methods.

Method	Operation
Operations on the Event Information page	In the left-side navigation pane, select an event to be enabled or disabled, and click the toggle switch to enable or disable the event.
Operations in Event Groups	<p>a. In the left-side navigation pane, select the event group that contains the event you want to enable or disable. Click the Event Information tab to go to the Event Information page.</p> <p>b. Click the toggle switch to enable or disable the event.</p> <p>You can also select the events to be enabled or disabled, and click the Off button or the On button at the bottom of the page to disable or enable multiple events at a time.</p>

### 7.7.2.4 View an event

After you have configured an event, you can view the newly created event and all the events that have been created.

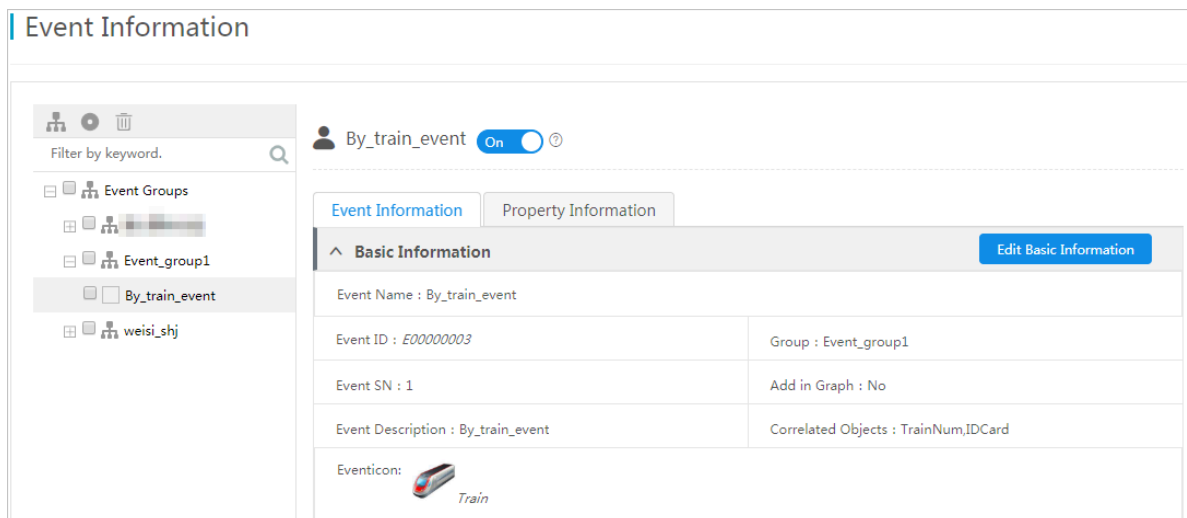
#### Prerequisites



**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.
3. In the left-side navigation pane, select an event and view the event details on the right side of the page.



**Two tabs are displayed: The Event Information tab and the Property Information tab. The Event Information tab displays the basic information of the event. The Property Information tab displays the properties, correlated objects and property mappings, advanced settings, location settings, and display settings.**

### 7.7.2.5 Modify an event

**You can modify the basic information of the event you have created based on your needs.**

### Prerequisites

**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Click the Events tab in the top navigation bar. The Event Information page appears.

3. In the left-side navigation pane, select an event to be modified, and click **Edit Basic Information** to modify the parameters based on your requirements.

The parameter configurations are described in [Table 7-30: Modify parameter configurations of events](#).

Table 7-30: Modify parameter configurations of events

Parameter	Description
Event Name, Group, Event Description, and Event Icon	For more information about parameter descriptions, see <a href="#">Create an event</a> .
Event SN	You can change the event sequence number based on your requirements.
Add in Graph	This parameter is not in use. You do not need to configure it.

4. After you have modified the parameters, click **Save**. A message is displayed, indicating that the modifications have been saved.


### 7.7.2.6 Delete an event

You can delete an event that is no longer used.

#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Administration Console of Graph Analytics](#).
2. Click the **Events** tab in the top navigation bar. The **Event Information** page appears.
3. In the left-side navigation pane, select the events to be deleted, and click the **Delete** icon  on the top.
4. In the dialog box that appears, click **OK**. A message is displayed, indicating that the selected events have been deleted.

## 7.8 View the business graph

You can directly view all configured link models on the Business Graph page.

Solid lines indicate first-degree links, and dotted lines indicate second-degree or multiple-degree links.

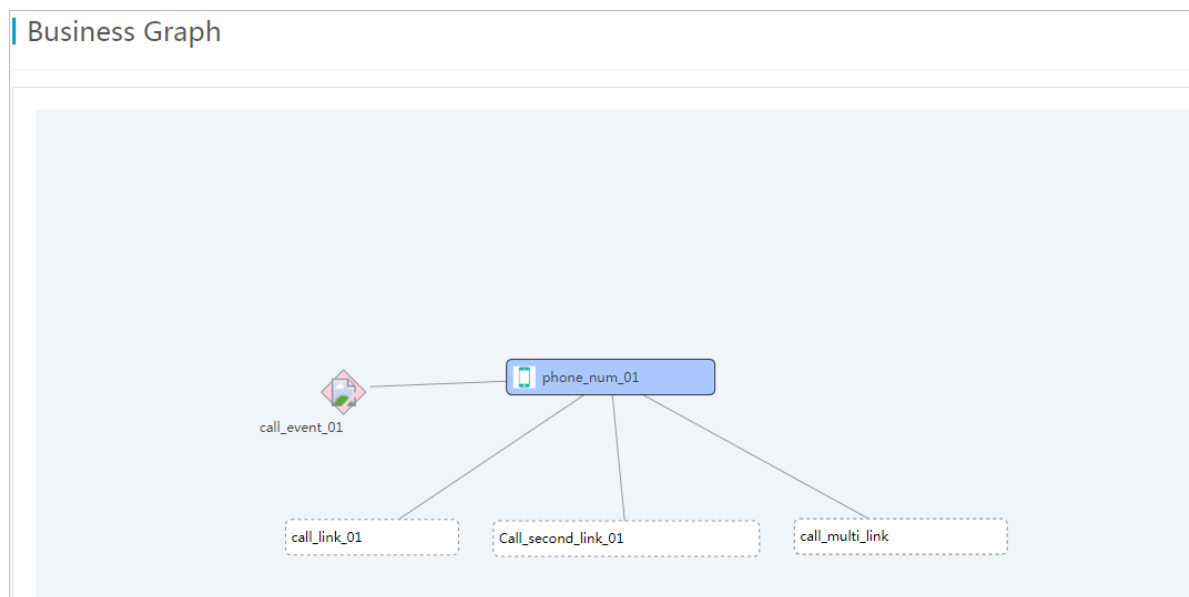
### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, click Business Graph.

On the Business Graph page, you can view the created link models between objects, links, and events.



## 7.9 Advanced configurations

### 7.9.1 Manage a system model

The Import Models page allows you to configure system models that are used to import data to Analytics Workbench.

### Prerequisites

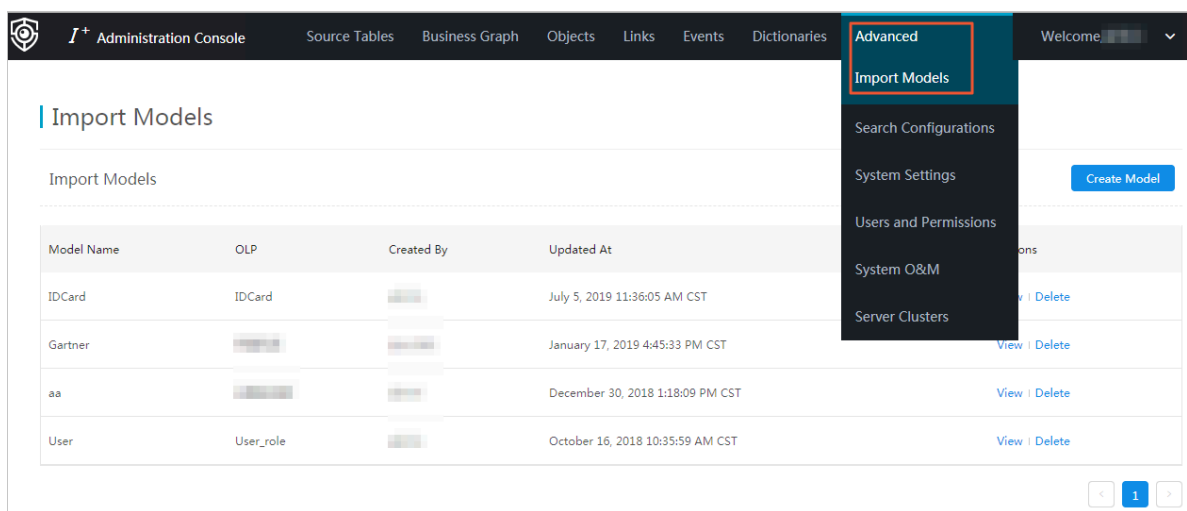
**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

## Context

**You can create, view, and delete models on the Import Models page. This topic provides examples to help you learn more about these operations.**

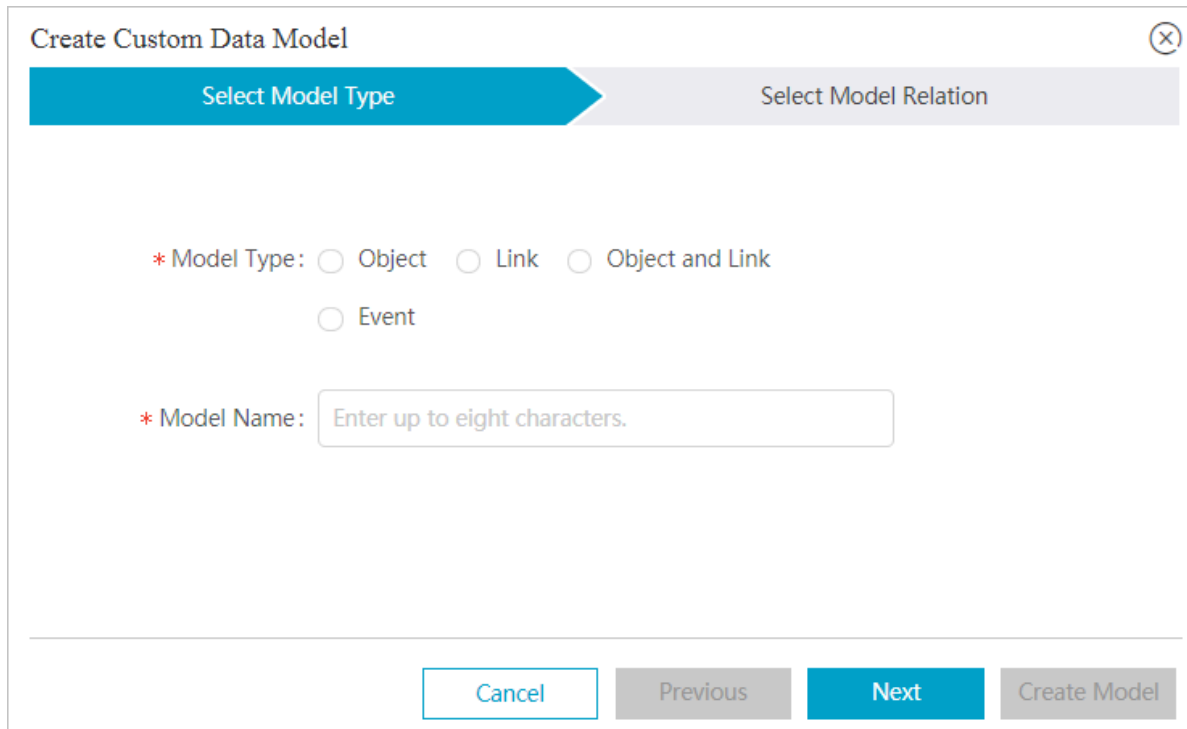
Create a model

1. [Log on to Administration Console of Graph Analytics.](#)
2. **In the top navigation bar, choose Advanced > Import Models.**



3. **On the Import Models page, click Create Model in the upper-right corner of the page.**

4. In the Create Custom Data Model dialog box, specify the Model Type and the Model Name.



The dialog box is titled "Create Custom Data Model" and has a close button (X) in the top right corner. It features a progress bar with two steps: "Select Model Type" (highlighted in blue) and "Select Model Relation" (greyed out). Below the progress bar, there are two sections. The first section is labeled "\* Model Type:" and contains four radio button options: "Object", "Link", "Object and Link", and "Event". The second section is labeled "\* Model Name:" and contains a text input field with the placeholder text "Enter up to eight characters." At the bottom of the dialog box, there are four buttons: "Cancel", "Previous", "Next", and "Create Model". The "Next" button is highlighted in blue, while the others are greyed out.

Create Custom Data Model

Select Model Type      Select Model Relation

\* Model Type: ☐ Object ☐ Link ☐ Object and Link  
☐ Event

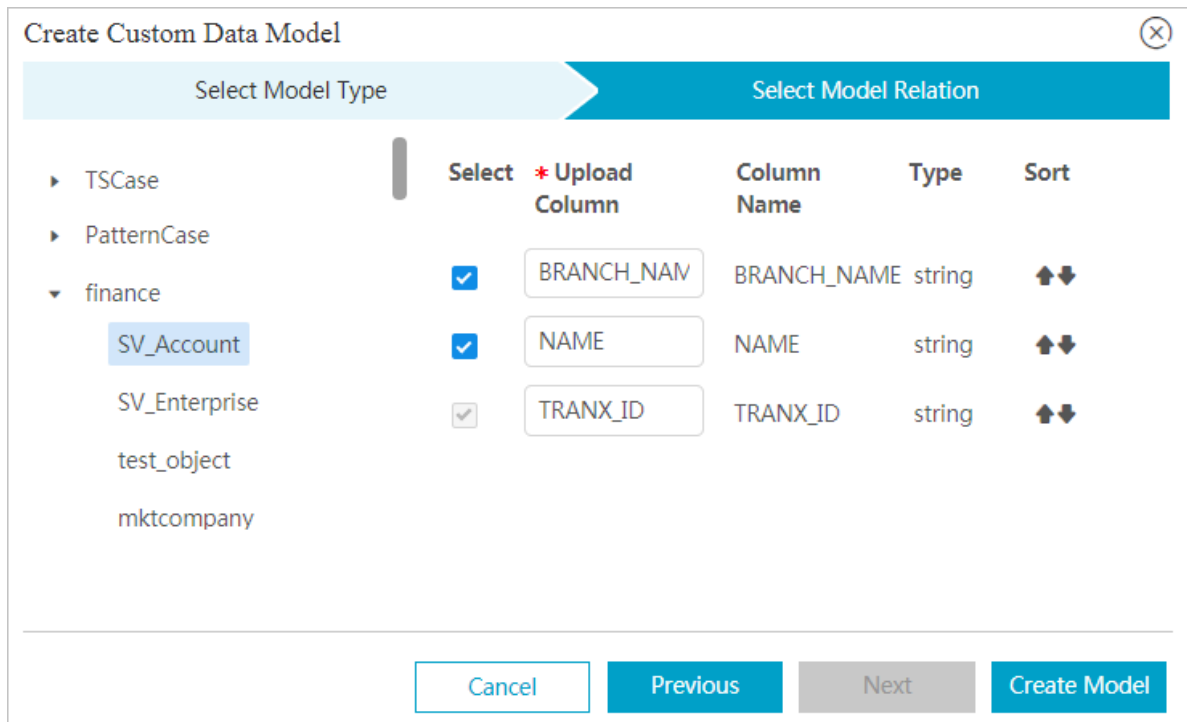
\* Model Name:

Cancel Previous Next Create Model

5. Click Next.

**6. Set the model relation in the Create Custom Data Model dialog box.**

Select an object, link, or event on the left side, and select the model columns on the right side based on the data to be imported. The columns of the primary key type are selected by default and cannot be operated.



The dialog box titled "Create Custom Data Model" has a close button (X) in the top right corner. It is divided into two tabs: "Select Model Type" (light blue) and "Select Model Relation" (dark blue). The "Select Model Relation" tab is active. On the left, a tree view shows a hierarchy: TSCase, PatternCase, and finance (expanded). Under "finance", the items are SV\_Account (highlighted), SV\_Enterprise, test\_object, and mktcompany. In the center, there are checkboxes for "Select" and "Upload Column". The "Select" column has checkboxes for BRANCH\_NAM, NAME, and TRANX\_ID, all of which are checked. The "Upload Column" column is empty. On the right, a table lists the selected columns with their names and types.

Select	Upload Column	Column Name	Type	Sort
<input checked="" type="checkbox"/>		BRANCH_NAM	BRANCH_NAME string	↑↓
<input checked="" type="checkbox"/>		NAME	NAME string	↑↓
<input checked="" type="checkbox"/>		TRANX_ID	TRANX_ID string	↑↓

At the bottom, there are four buttons: "Cancel", "Previous", "Next", and "Create Model".

**7. After you have configured the preceding parameters, click Create Model.**

View a model

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > Import Models**.

3. On the Import Models page, select a model and click View to view the details of the model in the Model Details dialog box that appears.

Model Details			✕	
Column Name	Column Type	Column ID		
BRANCH_NAME	Character	O00000072P0002		
NAME	Character	O00000072P0003		
TRANX_ID*	Character	O00000072P0001		
			< 1 >	
			Cancel OK	

Delete a model

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > Import Models**.
3. On the Import Models page, select a model and click **Delete**.
4. In the dialog box that appears, click **OK** to delete the model.

## 7.9.2 Configure a search item

You can configure search items to set the fields to be searched for in Analytics Workbench. After a search item is configured, it must be correlated with an object, link, or event property to take effect.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

- Before you delete a search item, you must first delete the links between the search item and the correlated properties of objects, links, and events.

## Context

This topic describes how to add, modify, and delete search items, and how to associate search items with the properties of objects, links, and events.

### Add a search item

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, choose **Advanced > Search Configurations**.
3. Click **Add Item** in the upper-left corner of the page that appears.

The parameters are described in [Table 7-31: Search item configuration parameters](#).

The screenshot shows the 'Search Configurations' interface. At the top left, there is a '+ Add Item' button. The main area contains a table with the following columns: 'Search Item Name', 'Search Item Type', 'Advanced Correlated Items', and 'Show in Main Search Box'. The table lists several search items, each with a trash icon for deletion. The first row is highlighted with a red box, showing an empty name field, a 'String Like' type, no correlated items, and a checked checkbox. Other rows show items like 'Name', 'IDcard\_num', 'BRANCH\_NAM', 'Account\_name', and 'Enterprise\_nan' with their respective types and correlated items.

Table 7-31: Search item configuration parameters

Parameter	Description
Search Item Name	Search Item Name is customized by the user. We recommend that you set the name according to the properties of the objects, links, or events that need to be correlated. For example, a mobile phone number, an ID number, or a person's name.
Search Item Type	The Search Item Type is used to set the data type that is supported by this search item. The Search Item Type must be consistent with the Query Type of the property of the object, link, or event to be correlated with.



Parameter	Description
Advanced Correlated Items	Advanced Correlated Items are used to group multiple search terms to search for data. You can select multiple configured search terms.
Show in Main Search Box	Sets whether the search item is displayed in the Search page of Analytics Workbench.

4. Click Add Item to add multiple search items as needed.
5. Click Save.

After a search item is added, you must correlate the search item to the properties of an object, link, or event. For more information, see [Correlate a search item to properties](#).

Correlate a search item to properties

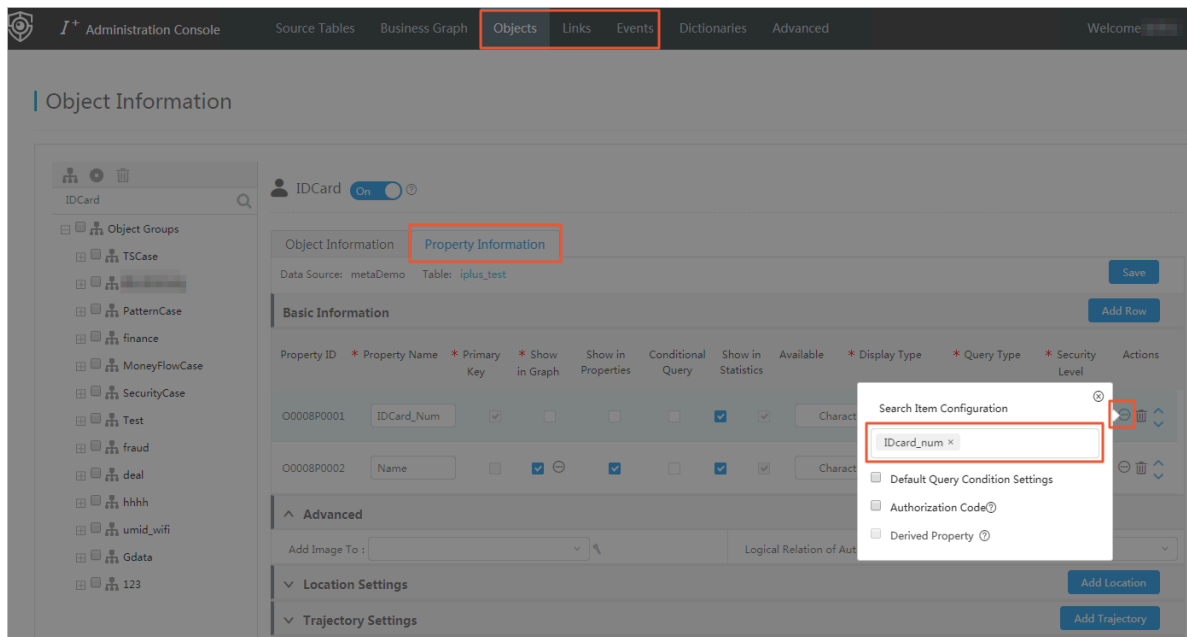
The configuration entries are as follows:

- **Objects:** Administration Console > Objects > selected the object to be correlated to > Property Information > Basic Information > more configurations.
- **Links:** Administration Console > Links > select the link to be correlated to > Property Information > Basic Information > more configurations.
- **Events:** Administration Console > Events > select the event to be correlated to > Property Information > Basic Information > more configurations.

This topic describes how to correlate a search item with the properties of an object as an example. You can correlate a search item with the properties of a link or event by using similar methods.

1. [Log on to Administration Console of Graph Analytics](#).
2. In the top navigation bar, click Objects.
3. In the left-side navigation pane of the Object Information page, click the name of the object to be configured and then click the Property Information tab on the right side.

4. In the Basic Information area, click the More icon (⋮) next to the property to be correlated, and then select the configured search items in Search Item Configuration.



5. After you have configured these parameters, click the (✕) icon to close the configuration box.
6. Click Save to save the configurations.

Modify a search item

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > Search Configurations**.
3. Modify the parameters of a search item as needed.
4. Click **Save**.

Delete a search item

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > Search Configurations**.
3. Click the (🗑️) icon to delete the corresponding search item.
4. Click **Save**.

## 7.9.3 System settings

### 7.9.3.1 Configure components

You can configure the functional components to enable or disable functions in Analytics Workbench and Administration Console and set basic information of Analytics Workbench. When a functional component is disabled, this function will not be displayed in the operation area.

#### Prerequisites

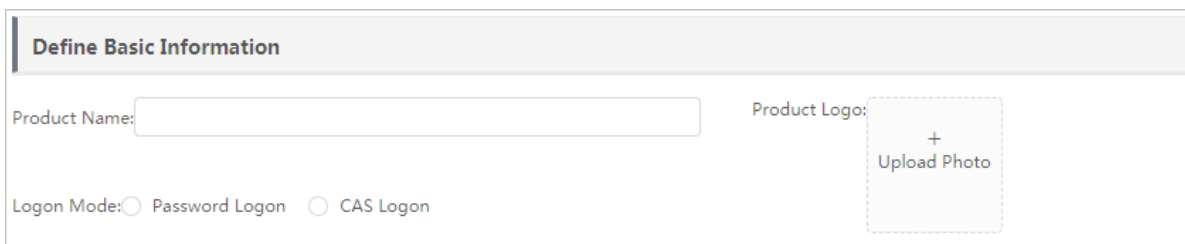
Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Context

The functional component settings take effect globally. Enable or disable components as needed with caution.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Functional Components**.
3. **Define basic information:** Set the Product Name, Product Logo, and Logon Mode parameters for Graph Analytics.



#### Note:

If you do not set the Product Name parameter and the Product Logo parameter, the default name and default Logo image are used.

If the CAS (Central Authentication Server) server is deployed in your environment, you can select CAS logon as the logon mode, as shown in [Figure 7-28: CAS logon](#).



#### Note:

**CAS is a single sign-on protocol. If you select CAS logon as the logon mode, you only need to log on once to access all trusted application systems.**

Figure 7-28: CAS logon

The screenshot shows the 'Define Basic Information' configuration page. It includes fields for 'Product Name', 'Product Logo' (with an 'Upload Photo' button), 'Logon Mode' (with radio buttons for 'Password Logon' and 'CAS Logon', where 'CAS Logon' is selected and highlighted with a red box), 'CAS Back-to-Graph Analytics URL', 'CAS Auth URL', and 'CAS Logon URL'.

- 4. Enable and disable functional components: Select the modules to be enabled and cancel the modules to be disabled.**

**If you select or clear a parent component, the child components will be automatically selected or cleared.**

The screenshot shows the functional components configuration page. It lists various components with checkboxes to enable or disable them. The components are grouped into sections: Search (On), Intelligent Search, Link Analysis (On), Management Console (On), OLP Configuration (Object Information, Link Information, Property Groups, Dictionary Settings, Source Data, Configuration View, Import Models, Event Information), User and Role Management (Organization and User Management, Role Management, Authorization Code), System Configuration (Functional Component Settings, Technical Parameter Settings, Business Parameter Settings, Object Icon Management, Search Configurations, Service List), System O&M (System Log), User System Settings (On), Intelligent Network (On), Open Platform (On), Network Analysis (On), and API Function (On). Each section has a 'Hide' button.

- 5. Click Save.**

## 7.9.3.2 Technical parameters

### 7.9.3.2.1 Path analysis settings

**In Graph Analytics, you can set the highest link degree of path analysis. Also, you can set whether to calculate only the nodes that are of the same type as the selected nodes.**

#### Prerequisites

**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

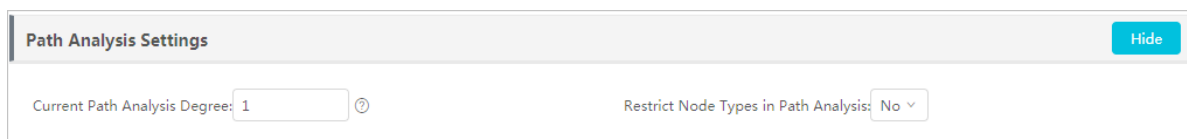
## Context

Typically, the default values are used for path analysis settings. Exercise caution when you modify the settings.

## Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Technical Parameters.**
3. In the Path Analysis Settings area, you can set the parameters based on your requirements.

The parameters are described in [Table 7-32: Path analysis configuration parameters.](#)



Path Analysis Settings Hide

Current Path Analysis Degree:  ⓘ

Restrict Node Types in Path Analysis:  ▾

Table 7-32: Path analysis configuration parameters

Parameter	Description
Current Path Analysis Degree	<p>The highest link degree supported by path analysis. The default value is 2 and the maximum value is 3.</p> <p>If the Current Path Analysis Degree parameter is set to <math>N</math>, only links of degree <math>N</math> or lower degrees will be analyzed. Links of a degree higher than <math>N</math> will not be analyzed.</p>
Restrict Node Types in Path Analysis	<p>Valid values:</p> <ul style="list-style-type: none"><li>• Yes: Only nodes of the same type as the selected node are calculated.</li><li>• No: All types of nodes are calculated.</li></ul> <p>The default value is No.</p>

4. Click **Save** in the upper-right corner.

### 7.9.3.2.2 Quick extension settings

Graph Analytics, you can set the highest link degree for quick extension.

## Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

## Context

Typically, the default values are used for quick extension settings. Exercise caution when you modify the settings.



### Note:

If the degree is set too high, the running performance of the system will be affected.

## Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Technical Parameters.**
3. In the Quick Extension Settings area, you can set the Extension Degree parameter based on your requirements.

**Extended Degree:** The highest link degree for quick extension. The default value is 2. If the Extended Degree parameter is set to  $N$ , only links of degree  $N$  or lower degrees will be analyzed. Links of a degree higher than  $N$  will not be analyzed.

4. Click **Save** in the upper-right corner.

### 7.9.3.2.3 Maximum node settings

In Graph Analytics, you can set the maximum number of nodes to be queried at the same time when you perform an analysis.

## Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

## Context

Typically, the default value is used for the maximum number of nodes to be queried. Exercise caution when you modify the number of nodes.

## Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Technical Parameters.**
3. In the **Maximum Node Settings** area, you can configure the parameters based on your requirements.

4. After you have completed the configurations, click **Save** in the upper-right corner.

### 7.9.3.3 Business parameters

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > System Settings > Business Parameters** from the top navigation bar. The **Business Parameters** page appears.

#### 7.9.3.3.1 Add double-click link settings

In Analytics Workbench, you can double-click an object to query the relationship between objects. In Administration Console, you can custom the relationships of an object to be queried when you double-click the object. If no custom configuration is set, all the first-degree relationships of the object are queried by default when you double-click the object.

#### Prerequisites

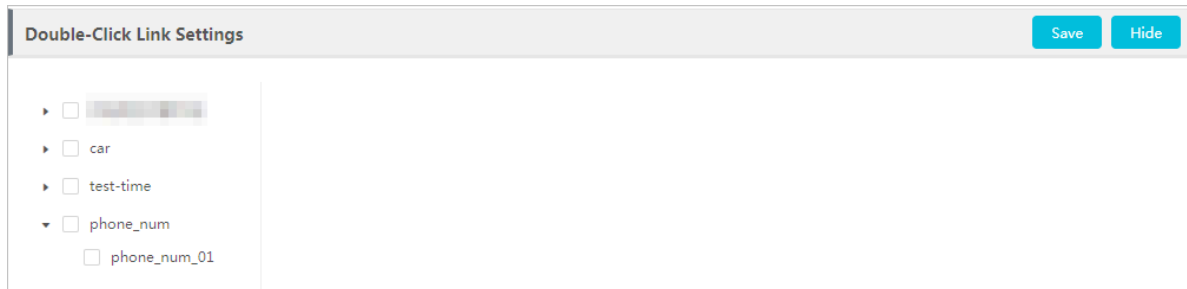
- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- The object has been created and referenced by the link.

#### Context

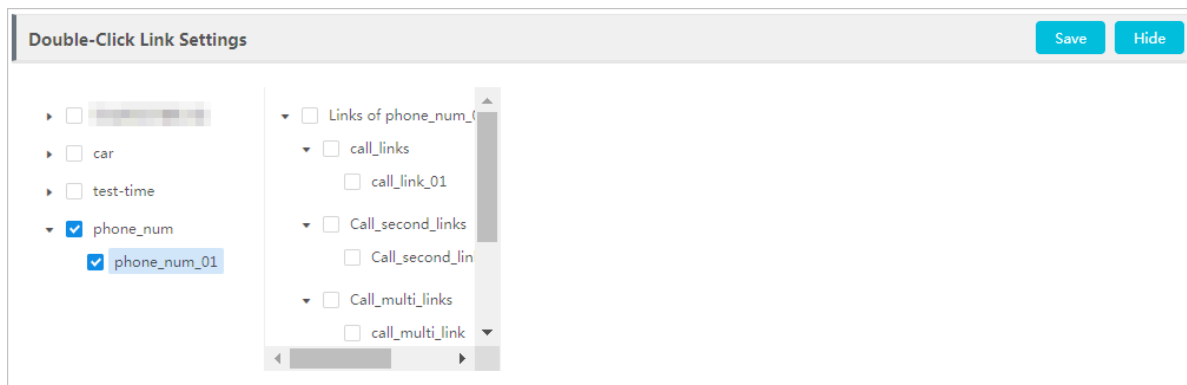
The double-click link settings take effect globally. Exercise caution when you configure the settings.

#### Procedure

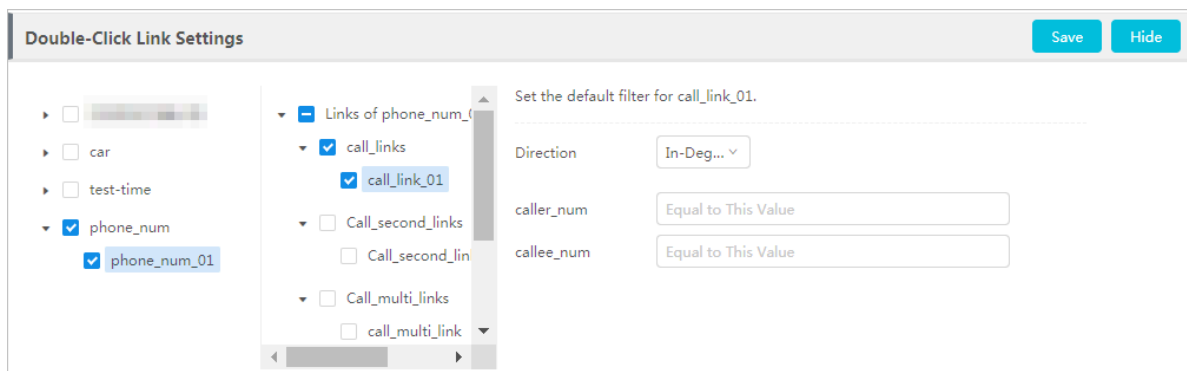
1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Business Parameters.**
3. In the **Double-Click Link Settings** area, select the objects to be configured in the left-side list.



4. In the list that is displayed on the right side of the area, select the links to be queried.



5. In the displayed area, set the filter conditions for querying the link.



6. If you need to query other links by double-clicking the object, you can continue to select other links and set the filter conditions.
7. Click **Save** to complete the double-click link settings of the object.



### 7.9.3.3.2 Double-click-disabled object settings

In Analytics Workbench, you can double-click a node to query the relationship network of the node. Administration Console allows you to enable or disable double-clicking on a specified object. By default, double-clicking is supported for all objects.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created an object.

#### Context

The double-click-disabled object settings take effect globally. Exercise caution when you configure the settings. We recommend that you disable the double-click operations on objects that generate large amounts of data.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Business Parameters.**
3. In the Double-Click-Disabled Object Settings area, click the Double-Click-Disabled Objects drop-down list, and select the objects.



4. Click Save.

### 7.9.3.3.3 Object grouping settings

Graph Analytics allows you to set the grouping conditions, so that objects or links of the same type can be grouped automatically when their quantity reaches

the threshold value. In a complex graph analytics, we recommend that you set reasonable grouping conditions to keep the analysis graph concise and clear.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created an object.

### Context

The Object Grouping Threshold and Object Grouping Degree parameters are described as follows:

- **Object Grouping Threshold:** If the number of object nodes of the same type that are connected to the same node exceeds the threshold, the connected nodes are grouped into one folder.
- **Object Grouping Degree:** If more than one node has the link of the specified degree with the same nodes, these nodes will be grouped into a folder. For example, set Object Grouping Degree to 2. If both node A and node B have second-degree links with nodes C and D, node A and node B will be grouped into one folder.

The value of Object Grouping Threshold defaults to 0, and the value of Object Grouping Degree defaults to 1. The default value indicates that no objects will be grouped automatically.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Business Parameters**.

3. In the Object Grouping Settings area, specify Object Grouping Threshold and Object Grouping Degree based on your requirements.

Object Name	Object Grouping Threshold	Object Grouping Degree
Source_account	0	1
mktclient	0	1
identity_card	0	1
wifi_fin	0	1

4. Click Save.

### 7.9.3.3.4 Configure lineage analysis

Before you perform a lineage analysis on a specified object node in Analytics Workbench, you need to configure the business parameters for this type of object in Administration Console.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- The object has been created and referenced by a link.

#### Context

Lineage analysis extends a specific business link to multiple degrees. The specific link typically refers to the lineage link and the same residence number.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Business Parameters**.
3. In the Lineage Analysis Settings area, click **Add** to add a new setting.
4. Select the object to be analyzed based on your requirements, and then select the object-related link to be analyzed.

Lineage Analysis Settings		Save	Add	Hide
Select Object:	phone_num_01	Select Object-Related Links:	call_link_01	

5. Click Save.

### 7.9.3.3.5 Intimacy measurement settings

You can set the weight for the relationship between specified objects, so that you can directly view the intimacy between these objects in the analysis result.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- An object has been created and has been referenced by a link.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Business Parameters**.
3. On the left side of the Intimacy Measurement Settings area, select an object and then click **Add Link Configuration** on the right side of the area.
4. Select a link type in the **Link Type** drop-down list, and then set the intimacy weight and other parameters.

Link Type	Origin Algorithm	Weight	Actions
Traffic_accident	<input type="checkbox"/>	0	Edit Delete
<input type="text"/>	<input type="checkbox"/>	0.0	Save Cancel

The star icon in front of an object indicates that the object has configured intimacy measurement settings, as shown in ID Card in the figure above.

5. After you have configured these parameters, click Save.
6. To add intimacy measurement settings for another Link Type, click **Add Link Configuration**.

To add intimacy measurement settings for another object, select the object in the left-side navigation pane, and then click **Add Link Configuration**.

7. After you have configured the intimacy measurement settings for all objects, click **Save** on the right side of the Intimacy Measurement Settings section.

### 7.9.3.3.6 Redirect URL settings

When you need to redirect to an external system from Graph Analytics, you can configure the URLs in External System Redirect URL Settings.

### 7.9.3.4 Object icons

#### 7.9.3.4.1 Upload an object icon

You can upload a local object avatar to the icon library in Graph Analytics.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- Make sure that you have prepared an avatar image in the PNG format. The recommended size is 32px \* 32px. We recommend that you limit the size to 320px \* 320px.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Object Icons**.
3. On the Object Icons page, click **Upload Icon**.
4. In the Upload Icon dialog box that appears, click the upload area to upload a local icon, and then specify the Name column.
5. Click **OK** to upload the object icon.

#### 7.9.3.4.2 Modify an object icon


You can modify the avatar image or the avatar name based on your requirements.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- Make sure that you have prepared an avatar image in the PNG format. The recommended size is 32px \* 32px. We recommend that you limit the size to 320px \* 320px.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top menu bar, choose **Advanced > System Settings > Object Icons**.

3. On the Object Icons page, move your mouse pointer over the icon, and click the **Modify icon** ()
4. In the Edit dialog box that appears, modify Name or upload a new avatar image.
5. Click OK.


### 7.9.3.4.3 Delete an object icon

You can delete the object icons that are no longer used.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- Before you delete an object icon, make sure that this icon is not used by another object.

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. In the top navigation bar, choose **Advanced > System Settings > Object Icons**.
3. On the Object Icons page, move your mouse pointer over the icon, and click the **Delete icon** ()
4. In the dialog box that appears, click OK to delete the object icon.

## 7.9.4 System labels

### 7.9.4.1 Create a group

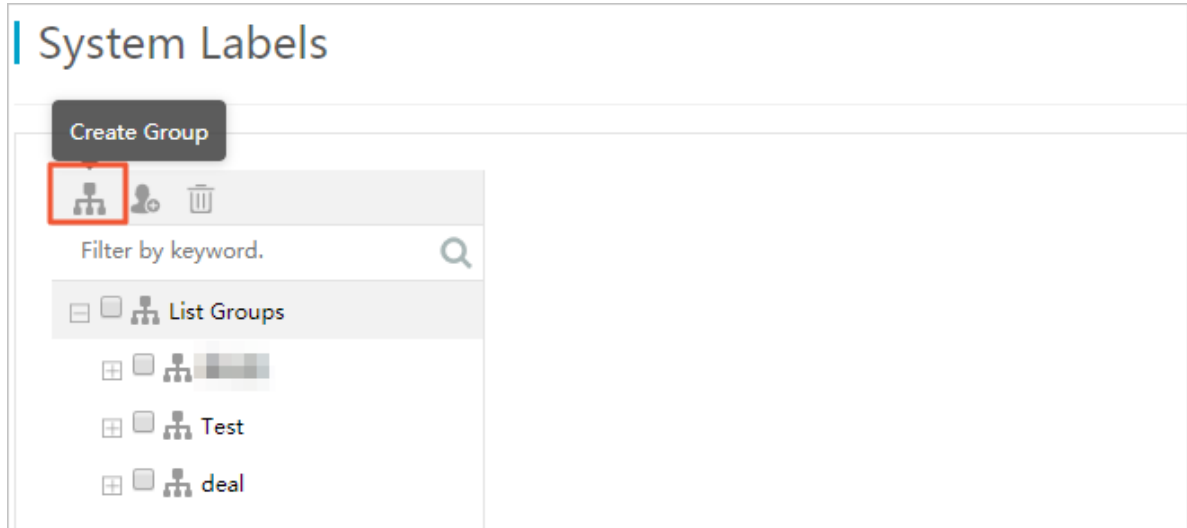
#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > Users and Permissions > System Labels** from the top navigation bar.

3. On the System Labels page, click the Create Group icon, as shown in [Figure 7-29](#):

*Create a group.*

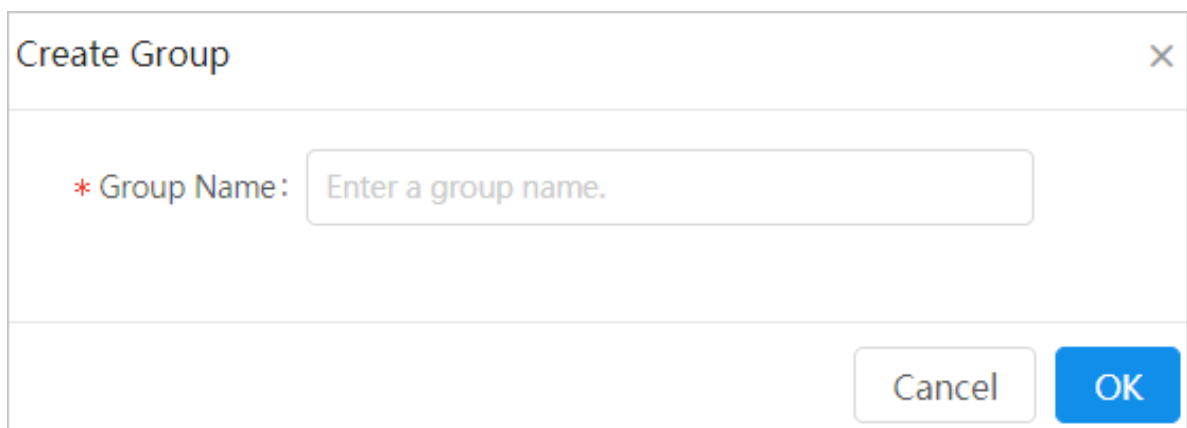
Figure 7-29: Create a group



4. In the Create Group dialog box that appears, enter a group name, as shown in

[Figure 7-30](#): *Enter a group name.*

Figure 7-30: Enter a group name



5. Click OK.

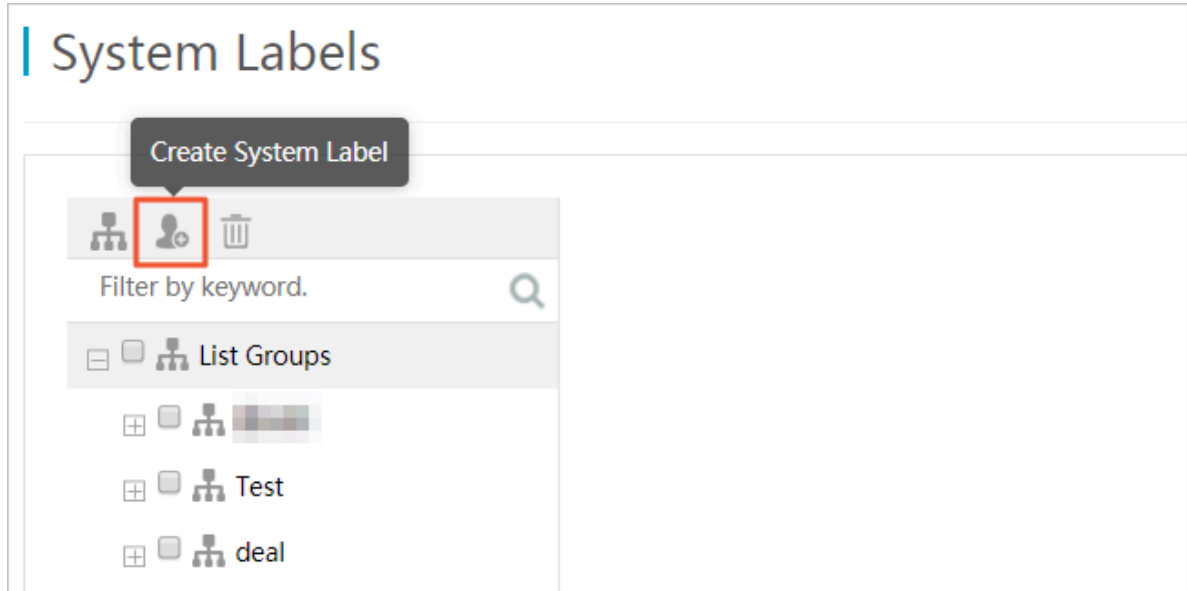
## 7.9.4.2 Create a system label

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > Users and Permissions > System Labels** from the top navigation bar.

3. On the System Labels page, click the Create System Label icon, as shown in [Figure 7-31: Create a system label](#).

Figure 7-31: Create a system label





4. In the **Configure System Label** dialog box, configure the required parameters, as shown in *Figure 7-32: Configure a system label*.

Figure 7-32: Configure a system label

**Configure System Label** [X]

\* System Label Na.. Enter a system label name.

\* Group: Select a group ▼

Rule Definition: Select a rule definition.

Label Color: Select a label color. ▼

\* Object Type: Select the object type. ▼

\* Data Sources: Select a data source. ▼

\* Table: Select a table. ▼

Object Type Filter-... Select an object type filter-by column. ▼

Cancel OK

The system label parameters are described as follows:

- **System Label Name:** the alias of the system label.
- **Group:** the group to which the system label belongs.
- **Rule Definition:** The processing rule that is applied when an object matches the system label. Value options include Not Display Object, Not Extend Object, Not Display Properties, Disable Property Statistics, and

Disable Behavior Display. **If you have selected Not Display Object, the other four options and the Label Color parameter become unavailable.**

- **Label Color:** the display color of the system label.
- **Object Type:** all available objects are displayed in the drop-down list. Choose an object as needed.

After you select the object type, a primary key parameter will appear in the dialog box. Select a column corresponding to the primary key property as needed.

- **Data Source:** the data source where the system label belongs.
- **Table:** the table where the system label belongs.
- **Object Type Filter by Column, Object Type Filter Operator, and Object Type Filter Value: Optional.** You can use these parameters to filter system labels. If you have specified any one of these parameters, the other two parameters are required.

5. Click OK.

### 7.9.4.3 Modify a system label

#### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > Users and Permissions > System Labels** from the top navigation bar.
3. On the **System Labels** page, click an object from the left-side navigation pane.
4. In the details area on the right side of the page, select a system label, and click **Edit**, as shown in [Figure 7-33: Modify a system label](#).

Figure 7-33: Modify a system label

<input type="checkbox"/>	System Label	Data Sources	Table	id	Object Type Filter-by Column	Object Type Filter Operator	Object Type Filter Value	Actions
<input type="checkbox"/>	User_ID		POC_YJPT_DGKH_CLEAN_sub	id				EditDelete

Delete

Create Label

< 1 >

5. Modify the parameters as needed. The object type cannot be modified.
6. Click OK.

## 7.9.4.4 Delete a system label

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > Users and Permissions > System Labels** from the top navigation bar.
3. Click **Delete** next to a system label, or select one or more system labels and then click **Delete** in the lower-left corner, as shown in [Figure 7-34: Delete a system label](#).

Figure 7-34: Delete a system label

<input type="checkbox"/>	System Label	Data Sources	Table	id	Object Type	Filter-by Column	Object Type Filter Operator	Object Type Filter Value	Actions
<input type="checkbox"/>	User_ID	skyview	POC_VJPT_DGKH_CLEAN_sub	id					EditDelete

Delete

< 1 >

## 7.9.5 System operations and maintenance

### 7.9.5.1 Audit logs

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > System O&M > Audit Log** from the top navigation bar. The **Query Audit Log** page appears.
3. Set the search filters, and then click **Query**, as shown in [Figure 7-35: Log query](#).

If you click **Query** without setting any filters, all data will be returned.

Figure 7-35: Log query

Query Audit Log

User:  IP Address:  Query Content:  Time Spent (ms):

Time Range:  Start Time ~ End Time

User ID	User Name	IP	Access Module	Started At	Ended At	Status	Cause of Failure	Time Spent	Details
No data									

## 7.9.6 View server clusters

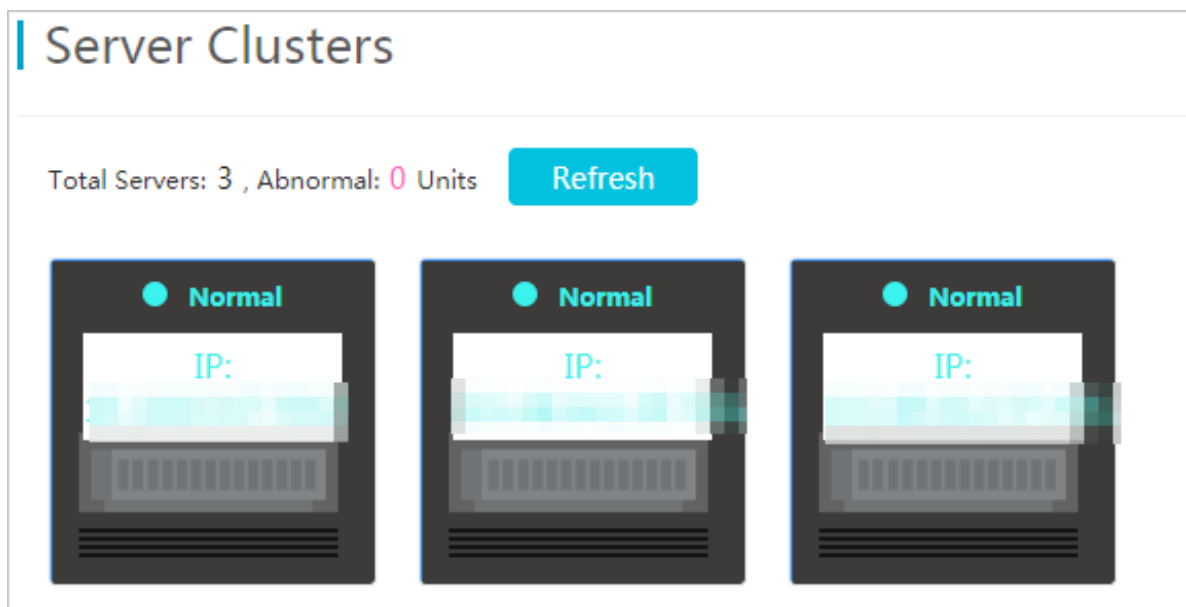
In Administration Console, you can view the information of all servers in a cluster, including the running status, server exceptions, the number of servers, IP addresses, and port numbers.

### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

### Procedure

1. [Log on to Administration Console of Graph Analytics.](#)
2. Choose **Advanced > Server Clusters** to go to the Server Clusters page.



In Administration Console, you can view the information of all servers in a cluster, including the running status, the number of servers and server exceptions, IP addresses, and port numbers.

3. If you have stayed on the current page for too long, you can click Refresh to view the latest information.

## 7.10 Import data

### 7.10.1 Model list

#### 7.10.1.1 Model overview

A model is a template used to import data to Graph Analytics. You can use models to import individual or small amounts of data to Graph Analytics.

Models include custom models and system models:

- Custom models are directly created by users on Analytics Workbench. Only the creator can view, download, modify, and delete the custom models.
- System models are created by the administrator in Administration Console. All users can view and download system models on Analytics Workbench. However, system models cannot be modified or deleted. To delete a system model, you must log on to Administration Console.


#### 7.10.1.2 View models

In Analytics Workbench, you can view information about the existing data models, including the model type, mapped OLEP, properties, and the property type. This helps you understand the existing models at any time and identify the data model that matches the data to be imported.

#### Prerequisites

All users can view the system models. However, to view a custom model, you must have the account and password of the user that created the model.

#### Procedure







1. [Log on to Analytics Workbench.](#)
2. Click the  icon in the upper-right corner of the page, and click the Model List tab. On the Model List page, you can view the creation time, model type, and mapped OLEP for all models.




**Note:**

Analytics Workbench supports custom and system models. Custom models are created by users on Analytics Workbench. For more information, see [Create](#)

*models*. System models are configured in Administration Console. For more information, see [Manage a system model](#).

Data List		Model List			Create Model
Model Name	Created At	Type	Mapped OLP	Type	Actions
 calllink	January 28, 2019 2:24:49 PM GST	Link	call_link_01	Custom	  
phone_num	January 22, 2019 5:17:00 PM GST	Object	phone_num_01	System	 

- Click the  icon next to the data model that you want to view, and view the property information about the data model in the dialog box that appears.

Model : calllink ✕

Property Name	Required	Property Type
callee_num*	Yes	Character (string)
caller_num*	Yes	Character (string)

Cancel Download Template File

If the data model meets the requirements for your data import, you can download this model and use this model as a template to sort the data to be imported.

- Optional: Click Download Template File to download the model file in the XLSX format.


### 7.10.1.3 Create models

The model list defines a column format for the uploaded file. The columns of the uploaded file will be mapped to the corresponding properties, including the object property, link property, object and link property, and event property. If the existing models cannot meet the requirements of the data to be imported, you must create a data model based on the target data.

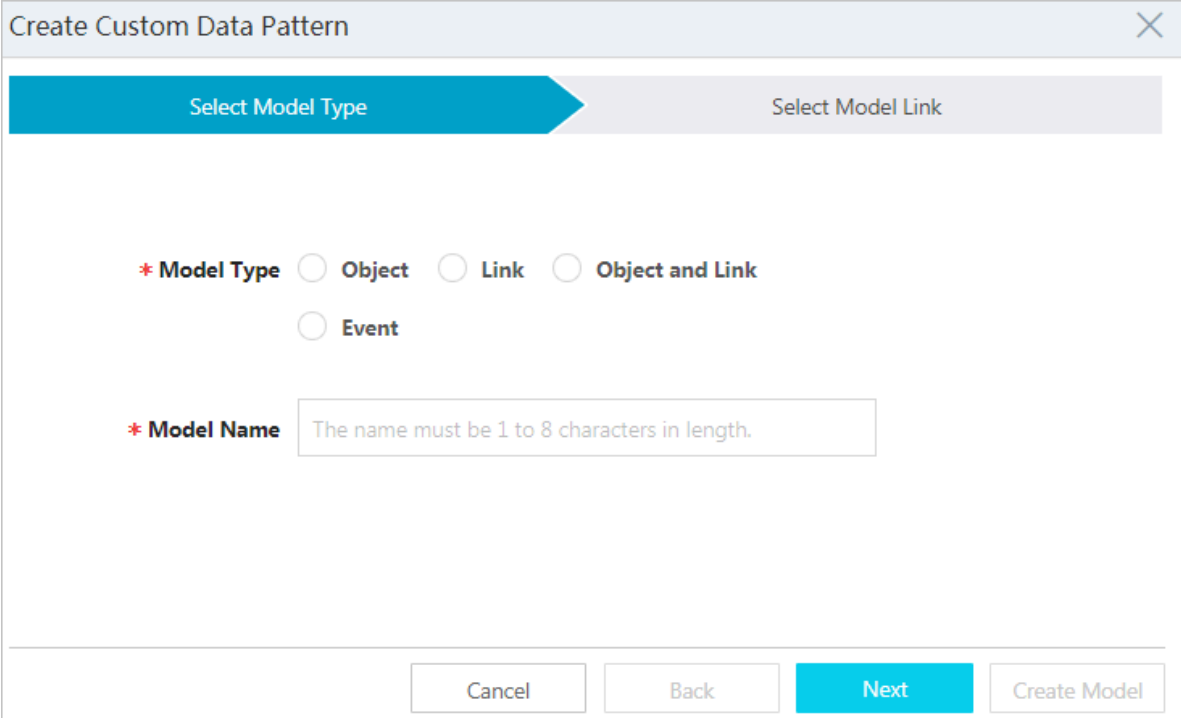
#### Prerequisites

You have obtained the account and password with the Import Data permissions.

## Procedure

1. [Log on to Analytics Workbench](#).
2. Click the  icon in the upper-right corner of the page, and then click the Model List tab. The Model List page appears.
3. Click Create Model in the upper-right corner of the page, and set the parameters in the Create Custom Data Model dialog box that appears.

For more information about the parameter settings, see [Table 7-33: Parameter configurations](#).



The dialog box titled "Create Custom Data Pattern" has a close button (X) in the top right corner. It features a progress bar with two steps: "Select Model Type" (active, blue) and "Select Model Link" (inactive, grey). Below the progress bar, there are two sections. The first section, labeled "\* Model Type", contains four radio button options: "Object", "Link", "Object and Link", and "Event". The second section, labeled "\* Model Name", contains a text input field with a placeholder message: "The name must be 1 to 8 characters in length." At the bottom of the dialog, there are four buttons: "Cancel", "Back", "Next" (highlighted in blue), and "Create Model".

Table 7-33: Parameter configurations

Parameter	Description
Model Type	You can create multiple model types, including Object, Link, Object and Link, and Event. Specify a model type as needed.
Model Name	Enter a model name. We recommend that you enter a name that is easy to understand.

4. Click Next, select an object, link, or event, select the required columns, and then set the name of Upload Data Column.

Select	Upload Data Column	Model Column Name	Type	Sequence
<input checked="" type="checkbox"/>	BRANCH_NAME	BRANCH_NAME	string	↑↓
<input checked="" type="checkbox"/>	NAME	NAME	string	↑↓
<input checked="" type="checkbox"/>	TRANX_ID	TRANX_ID	string	↑↓

5. Click Create Model. A success message is displayed, indicating that the model has been saved.

#### 7.10.1.4 Modify model names

If a model name is obscure or does not match the model content, you can modify the model name.

##### Prerequisites


Make sure that you have obtained the account and password of the user that created the model that you want to modify.



##### Note:

In Graph Analytics, you can only modify the names of user-defined models. You are not allowed to modify the system models.

##### Procedure

1. [Log on to Analytics Workbench](#).
2. Click the  icon in the upper-right corner of the page, and click the Model List tab. The Model List page appears.



3. Click the  icon in the front of the model that you want to edit, rename the model, and then click OK.



### 7.10.1.5 Download a model

Before you import data, you must download a compatible model, which is an .xlsx file, and use this model as a template to sort the data to be imported.

#### Prerequisites

You have obtained the account and password with the Import Data permissions.

#### Procedure

1. [Log on to Analytics Workbench.](#)
2. Click the  icon in the upper-right corner of the page, and click the Model List tab. On the Model List page, you can view the creation time, model type, and mapped OLEP for all models.
3. Click the  icon next to the model that you want to download, and save the model as prompted.

#### What's next

Use this model as a template to organize the data to be imported, and then import the collected data to Graph Analytics. For more information about the detailed operations, see [Import data.](#)

### 7.10.1.6 Delete a model

You can delete the custom models that are no longer used.

#### Prerequisites

Make sure that you have obtained the account and password of the user that created the model you want to delete.





#### Note:

Graph Analytics only supports deleting user-defined models. To delete a system model, you need to log on to Administration Console. For more information, see [Manage a system model.](#)

#### Procedure

1. [Log on to Analytics Workbench.](#)

2. Click the  icon in the upper-right corner of the page, and click the Model List tab. On the Model List page, you can view the creation time, model type, and mapped OLEP for all models.
3. Click the  icon next to the model that you want to delete.

## 7.10.2 Import data

Analytics Workbench supports importing data in the csv, txt, xls, or xlsx format.

You can analyze small amounts of data or individual pieces of data that are missing from the data source.

### Prerequisites

- The import data source is added when you configure the data source [Create data sources](#).
- You have obtained an account and a password with the Import Data permissions.
- You have created a model matched to the data to be imported. For more information, see [Create models](#).
- You have organized the data to be imported according to the template file, or sorted the data in the format required by the template, such as the csv, txt, xls, or xlsx format.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Click the  icon in the upper-right corner. In the dialog box that appears, select the file to be imported.

3. Click Upload, and set the read format of the file according to the data content to be imported.

The parameter configurations are described in [Table 7-34: Parameter configurations and descriptions](#).

After the file is imported, you can see the data preview of the file in the dialog box. Only the first 10 lines of the file content and the total number of lines are displayed.

Import Data - 2. Select Model

Select a file to upload: IDcard.xlsx Upload

**Data Preview** Total: 3 Items

A0	A1	A2	A3
32	John	2010-10-12 00:00:00	male
23	Lili	1997-09-10 00:00:00	female
13	Tom	2010-10-12 00:00:00	male

Field Separator: ☒ Comma (,) ☐ Semicolon (;) ☐ Tab

☐ Pre-Filter 1 Row Table Head

**Select Model** ☒ Object ☐ Link ☐ Object and Link ☐ Event

System Model ☐ phone\_num ☐ fsd ☐ ygdf ☐ dd ☐ hhh

Custom Model ☐ PhoneNum ☒ ID\_Card Create Model

Cancel Next

Table 7-34: Parameter configurations and descriptions

Parameter	Description
Column Separator	Sets the internal column separator for each line of the content.
Encoding Method	Specifies the character encoding for the file content.
Select Model	After you have selected the model type corresponding to the uploaded data, you can see the currently available system models in the System Model tab and custom models in the Custom Model tab. If you do not have a model that matches the data to be imported, you can click Create Model to create a new model in real time. For more information, see <a href="#">Create models</a> .

4. After you have configured the parameters, click Next to set the data name and the mapping relationship between the columns and model properties of the data to be imported.

Import Data - 3. Configure Data

Selected Model - ID\_Card

Upload Data - IDcard.xlsx

IdentityCard*	Name	Birthday	Sex
3	John	2010-10-12 00:00:00	male
2	Lili	1997-09-10 00:00:00	female
1	Tom	2010-10-12 00:00:00	male

Data Name:

Back

Submit Data Import

5. After you have configured the parameters, click Submit.

## 7.10.3 Data list


### 7.10.3.1 View data

Analytics Workbench allows you to view the imported data at any time, so that you can better understand the data.





#### Prerequisites


You have obtained the account and password with the Import Data permissions.

#### Procedure

1. [Log on to Analytics Workbench](#).
2. Click the  icon in the upper-right corner of the page to go to the Data List tab page. You can view all data that has been imported.

The Data List page displays information about all imported data, including the data name, model name, upload time, OLEP type, mapped OLEP, and the import status.

Data List		Model List				
Data Name	Model Name	Upload Time	OLEP Type	Mapped OLEP	Import Status	Actions
 IDCard	IDCard	July 15, 2019 11:12:39 AM CST	Object	IDCard	Success	  

3. Click the  icon next to a data model that you want to view, and view the property information about this data in the dialog box that appears.

### 7.10.3.2 Edit a data name





If a data name is obscure or does not match the data content, you can modify the data name.

#### Prerequisites

You have obtained the account and password with the Import Data permissions.

#### Procedure

1. [Log on to Analytics Workbench.](#)
2. Click the  icon in the upper-right corner of the page to go to the Data List page.

Data List		Model List				
Data Name	Model Name	Upload Time	OLP Type	Mapped OLP	Import Status	Actions
 IDCard	IDCard	July 15, 2019 11:12:39 AM CST	Object	IDCard	Success	  

3. Click the  icon in front of the data name, rename the data, and then click OK.

### 7.10.3.3 Import data to Graph





You can use this feature to import a data file to Graph to analyze the data quickly.

#### Prerequisites

You have obtained the account and password with the Import Data permissions.

#### Procedure

1. [Log on to Analytics Workbench.](#)
2. Click the  icon in the upper-right corner of the page to go to the Data List page.

Data List		Model List				
Data Name	Model Name	Upload Time	OLP Type	Mapped OLP	Import Status	Actions
 IDCard	IDCard	July 15, 2019 11:12:39 AM CST	Object	IDCard	Success	  

3. Click the  icon next to the specified data file to present the data in Graph.

### 7.10.3.4 Delete data





You can delete unnecessary data.

#### Prerequisites

You have obtained the account and password with the Import Data permissions.

## Procedure

1. [Log on to Analytics Workbench](#).
2. Click the  icon in the upper-right corner of the page to go to the Data List page.

Data List		Model List				
Data Name	Model Name	Upload Time	OLP Type	Mapped OLP	Import Status	Actions
 IDCard	IDCard	July 15, 2019 11:12:39 AM CST	Object	IDCard	Success	  

3. Click the  icon next to the specified data to delete the data from Graph Analytics.

## 7.11 Search

### 7.11.1 Search

Search is one of the two key modules of Graph Analytics. Research staffs can use the Search module to find and view different objects, such as mobile phones or identity card information. In this topic, you can learn about the features and the entry of the Search interface.

If you have obtained some fuzzy information, you can use the Search module to find objects and link records related to the information. These records can be added as independent object nodes to the relationship analysis, and can be used as a starting point for analysis and decision-making.

In the analysis process, you can expand the information step by step and refine the analysis by keywords, such as name and address. The Search module provides a search tool that allows you to retrieve the object information and locate the target information quickly.

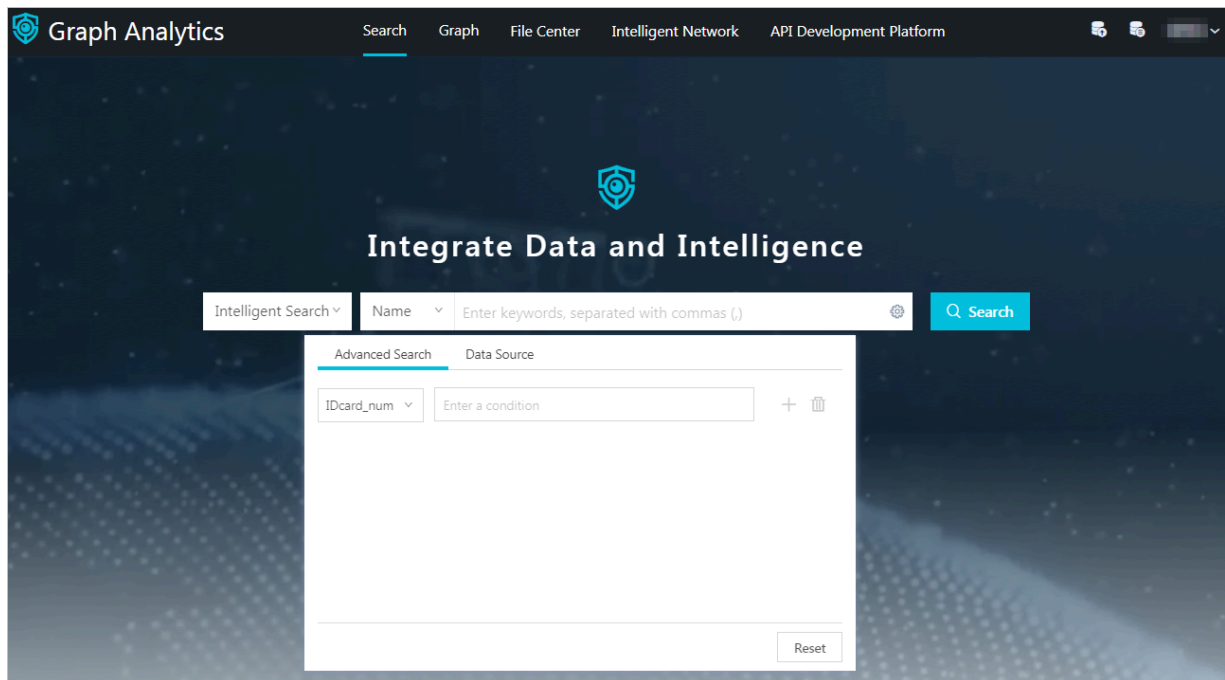


#### Notice:

In case of empty search conditions, no results will be returned, and a message Enter at least one condition value. will appear.

The Search module also serves as the entry to the Graph module. You can import the information of retrieved objects to the Graph module for further link extensions and link analyses.

Figure 7-36: Search



### 7.11.2 Simple search

You can use this feature to quickly search for objects that contain a certain type of keyword. Fuzzy search is supported.

#### Prerequisites

A search item has been configured for the target object, and the search item has been associated with the property of the object. For more information, see [Configure a search item](#).

#### Context

When you perform a simple search, you only need to select a keyword type and enter one or more keywords.

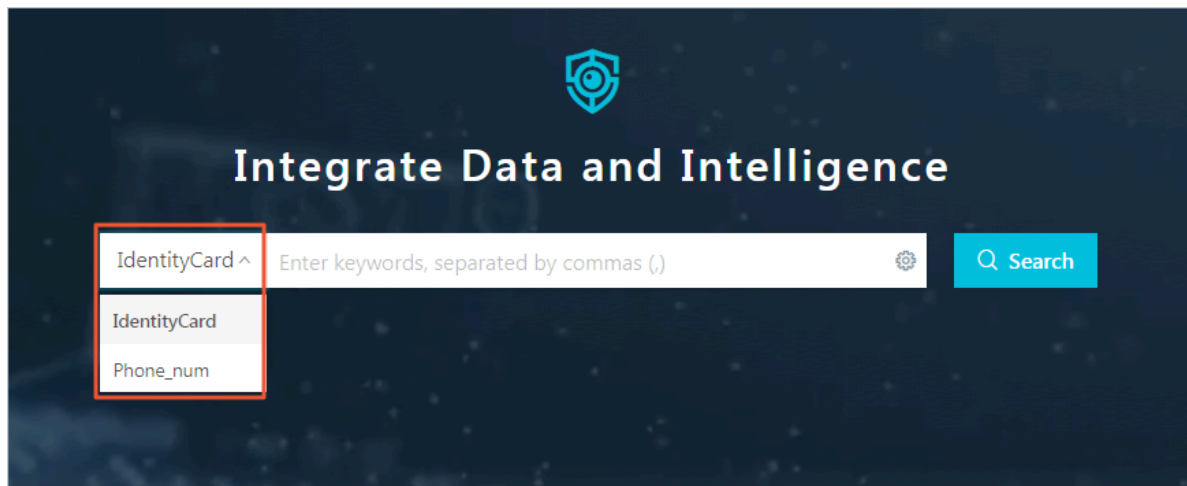
#### Procedure

1. [Log on to Analytics Workbench](#).
2. Click Search on the top navigation bar to go to the Search page.

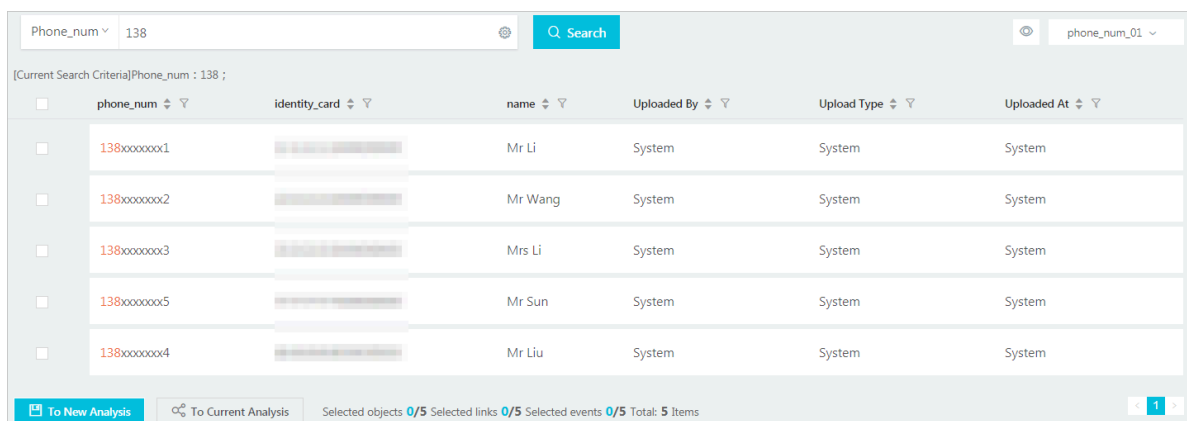
3. Select a search item in the drop-down list as a keyword type, and enter one or more key words based on the search item you select.

Fuzzy search is supported. For example, if you want to search for objects whose phone number contains 138, you can just enter 138 in the search box.

Figure 7-37: Search box



4. Click Search or press Enter to start a search.



	phone_num	identity_card	name	Uploaded By	Upload Type	Uploaded At
<input type="checkbox"/>	138xxxxxx1		Mr Li	System	System	System
<input type="checkbox"/>	138xxxxxx2		Mr Wang	System	System	System
<input type="checkbox"/>	138xxxxxx3		Mrs Li	System	System	System
<input type="checkbox"/>	138xxxxxx5		Mr Sun	System	System	System
<input type="checkbox"/>	138xxxxxx4		Mr Liu	System	System	System

### 7.11.3 Advanced search

Advanced Search supports fuzzy search and multiple search conditions.

#### Prerequisites

You have configured a search item with an advanced association item for the target object, and the search item has been associated with the property of the object. For more information, see [Configure a search item](#).

#### Context



**You can specify the search terms in Advanced Search in the same way you perform a simple search. You can specify the advanced correlated items for the selected search terms. This is similar to a combined search based on multiple keyword types. You can also specify the data source items to be searched, which is similar to specifying the search range.**

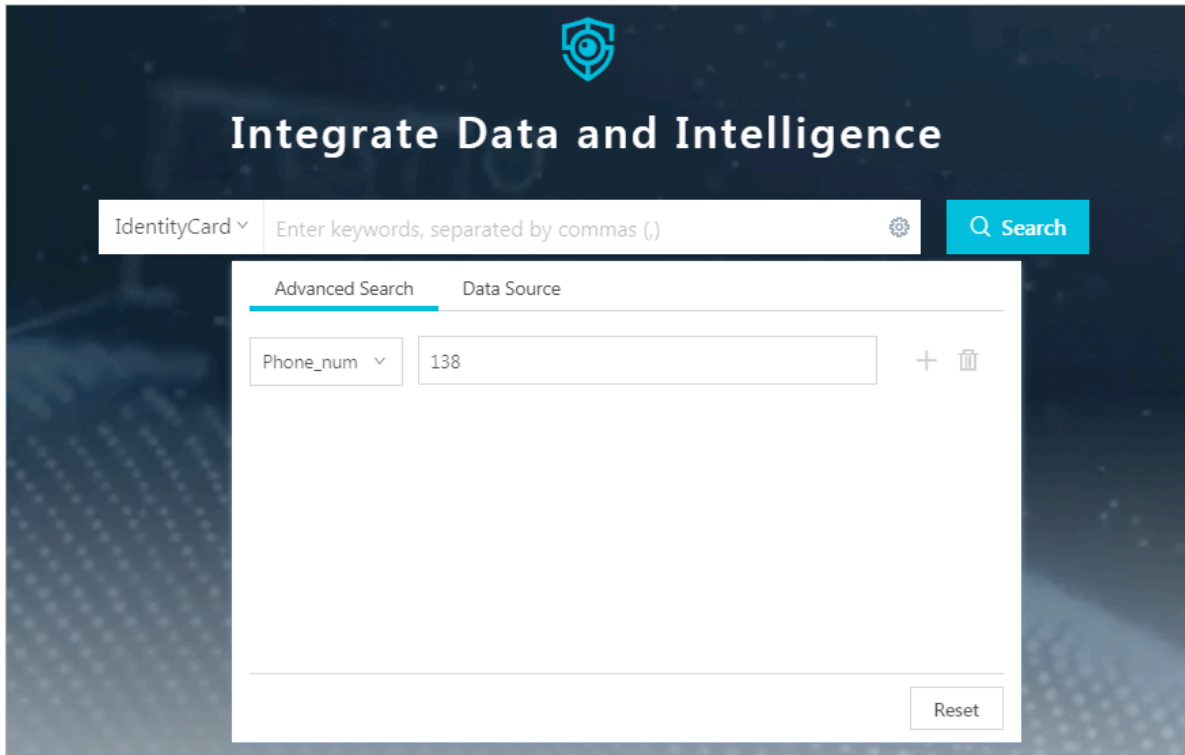
**This topic describes a sample search for objects named liqing344 whose identity card contains 234.**

### **Procedure**

1. [Log on to Analytics Workbench](#).
2. Click Search on the top navigation bar to go to the Search page.
3. Select a search item in the drop-down list as a keyword type, and enter one or more key words based on the search item you select.

**In this example, the search item is set to ID Card and the keyword is 234.**

- Click the  icon next to the search box to set the condition to perform an advanced search.

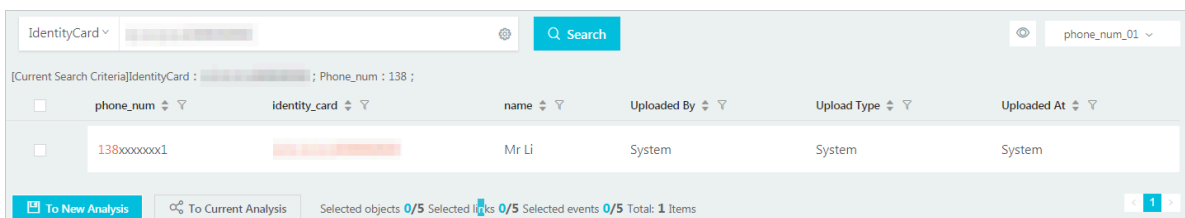


**Advanced Search displays the Advanced Correlated Items of the currently selected item. In this case, the advanced correlated item is set as Name, and the keyword is liqing344.**

**Data Source:** The data sources are the objects and links defined in Graph Analytics. Here you only need to select the data sources you want to search.

**Reset:** You can reset the search condition to the initial status.

- After you have configured these parameters, click Search or press Enter to start a search.



## 7.11.4 View and analyze search results

After you have completed a search, you can view the search results and send the specified content to the graph to perform an analysis.

### Prerequisites

You have completed a search, and the search results are not empty. For more information, see [Simple search](#) or [Advanced search](#).

## Procedure

1. [Log on to Analytics Workbench](#).
2. Search for objects and view the search results. For more information, see [Simple search](#) or [Advanced search](#).

The search results for this example are as follows:

Phone\_num ▾138

Search

phone\_num\_01 ▾

[Current Search Criteria]Phone\_num : 138 ;

phone\_num ▾ ▾

identity\_card ▾ ▾

name ▾ ▾

Uploaded By ▾ ▾

Upload Type ▾ ▾

Uploaded At ▾ ▾

138xxxxxx1

Mr Li

System

System

System

138xxxxxx2

Mr Wang

System

System

System

138xxxxxx3

Mrs Li

System

System

System

138xxxxxx5

Mr Sun

System

System

System

138xxxxxx4

Mr Liu

System

System

System

To New Analysis

To Current Analysis


Selected objects 3/5

Selected links 0/5

Selected events 0/5

Total: 5 Items

1

3. Select part of or all of the search results. A total of 10 records are displayed on the current page. Click the  icon in the upper-right corner. The search results only show the information of the selected objects.

Phone\_num ▾138

⚙

🔍 Search

🔍

phone\_num\_01 ▾

[Current Search Criteria]Phone\_num : 138 ;

<input checked="" type="checkbox"/>	phone_num ▴ ▾	identity_card ▴ ▾	name ▴ ▾	Uploaded By ▴ ▾	Upload Type ▴ ▾	Uploaded At ▴ ▾
<input checked="" type="checkbox"/>	138xxxxxx1		Mr Li	System	System	System
<input checked="" type="checkbox"/>	138xxxxxx2		Mr Wang	System	System	System
<input checked="" type="checkbox"/>	138xxxxxx3		Mrs Li	System	System	System

📄 To New Analysis

🔗 To Current Analysis

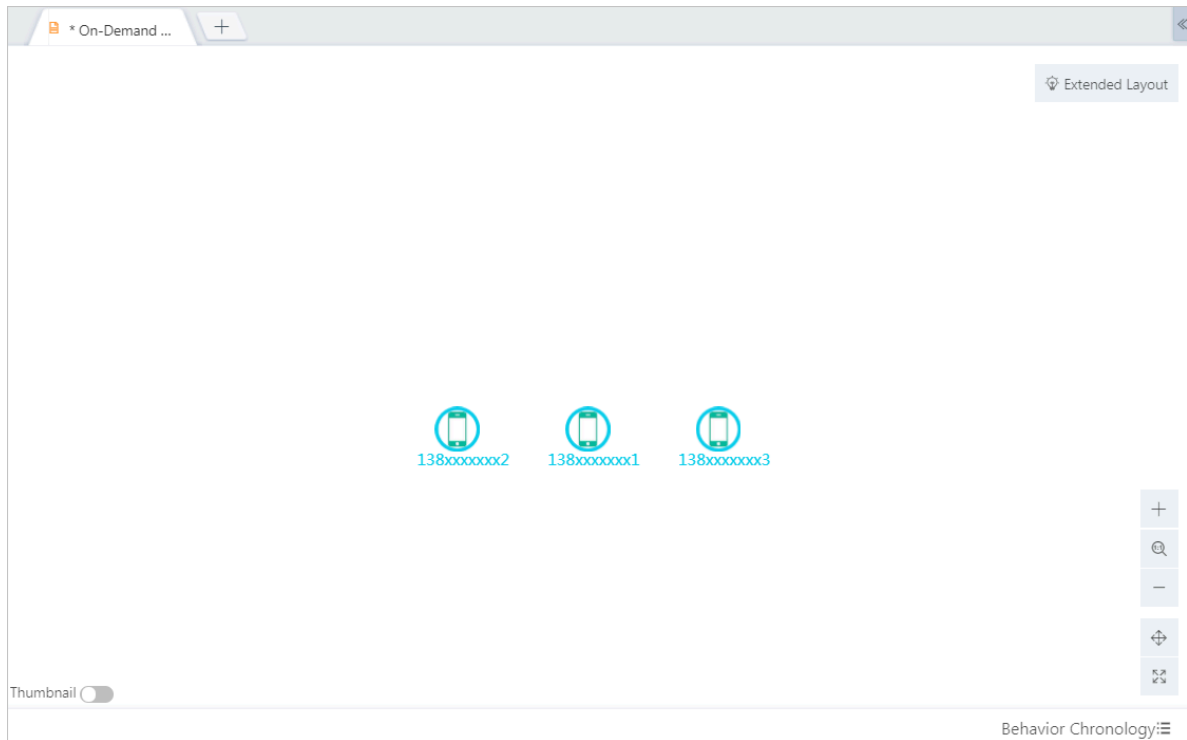
Selected objects 3/5 Selected links 0/5 Selected events 0/5 Total: 3 Items

<

1

>

4. Click the To New Analysis icon in the lower-left corner, or the To Current Analysis icon. Send the selected search results to the graph to perform an analysis.



## 7.12 Graph

### 7.12.1 Graph

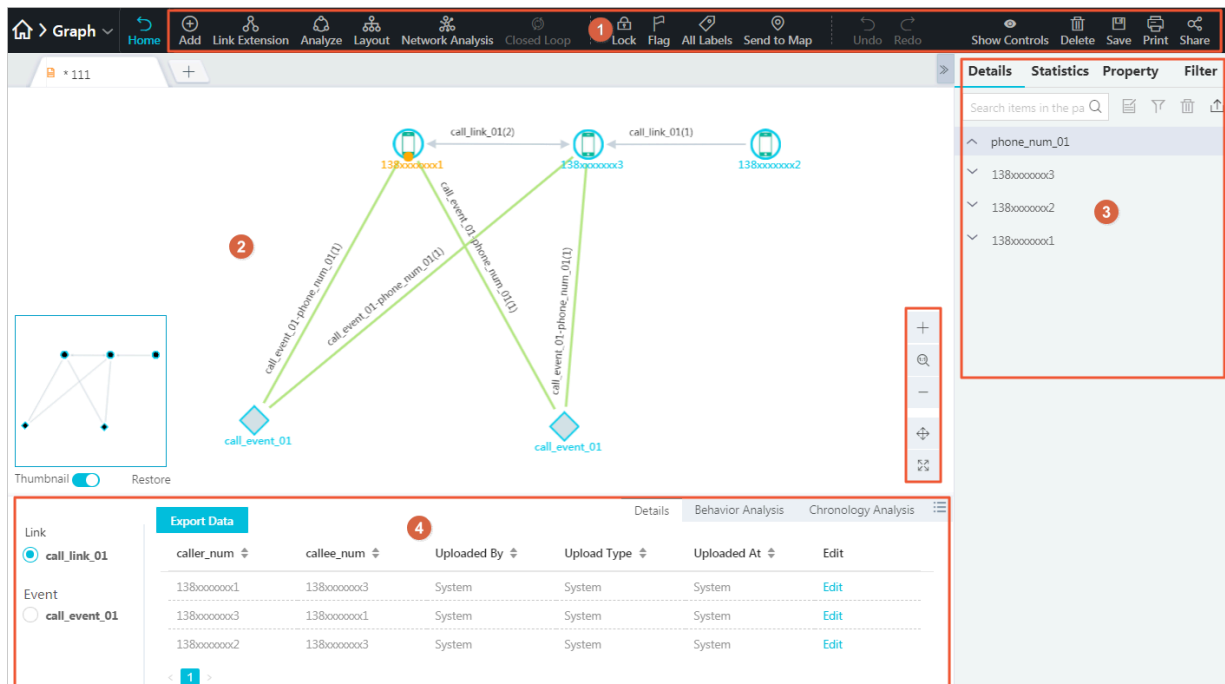
**Graph is the operation interface of Analytics Workbench. You can create an analysis and view the analysis results in Graph. In this topic, you can learn about the features, function modules, and other information of Graph.**

**Graph is the core module of Analytics Workbench. This module covers most of the analysis and decision-making scenarios. The Graph module displays the relationship topologies among all objects. You can perform business computing and interactive graphic operations among multiple objects, and arrange visualized layouts based on your needs.**

**Meanwhile, user spaces and information cube networks are used as the assisting components to cover a wider range of analysis scenarios.**

The Graph interface includes the following four areas.

Figure 7-38: Graph



- Area 1: functions
- Area 2: main graph area
- Area 3: properties and statistics
- Area 4: behavior analysis and chronology analysis

## 7.12.2 Analysis types

Graph Analytics supports four types of analyses: temporary analysis, common analysis, shared file analysis, and import data analysis.

### Temporary analysis

An analysis that has just been created by the user is known as a temporary analysis. Temporary analyses have the same operations and features as other types of analyses, but temporary analyses cannot be shared before they are saved.

### Common analysis

After a new analysis is saved and opened again, it is called a common analysis. The common analysis is the most commonly used analysis in Graph Analytics.

### Shared file analysis

The analysis of shared files has the following statuses:

- **Shared file - initial file**

After a common analysis is shared, it becomes a shared file. Each shared file will generate an initial file that is consistent with the original common analysis.

- **Shared file - history analysis (draft)**

After the initial file has been edited by the shared member, a history analysis, also known as draft analysis, will be generated.

- **Shared file - merged file**

The system automatically merges multiple historical analyses into one analysis. Users can also merge the analyses manually.

#### Import data analysis

The import data analysis imports data in the Data List to the graph area to perform analyses. This type of analysis cannot be saved.

### 7.12.3 Create analyses

After you log on to Analytics Workbench, you must create an analysis and add the objects to be analyzed as nodes before you analyze the nodes.

#### Prerequisites

- Make sure that you have obtained an account and a password with the permission to perform graphic operations.
- Make sure that you have created source tables, objects, links, and events.
- Make sure that you have obtained data in the tables that have been mapped to the primary keys of the objects to be analyzed. You can obtain the data by querying the corresponding tables in the database.


#### Procedure

1. [Log on to Analytics Workbench](#).
2. Click **Create Analysis**. A **Temporary Analysis** tab page appears.

3. Click Add in the toolbar and then click the blank space, or right-click the blank space and select Add Node. Set the parameters in the Add Node dialog box that appears.

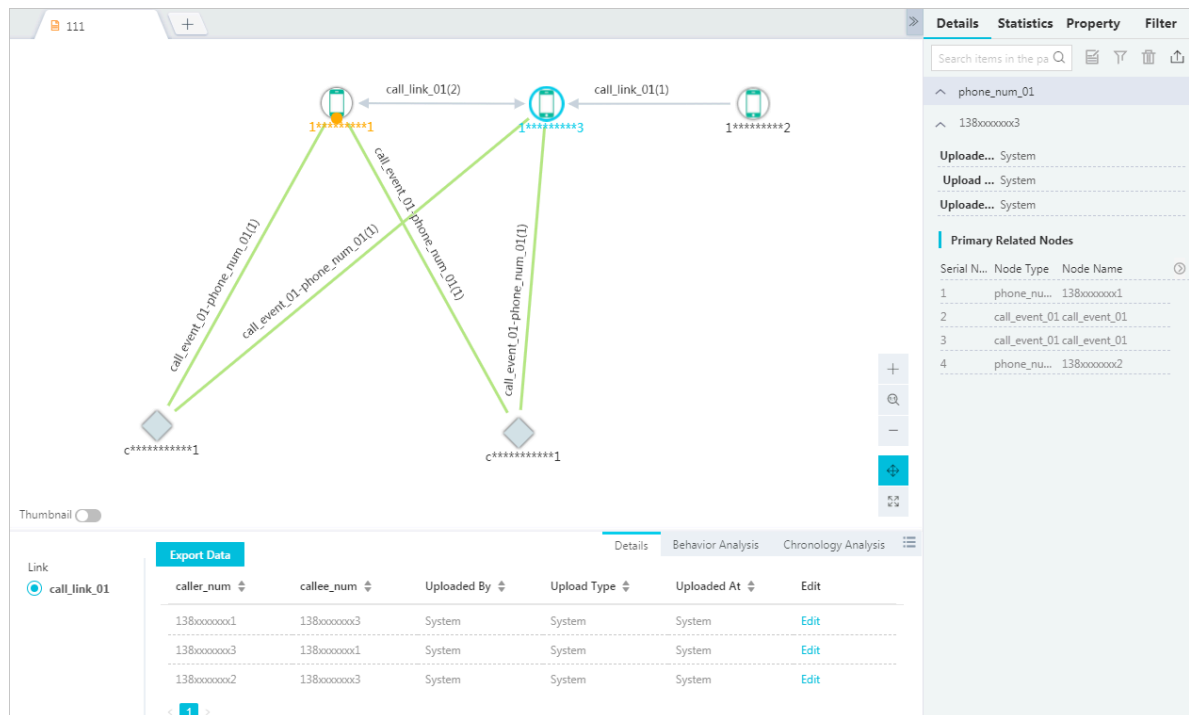
The parameters are described in [Table 7-35: Parameters descriptions for adding a node](#).

Table 7-35: Parameters descriptions for adding a node

Parameter	Description
Object type	<p>The drop-down list displays all created objects. Select an object as needed.</p> <div> <b>Note:</b> Graph Analytics supports adding compound nodes. A compound node is defined by multiple sub-types. For example, you can specify two sub-types for the object type "person": ID card and passport. The person can be uniquely identified by either the ID card or the passport.</div>
Text area	<p>Enter one or more primary key values.</p> <p>Separate multiple primary key values with commas (,).</p>

4. Click OK.

5. Right-click a node that has been added, and select Quick Extension. The system automatically performs a link analysis based on the configured data sources, objects, links, and events, and displays the analysis results in a graph.



6. Select one or more objects, links, or events. Click Behavior Chronology in the lower-right corner to see the corresponding Details, Behavior Analysis, and Chronology Analysis information.
  7. Select one or more objects, links, or events. Click the icon (⏪) in the upper-right corner of the right-side pane to see the corresponding information on the Details, Statistics, Property and Filter tabs.
  8. After the analysis has been completed, click Save in the upper-right corner. In the Save Analysis dialog box, enter a File Name and select a folder, and then click OK. A success message is displayed after the file has been saved.
- After you have saved the analysis file, if a collaborative analysis is required, you can share this personal analysis with other members.
9. Click the Share icon in the upper-right corner to specify the members you want to share this analysis with.



## 7.12.4 Add a node

Data analyses are based on nodes. Before you perform a data analysis, you need to add the object to be analyzed as a node.

### Prerequisites

- Make sure that you have obtained an account and a password with the permission to perform graphic operations.
- You have created a source data table, object definitions, link definitions, and event definitions. For more information about these operations, see [Data sources](#) and [Object information](#).
- Make sure that you have obtained data in the tables that have been mapped to the primary keys of the objects to be analyzed. You can obtain the data by querying the corresponding tables in the database.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis file.

3. Click **Add Node** and click anywhere in the blank space. Or right-click anywhere on the blank space, select **Add Node**, and then set the parameters in the **Add Node** dialog box.

The parameters are described in [Table 7-36: Parameter descriptions](#).

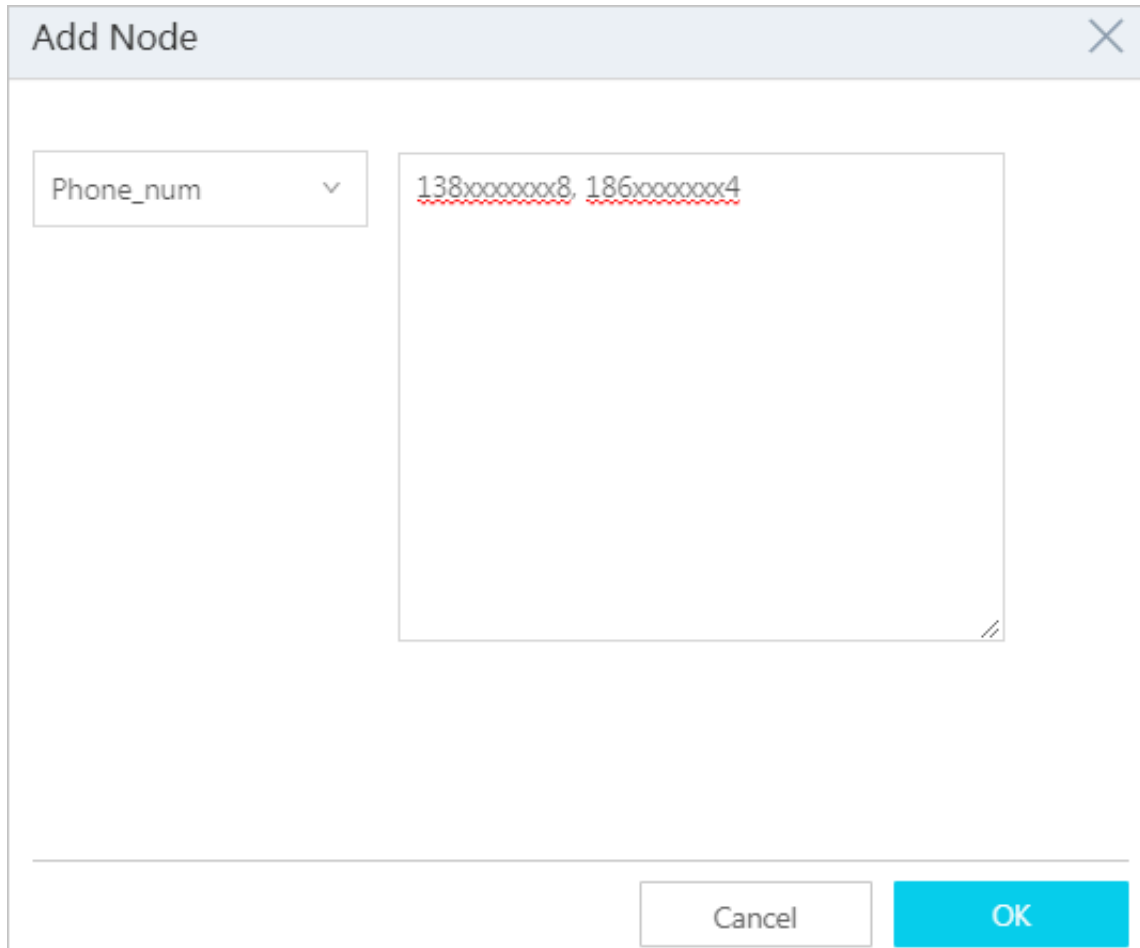



Table 7-36: Parameter descriptions

Parameter	Description
Object Type	<p>Displays all created objects. You can select an object as needed.</p> <div> <b>Note:</b> Graph Analytics supports adding compound nodes. For example, you can specify two sub-types for object type "person": ID card and passport. The person can be uniquely identified by either the ID card or the passport.</div>

Parameter	Description
Text Area	<p>Enter one or more primary key values in the data table corresponding to the Object Information.</p> <p>Multiple primary key values must be separated by commas (,).</p>

4. After you have configured the parameters, click OK.

If you want to add only one node, you can click **Add Node** in the toolbar and click the graph area to place the node. Or, you can right-click anywhere in the blank space and select **Add Node** in the short-cut menu to place the node. When you add multiple nodes, the nodes are placed in a horizontal line layout or a matrix layout by default. You can configure the layout in **Advanced > System Settings > Functional Components > Relationship Network > add multiple node layouts**.

5. Click **Save** in the upper-right corner. In the **Save Analysis** dialog box that appears, enter the **File Name** and select a file directory. Click **OK**. A message is displayed, indicating that you have saved the file.

### 7.12.5 Delete nodes, links, and events

You can delete the selected content in the current graph area. You can quickly delete the nodes, links, and events.

#### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.

#### Context

When you delete object nodes, links and events related to these object nodes are also deleted.

#### Procedure

1. [Log on to Analytics Workbench](#), and open an existing analysis file, or create a new analysis.

2. You can delete object nodes, links, or events by using the following methods.

Method	Operation
In the toolbar	Select one or more nodes, links, or events in the canvas. Click the Delete icon in the toolbar. A message is displayed, indicating that the item has been deleted.
In the right-click menu	Select one or more object nodes, links, or events in the canvas, right-click a selected item, and select Delete Selected. A message is displayed, indicating that the item has been deleted.

### 7.12.6 Link extension

You can use link extension to search for all objects that are related to a specific object. You can discover associated clues and intelligence from large amounts of unrelated information, and turn the information into useful intelligence.

#### Prerequisites

- An analysis file exists. Add a new analysis as shown in [Create analyses](#).
- A node object exists. Add a new node as shown in [Add a node](#).

#### Context

Link extension supports the following two modes.

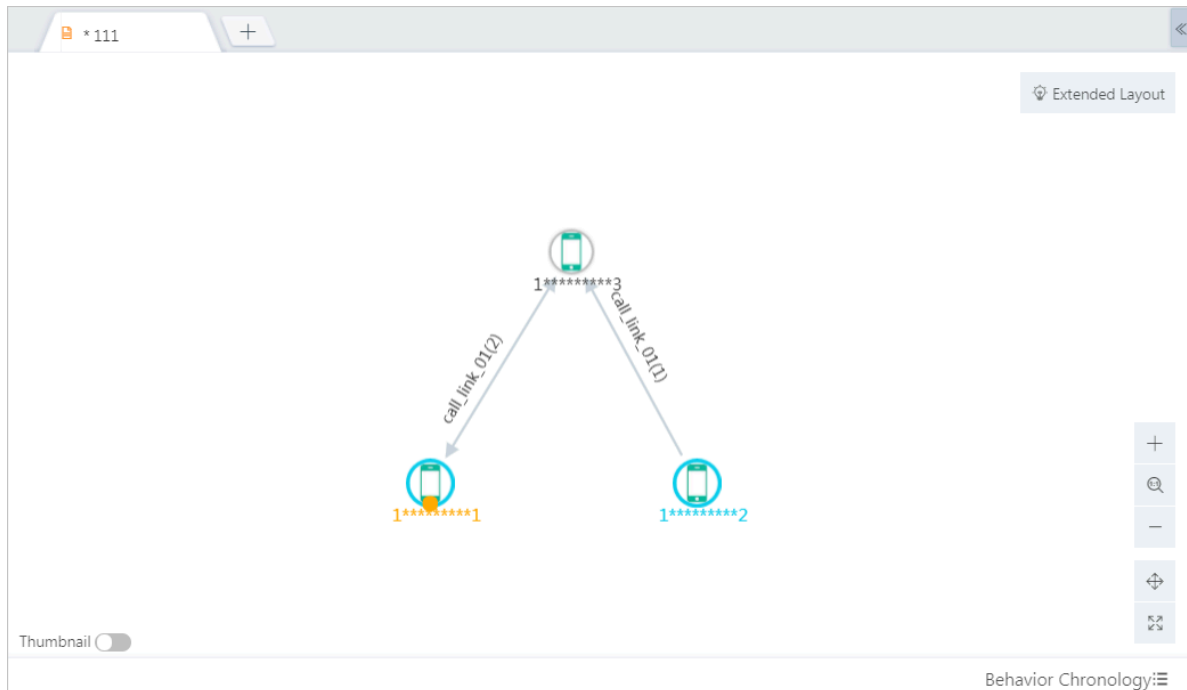
Link extension mode	Description	Reference procedure
Simple mode	Double-click a node to perform a first-degree link analysis on the node. Only the associated information for the first-degree link can be extended.	<a href="#">Step 3</a>
Advanced mode	Filter specific conditions based on business experience to extend the associated information.	<a href="#">Step 4 to Step 7</a>

#### Procedure

1. [Log on to Analytics Workbench](#).
2. Click an existing analysis file to open the file on the Graph page, or click Create Analysis to create an analysis file and add nodes.

Perform a simple analysis on a specific node.

### 3. Double-click a node to perform a first-degree link analysis.



Perform an advanced analysis on one or more nodes.

4. Select one or more nodes and click the Link Extension icon in the toolbar. You can select a maximum of 1,000 nodes. In the Link Extension dialog box, select the source objects.
5. Click Next, select links and the events, and specify the parameters for the links and events.

Link Extension

Original Objects

Link Type

Target Object

☐ All Links
 ☒ call\_links
 ☒ call\_link\_01
 ☒ Call\_second\_link...
 ☒ Call\_second\_link...
 ☐ Call\_multi\_links
 ☐ call\_multi\_link
 ☐ Custom Link
 ☐ Custom

Link Description : call\_link\_01

\* Direction

☐ Outbound

☐ Inbound

☒ Bidirectional

Basic Properties

caller\_num

callee\_num

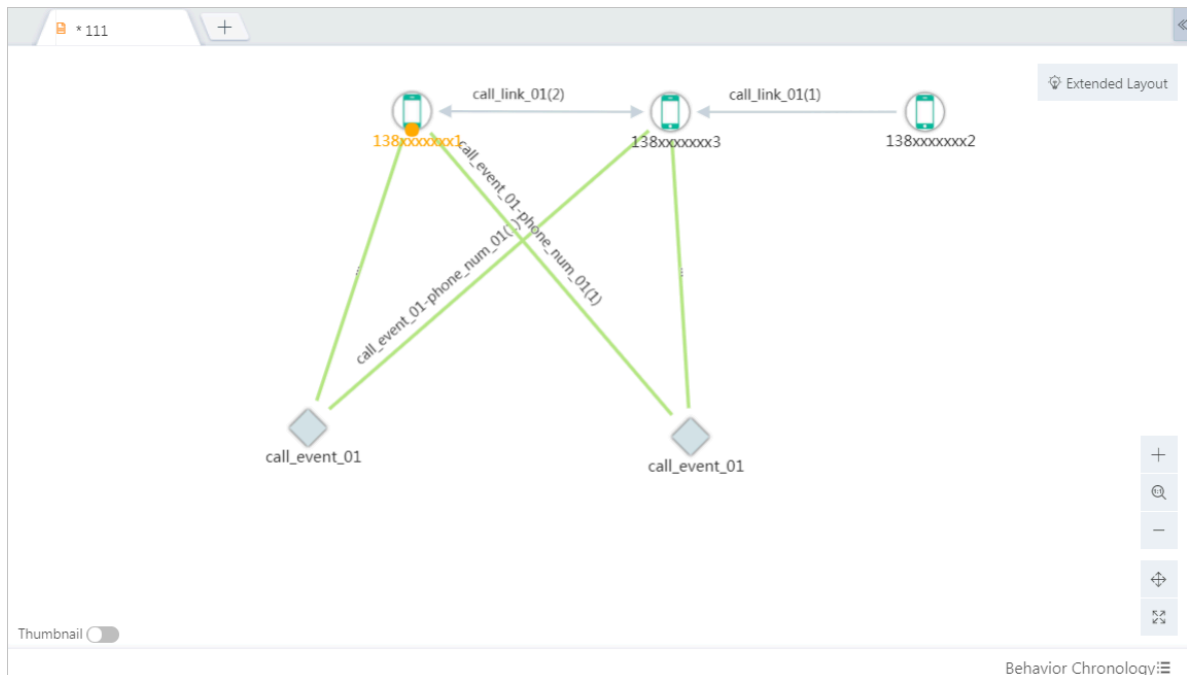
Cancel

Back

Next

Analyze

6. Click Next, and select the target objects.
7. Click Analyze to extend the link of the selected nodes.



## 7.12.7 Graphic operations

### 7.12.7.1 Move canvases

You can move the entire graph area by moving the canvas to perform comparative analysis.

#### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.


#### Procedure

1. [Log on to Analytics Workbench](#), and open an existing analysis file or create a new analysis.
2. You can move the canvas by using the following methods.



**Note:**

After you move the canvas, it will take a short time to reproduce the canvas. There will be a short delay. During that time, do not operate the canvas so as to prevent unexpected results.

Method	Operation
In the tool bar	Click the Enter Move Canvas Mode icon  in the lower-right corner of the graph area and drag the canvas with your mouse.
Space key + mouse	Long press the Space key and drag the canvas with your mouse.
Use the scroll wheel	Move the canvas upwards or downwards with the scroll wheel of your mouse.

### 7.12.7.2 Zoom in and zoom out canvases

Graph Analytics enables you to zoom in and zoom out the content in the graph area to perform a comparative analysis from a full-graph or partial-graph perspective.

#### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.




#### Procedure

1. [Log on to Analytics Workbench](#), and open an existing analysis file or create an analysis.
2. You can zoom in or zoom out the canvas in the following methods.



#### Note:

After you zoom in or zoom out the canvas, it will take a short time to reproduce the canvas. There will be a short delay. During that time, do not operate the canvas to prevent unexpected results.

Method	Operation
Shift key + mouse pointer	Long press the Shift key and zoom in or zoom out the canvas with the scroll wheel of your mouse.
Interface icons	<ul style="list-style-type: none"><li>• Click the Zoom In icon  to scale the size of the canvas.</li><li>• Click the Zoom Out icon  to scale the size of the canvas.</li><li>• Click the Restore Original Ratio icon  to restore the original ratio of the canvas.</li></ul>

### 7.12.7.3 Undo and redo operations

Graph Analytics supports undo and redo operations. In case of a mistake, you can quickly restore the graph area to the status prior to the operation.

#### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.

#### Background information

The undo and redo functions are described as follows:

- **Undo:** You can temporarily store the graph area status prior to the operation into the operation history. You can restore a historical version by undoing the operation. A maximum of 20 steps can be undone.
- **Redo:** After the Undo operation, you can also restore the analysis to a status prior to the undo operation.

The undo and redo operations can be performed to add nodes, add links, save virtual nodes, merge or split nodes. Undo and redo operations are applicable to layout features, link analysis features, and paste operations, and can be used to delete nodes or links.

#### Procedure

1. [Log on to Analytics Workbench](#), and open an existing analysis file or create a new analysis to perform multiple operations.
2. If you need to go back to a specific status, you can undo the operation by using the following methods.

Method	Description
Undo	<p>In the toolbar, click the Undo icon to undo the operation and restore the area to the status prior to the operation.</p> <p>You can perform the undo operation at a maximum of 20 steps.</p>



Method	Description
Redo	<p>In the tool bar, click the Redo icon to restore the graph area to the status prior to the undo operation.</p> <p>You can perform the redo operation at a maximum of 20 steps.</p>

#### 7.12.7.4 View thumbnails

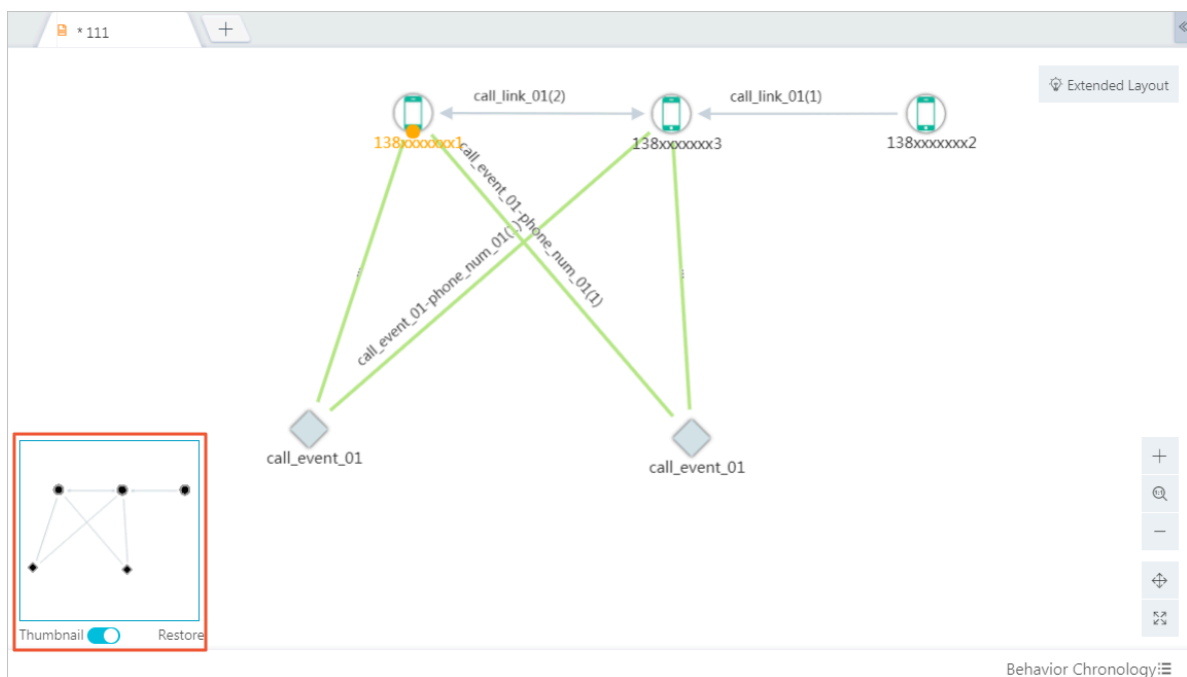
The thumbnail tab is located in the lower-left corner of the graph area. After you click the thumbnail tab, the thumbnail of the current full graph is displayed. The position of the visible area is framed out to help you view the analysis graph easily.

##### Prerequisites

Make sure that you have obtained an account and a password with the permission to perform graphic operations.

##### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis.
3. Click the Thumbnail icon in the lower-left corner to open the thumbnail of the current graph area.



4. You can drag the rectangular box in the middle of the thumbnail to drag the corresponding nodes in the canvas.

### 7.12.7.5 Right-click menu

The right-click menu in the graph area enables you to quickly operate on nodes, links, or events. In this topic, a vertex refers to a node, and an edge refers to a link or an event.

[Log on to Analytics Workbench](#), and open an existing analysis file or create an analysis.

The right-click menu varies depending on the position where you click on.

Right-click anywhere on the blank space

**Right-click anywhere on the blank space.** For more information, see [Table 7-37: Right-click menu description](#).

Table 7-37: Right-click menu description

Menu option	Description
Add Node	Adds nodes to the graph area. For more information, see <a href="#">Add a node</a> .
Select All Nodes	Selects all nodes in the graph area.
Select All Events	Selects all events in the graph area.
Select All Links	Selects all links in the graph area.
Paste	Pastes the content in the clipboard to the current analysis. This option will be grayed out if no objects or links exist in the clipboard.

Right-click a node or a link

**Right-click a vertex or an edge in the graph area.** For more information, see [Table 7-38: Right-click menu description](#).

Table 7-38: Right-click menu description

Menu option	Description
Add Link	<p>Adds a link for the selected two nodes to establish a relationship network. Graph Analytics supports directed and undirected links.</p> <ul style="list-style-type: none"><li>• <b>Directed link:</b> A clearly defined link with specific direction . The link line usually has arrows, as in the case of phone call relationship.</li><li>• <b>Undirected link:</b> A link with no clear direction. For example: people taking a train.</li></ul>
Add Label	<p>Adds label information to the selected nodes for easy analysis and identification. For more information about labels, see <a href="#">Add user labels</a>.</p>
Delete Selected	<p>Deletes the selected nodes or links from the graph area. For more information, see <a href="#">Delete nodes, links, and events</a>.</p>
Inverse Selection	<p>Selects all nodes other than the currently selected nodes.</p>
Select Correlated Nodes	<p>Selects all nodes that are correlated to the currently selected nodes.</p> <p>This option helps users locate the information for analysis and quickly select the correlated nodes. This option applies only to object nodes.</p>
Merge Selected	<p>Merges two or more selected nodes and displays the results.</p> <p>Users can sort, merge, visualize, and simplify the information in the entire graph area. This option applies only to object nodes, and the number of selected nodes must be greater than or equal to 2. Meanwhile, after the Merge Selected option is applied, the object nodes and links in the Selected state in the graph area remain unchanged.</p>

Menu option	Description
<b>Split Selected</b>	<p><b>Splits the merged nodes and displays the results.</b></p> <p>Users can extract information from a merged node for individual analysis. This option applies only to object nodes. Make sure that at least one node in the selected nodes is part of a merged node. Meanwhile, after the <b>Split Selected</b> option is applied, the object nodes and links in the <b>Selected</b> state in the graph area remain unchanged.</p> <p>The split operation filters the label information and property information of merged nodes by type. An UI example is shown in <a href="#">Figure 7-39: Split Filtered dialog box</a>, and the parameters are described in <a href="#">Table 7-39: Filter descriptions</a>.</p> <p>The split options are described in <a href="#">Table 7-40: Split options</a>.</p>
<b>Event Chain</b>	<b>Displays all events that occur on a specific object within a specified time range.</b>
<b>Clear Event Chain</b>	<b>Clears all event chains of an object within a specified time range.</b>
<b>Show Event</b>	<b>Shows an event. When an event involves multiple objects, you can use this operation to show the event and view all objects involved in the event.</b>
<b>Hide Event</b>	<b>Hides an event that has been shown.</b>
<b>Quick Extension</b>	<b>Performs a quick extension on the selected nodes, obtains the graphic results of the quick extension, and makes force-directed layouts.</b>
<b>Node Redirect</b>	<p><b>Redirects to a third-party business system by node type according to the URL configured in Administration Console.</b></p> <p>Node Redirect is a channel that connects Graph Analytics to other products. For example, if the object information can be viewed only by a third-party system, you can click Node Redirect to view the information in the third-party system. Configure the redirect URL in Administration Console depending on the specific situation.</p>

Menu option	Description
Copy	Copies the selected object nodes or link graphs and pastes them to the clipboard. You can copy object nodes and link graphs.
Roll Up	<p>If a multi-degree link is established between two objects, multiple intermediary objects are displayed between the two nodes. Select these nodes and click Roll Up to hide these intermediary nodes. After the intermediary objects are hidden, a dotted line is displayed between the two objects to represent the multi-degree link.</p> <p>To show these intermediary objects, press Ctrl and click on the dotted line.</p>
Save Virtual Node	Persists virtual nodes.
View Selected Node Details	You can view the details of a selected node in the right-side pane.

Figure 7-39: Split Filtered dialog box

Split Filtered

(Select an object for splitting and related property conditions.)

▼

☒ All

☒ phone\_num\_01

Label

No filter conditions.

Number of th...

2

-

4

identity\_card

Enter keywords, separated by space

name

Enter keywords, separated by space

phone\_num

Enter keywords, separated by space

Cancel

Split Selected and Delete Unselected

Only Delete Unselected

Only Split Selected

Table 7-39: Filter descriptions

Filter	Description
Labels	All labels are enumerated. You can delete some labels as needed.
Time type	Lists the maximum and minimum values of the time property. You can adjust the value range, for example, the departure time.
Value type	Lists the maximum and minimum numerical values. You can adjust the value range, for example, age.
Dictionary type	Like the Label filter, all dictionaries are enumerated. You can delete some dictionaries as needed.
String	Supports fuzzy searches. As shown in <a href="#">Figure 7-40: Fuzzy search</a> , search for the mobile phone number containing 189. Each item in the search results can be deleted by clicking the Delete icon next to the item. If no results have been found, the message "No results have been found." is displayed.

Figure 7-40: Fuzzy search

Split Filtered (Select an object for splitting and related property conditions.)

☒ All

☒ phone\_num\_01

Label No filter conditions.

Number of th... 2 - 4

identity\_card Enter keywords, separated by space

name Enter keywords, separated by space

phone\_num 138 Collapse All

138xxxxxxxx2 138xxxxxxxx3

Cancel

Split Selected and Delete Unselected

Only Delete Unselected

Only Split Selected

Table 7-40: Split options

Option	Description
Cancel	Cancels the split operation.
Split Selected and Delete Unselected	Splits the nodes that meet the conditions, and deletes the other nodes.
Only Delete Unselected	Retains nodes that meet the conditions in the merged node and deletes the other nodes.
Only Split Selected	Splits nodes that meet the conditions from the merged node, and retains the other nodes in the merged node.

## 7.12.8 Analyze

### 7.12.8.1 Group Analysis

You can use group analysis to perform multiple analyses on the relationships between any two objects in the group.

#### Prerequisites

- Make sure that you have obtained an account and a password with the permission to perform graphic operations.
- An analysis file exists. Add a new analysis as shown in [Create analyses](#).
- Two or more node objects already exist. Add a new object node as shown in [Add a node](#).

#### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select two or more object nodes in the graph area.
3. Choose **Analyze > Group Analysis**. In the Group Analysis dialog box that appears, select the start object to be analyzed.

4. Click Next, select links, and set conditions for each link based on your needs.

Group Analysis

Original Objects

Link Type

☐ All Links  
☒ call\_links  
☒ call\_link\_01  
☐ Call\_second\_link...  
☐ Call\_multi\_links  
☐ Call\_multi\_link  
☐ Custom Link  
☐ Custom

**Link Description :**

\* **Direction** ☐ Outbound ☐ Inbound ☒ Bidirectional

**Basic Properties**

**caller\_num**

**callee\_num**

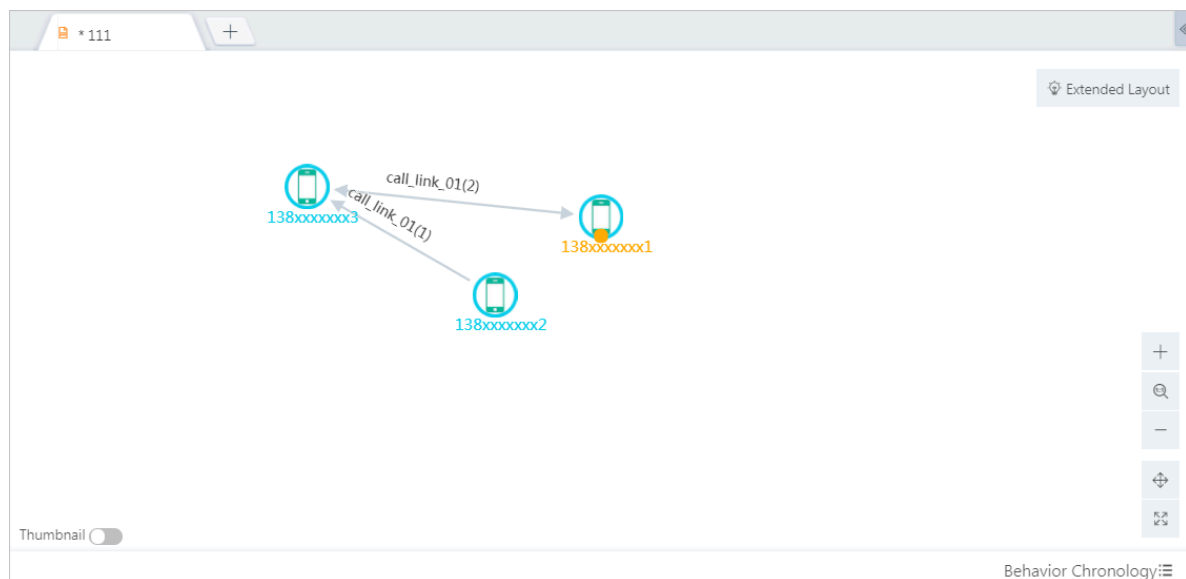
Cancel

Previous

Next

Analyze

5. Click Analyze to complete a group analysis for the selected node.



## 7.12.8.2 Common neighbor analysis

This feature allows you to analyze the commonly associated objects for a group of identical objects or different objects. The correlated degree is set to 2 by default.

### Prerequisites

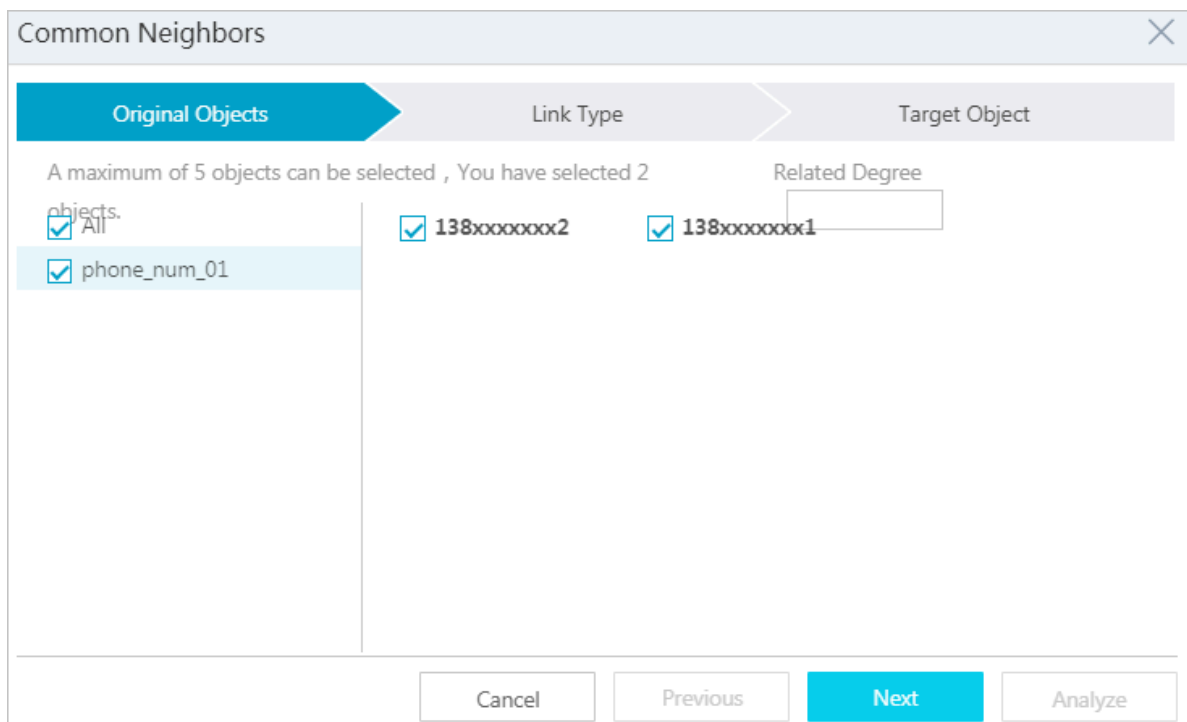
- You have obtained an account and a password with the permission to perform graphic operations.



- An analysis file exists. For more information about how to add an analysis, see [Create analyses](#).
- Two or more node objects already exist. For more information about how to add a node, see [Add a node](#).

## Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select two object nodes in the graph area.
3. In the toolbar, choose **Analyze > Common Neighbors**. In the Common Neighbors dialog box that appears, select the start object to be analyzed.



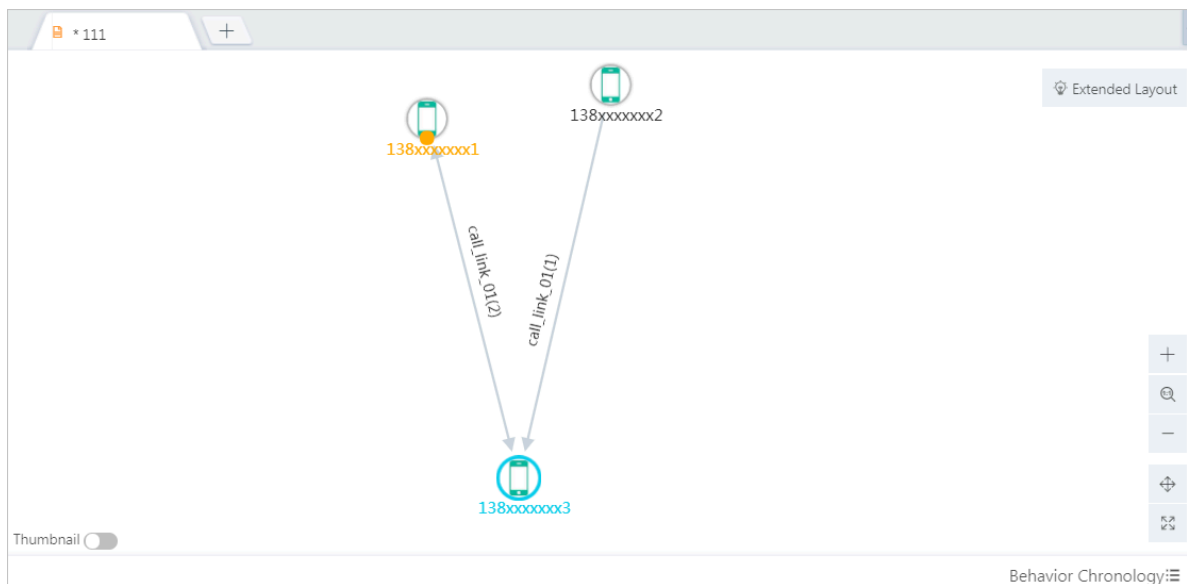
The screenshot shows the 'Common Neighbors' dialog box. It has a title bar with a close button (X). Below the title bar is a navigation bar with three tabs: 'Original Objects' (active, blue), 'Link Type', and 'Target Object'. Below the navigation bar, there is a text label: 'A maximum of 5 objects can be selected , You have selected 2'. Below this label is a list of objects: 'objects.' (with a checkbox), 'All' (with a checked checkbox), and 'phone\_num\_01' (with a checked checkbox). To the right of the list, there are two columns: 'Link Type' and 'Related Degree'. Under 'Link Type', there are two entries: '138xxxxxxx2' (with a checked checkbox) and '138xxxxxxx1' (with a checked checkbox). Under 'Related Degree', there is a text input field with the value '1'. At the bottom of the dialog box, there are four buttons: 'Cancel', 'Previous', 'Next' (highlighted in blue), and 'Analyze'.

4. Click **Next** to select links and set conditions for each link based on your needs.

The correlated degree is set to 2 by default.

You can set the link conditions by time, date, value, enumerated type, string equality, and fuzzy string matching.

**5. Click Analyze to complete a common neighbor analysis for the target node.**



### 7.12.8.3 Lineage analysis

Lineage analysis allows you to search the lineage between certain types of objects based on the lineage configurations in Administration Console.

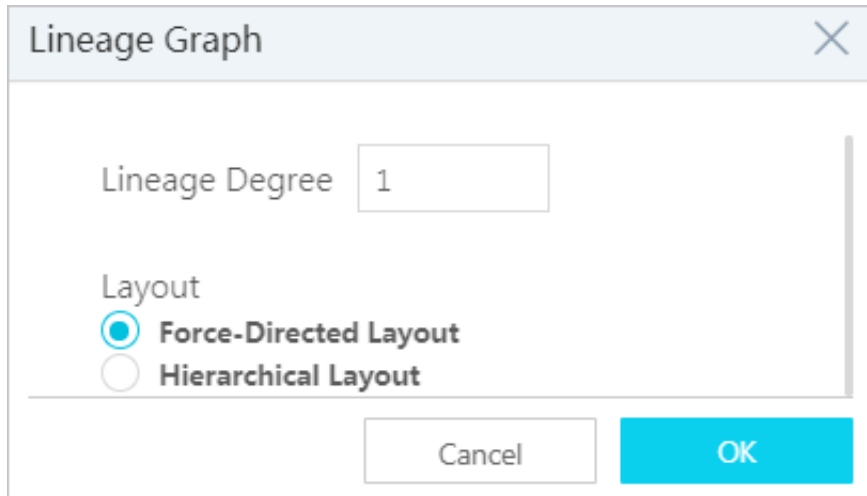
#### Prerequisites

- You have obtained an account and a password with the permission to perform graphic operations.
- An analysis file exists. For more information about how to add an analysis, see [Create analyses](#).
- A node object exists. For more information about how to add a node, see [Add a node](#).

#### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select one or more nodes in the graph area to perform a lineage analysis.

3. In the toolbar, choose **Analyze > Lineage Analysis**. In the dialog box that appears, specify Lineage Degree and Layout.

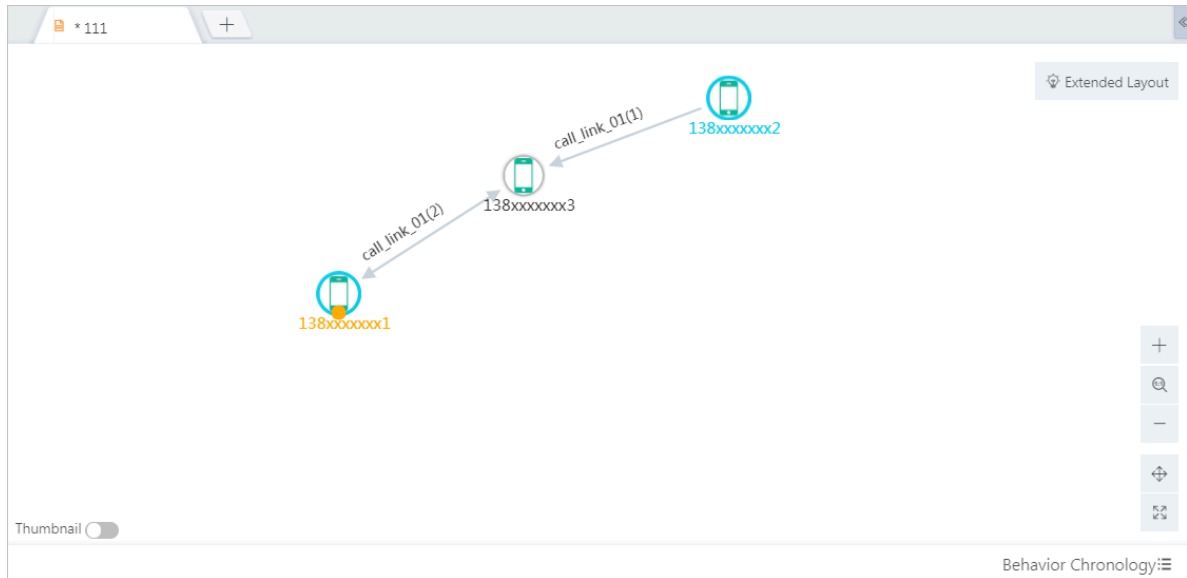


The image shows a dialog box titled "Lineage Graph" with a close button (X) in the top right corner. Inside the dialog, there is a section labeled "Lineage Degree" with a text input field containing the number "1". Below this, there is a section labeled "Layout" with two radio button options: "Force-Directed Layout" (which is selected, indicated by a blue dot) and "Hierarchical Layout" (which is unselected, indicated by a white dot). At the bottom of the dialog, there are two buttons: "Cancel" and "OK".

4. Click OK to complete the lineage analysis of the target node.

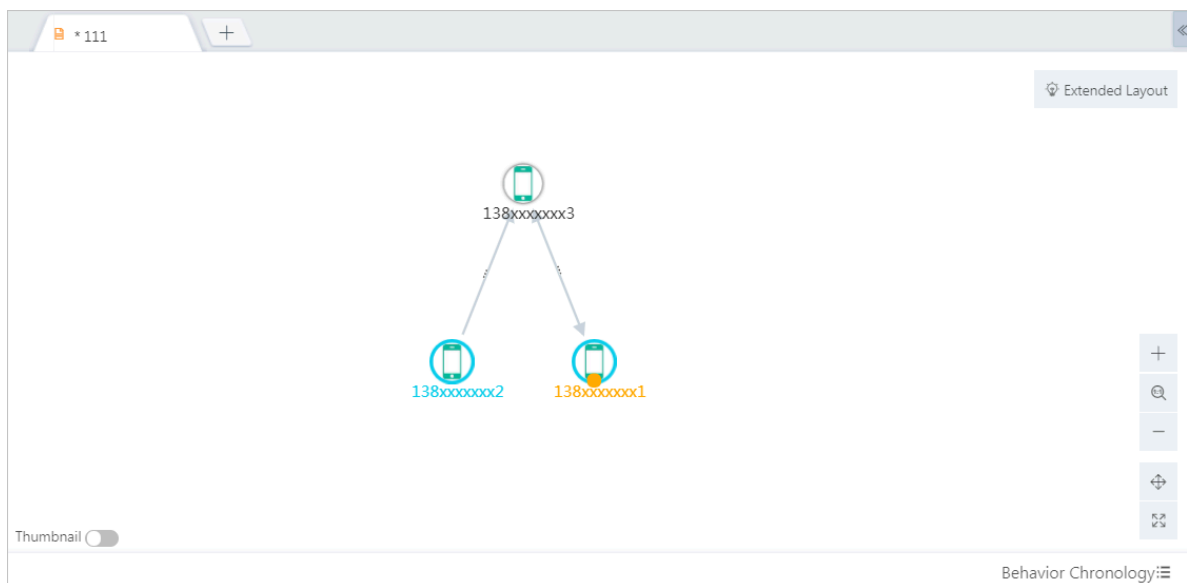
**The Lineage Degree is 1. The Layout is Force-Directed Layout. The result is shown in [Figure 7-41: Analysis results in the force-directed layout](#).**

Figure 7-41: Analysis results in the force-directed layout



**Assume that you have set the first-degree link and selected the hierarchical layout. The analysis results are shown in [Figure 7-42: Analysis results in the hierarchical layout](#).**

Figure 7-42: Analysis results in the hierarchical layout



## 7.12.8.4 Path analysis

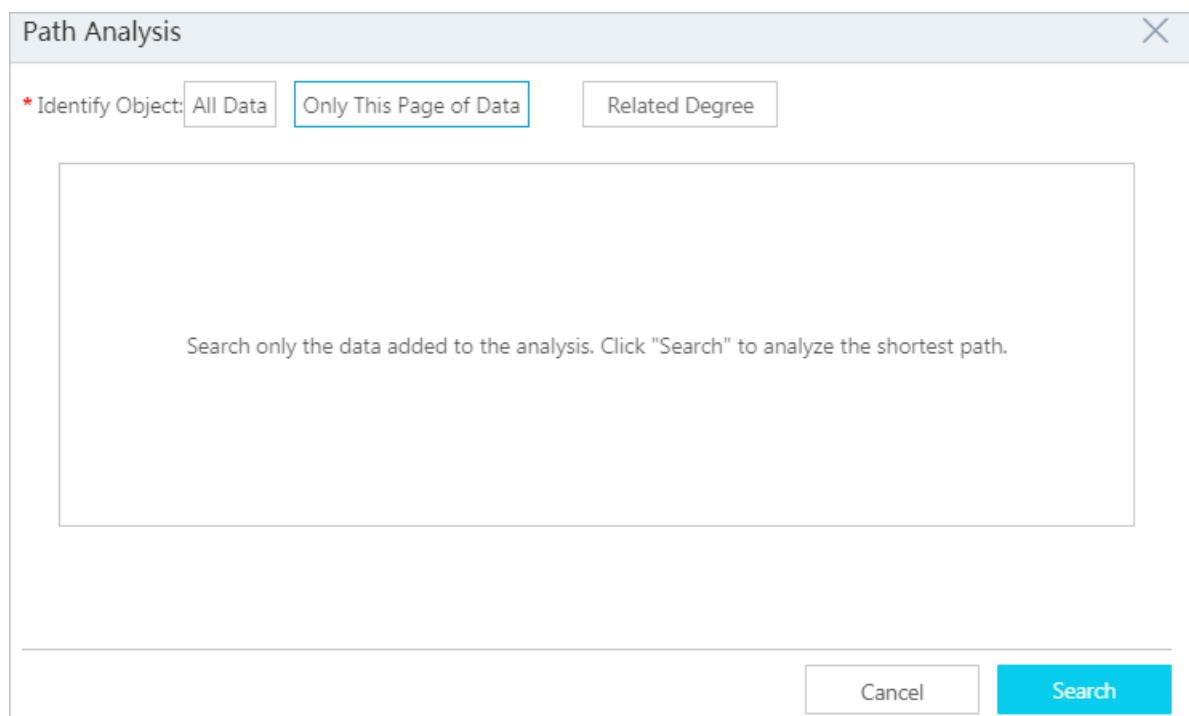
This feature allows you to analyze the link path between two objects.

### Prerequisites

- You have obtained an account and a password with the permission to perform graphic operations.
- An analysis file exists. For more information about how to add an analysis, see [Create analyses](#).
- Two or more node objects already exist. For more information about how to add a node, see [Add a node](#).

### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select two object nodes in the graph area.
3. In the toolbar, choose **Analyze > Path Analysis**. The Path Analysis dialog box appears.



The path analysis supports performing analyses on All Data and Only Data of This Page. For more information about the path analysis, see the following procedure.

4. Perform path analysis on all data: Select All Data to perform a path analysis and specify the relevant parameters.

**All Data:** Set the link condition, correlated degree, and whether to show the hotspot data, as shown in [Figure 7-43: Set conditions for the path analysis on all data](#).

If you select Show Hotspot Data, the hotspot data is displayed but not extended.



**Note:**

The system administrator monitors the hotspot data and adds a red tab to the hotspot node to inform the users of the hotspot data.

Figure 7-43: Set conditions for the path analysis on all data

Path Analysis

\* Identify Object:

All Data

Only This Page of Data

6

☐ Show Hotspot Data

☐ All Links
 

^

☒ TestLinkGroup
 

^

☒ LegalP>Enterpri...
 

^

☒ Investor>Enterp...
 

^

☐ call\_links
 

^

☐ call\_link\_01
 

^

☐ Custom Link
 

^

☐ Custom
 

^

Link Description :

'LegalP>Enterprise' There are no filter conditions under the specified link.

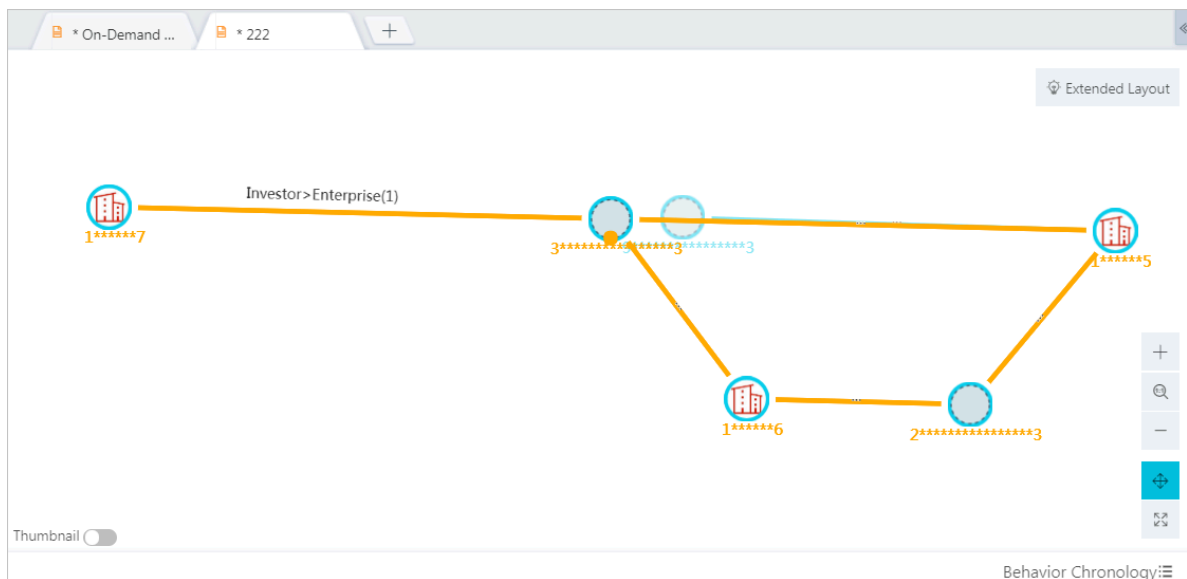
Cancel

Search

1696

Issue: 20200116

5. Click Search to perform a path analysis on all data.



6. Analyze only data of this page: Select Only Data of This Page and specify the correlated degree.

Analyzing only data of this page is to analyze the path between two nodes on the current page. This feature supports analyzing the shortest path and paths with a specified correlated degree.

**Correlated Degree:** If this parameter is not specified, the shortest path is analyzed by default. If you enter N, all paths with degrees less than or equal to N will be searched. N is specified by Administration Console and cannot be higher than 6.

## 7. Click Search to perform a path analysis on the data of this page.

Figure 7-44: Results of shortest path analysis - only data of this page

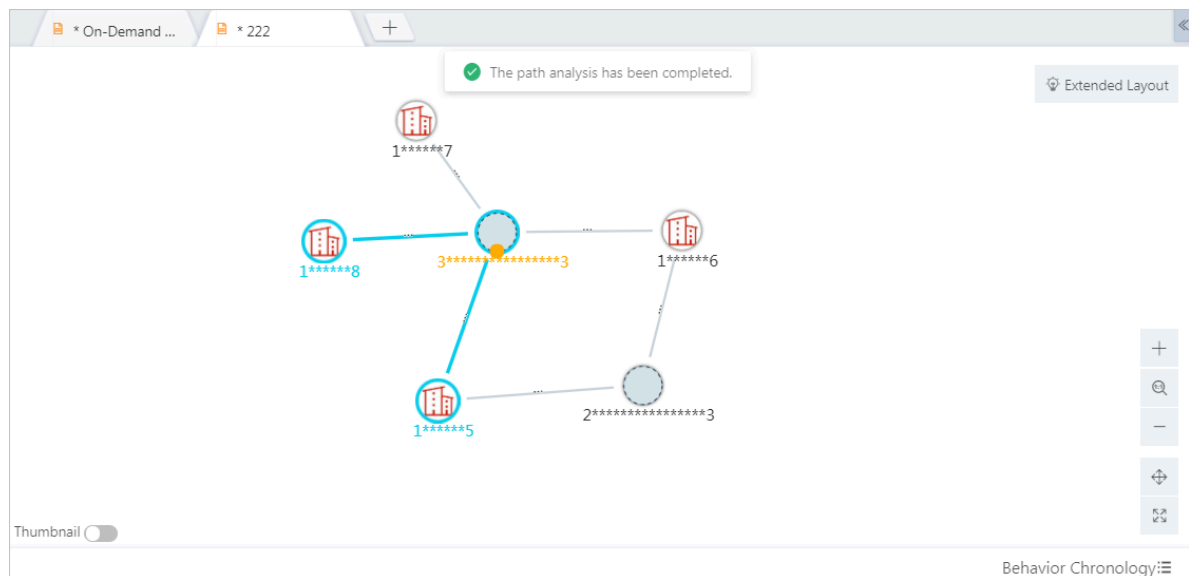
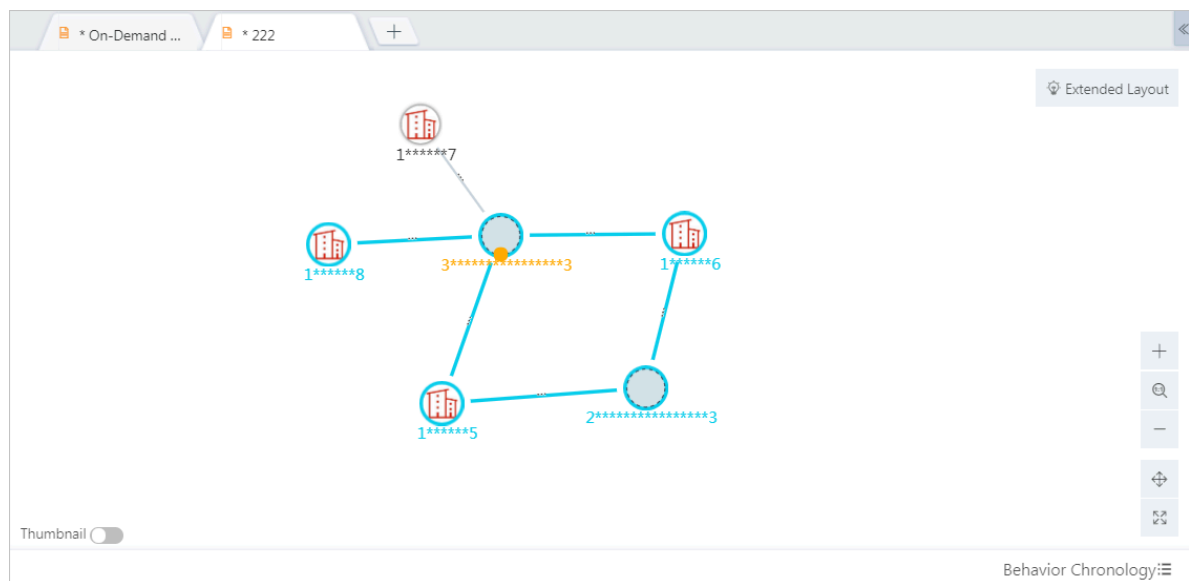


Figure 7-45: Results of the path analysis with a specified correlated degree - only data of this page



### 7.12.8.5 Backbone analysis

Based on the membership network on the current page, the backbone analysis uses smart service algorithms to help you explore the key nodes in a relationship network.

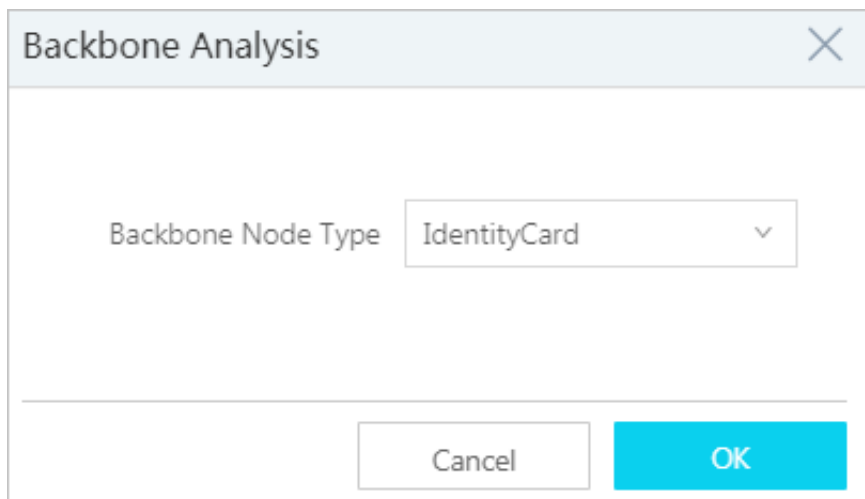
#### Prerequisites



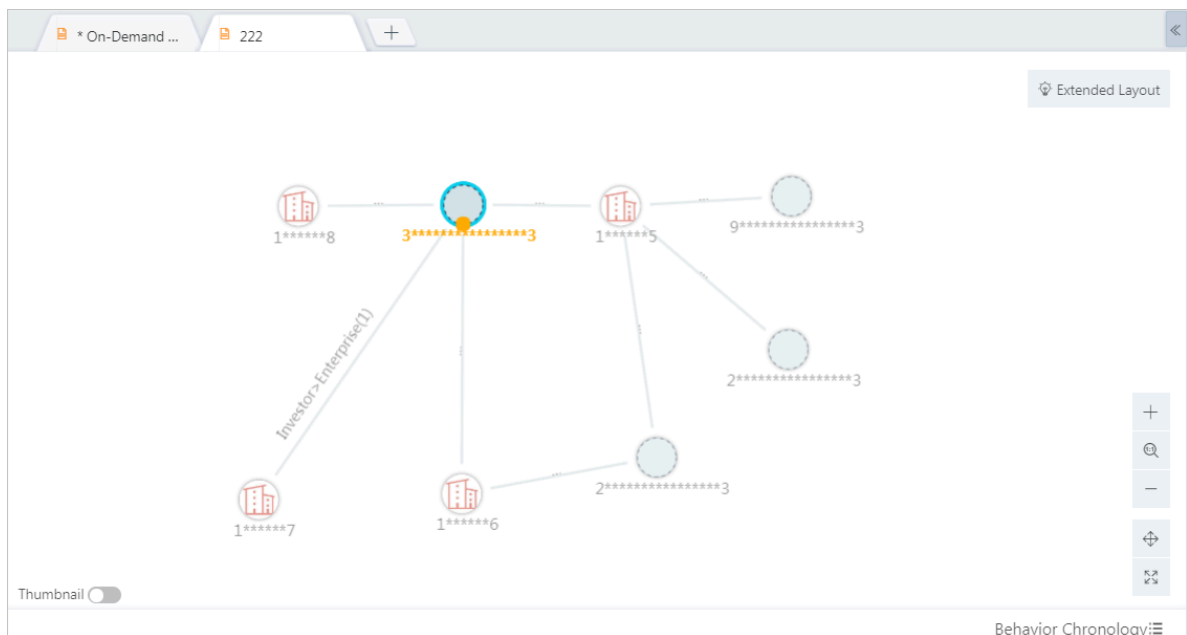
- You have obtained an account and a password with the permission to perform graphic operations.
- An analysis file exists. For more information about how to add an analysis, see [Create analyses](#).
- An object node already exists. For more information about how to add a node, see [Add a node](#).

## Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select an object node in the graph area.
3. In the toolbar, choose **Analyze > Backbone Analysis**, and set the Backbone Node Type in the dialog box that appears.



4. Click OK, and the key nodes in the graph area are highlighted.



### 7.12.8.6 Intimacy measurements

Perform an intimacy measurement to query the intimacy among objects of a specific type based on the intimacy settings configured in Administration Console.

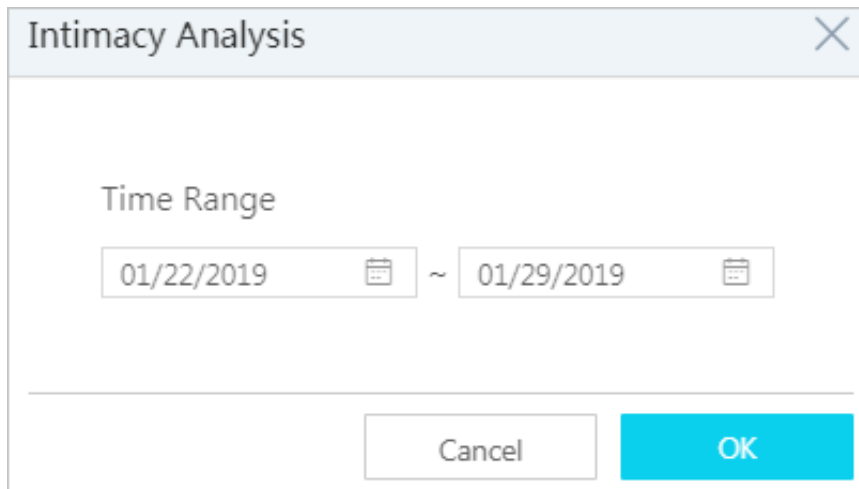
#### Prerequisites

- Make sure that you have obtained an account and a password with the permissions to perform graphic operations and intimacy measurements.
- Make sure that you have configured intimacy settings in Administration Console. For more information, see [Intimacy measurement settings](#).
- An analysis file already exists. For more information about how to create a new analysis, see [Create analyses](#).
- An object node already exists. For more information about how to add a new node, see [Add a node](#).

#### Procedure

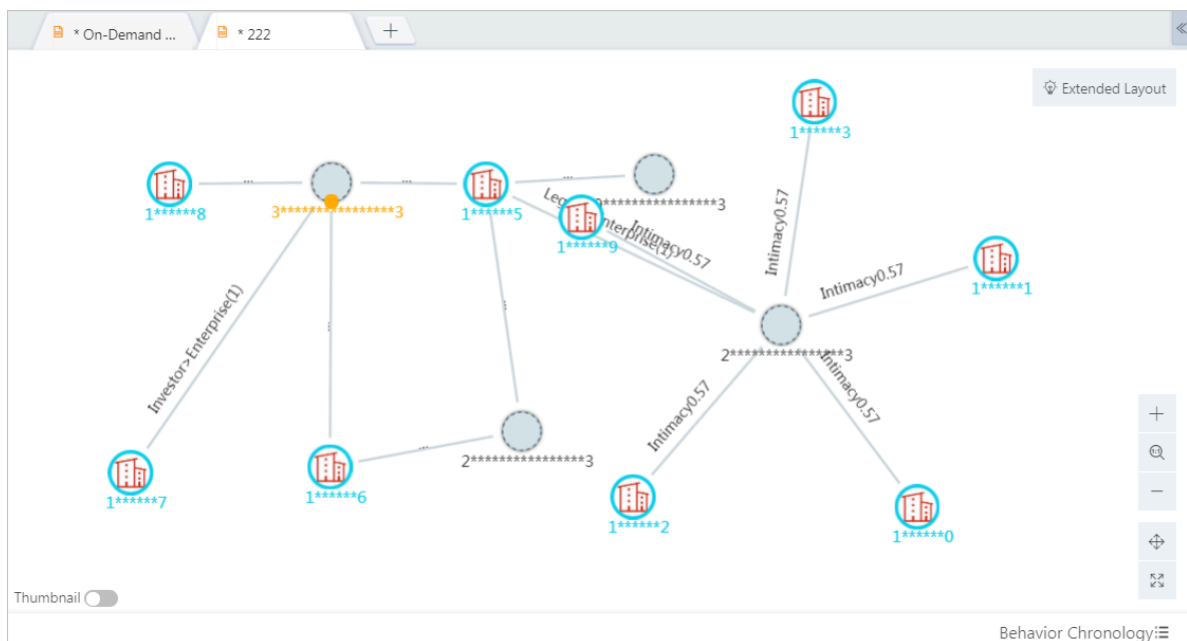
1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create an analysis, and select one or more nodes of the same type that have been created in the graph area.

3. Choose **Analyze > Intimacy Measurement** from the toolbar. In the **Intimacy Measurement** dialog box, set the time range.



The image shows a dialog box titled "Intimacy Analysis" with a close button (X) in the top right corner. Inside the dialog, there is a section labeled "Time Range" with two date input fields separated by a tilde (~). The first field contains "01/22/2019" and the second field contains "01/29/2019". Below the date fields, there are two buttons: "Cancel" and "OK".

4. Click **OK** to perform an intimacy measurement on the selected nodes.



## 7.12.9 Lock or unlock nodes

The node locking function keeps the node in a fixed position in the canvas to facilitate your operations.

### Prerequisites


Make sure that you have obtained an account and a password with the permission to perform graphic operations.

### Context

Except for nodes in the Force-Directed Layout, nodes in other layouts cannot be dragged. However, when the canvas moves as a whole, the locked nodes will move with the canvas.

### Procedure

1. [Log on to Analytics Workbench](#), and open an existing analysis file or create an analysis.
2. You can lock or unlock nodes by using the following methods.

Method	Operation
Lock nodes	<p>In the canvas, select one or more nodes, including merged nodes, and click the Lock icon  in the toolbar to lock the selected nodes or merged nodes.</p> <p>When a node is locked, you can see a gray lock at the top left of the node, as shown in <a href="#">Figure 7-46: Lock nodes</a>.</p>



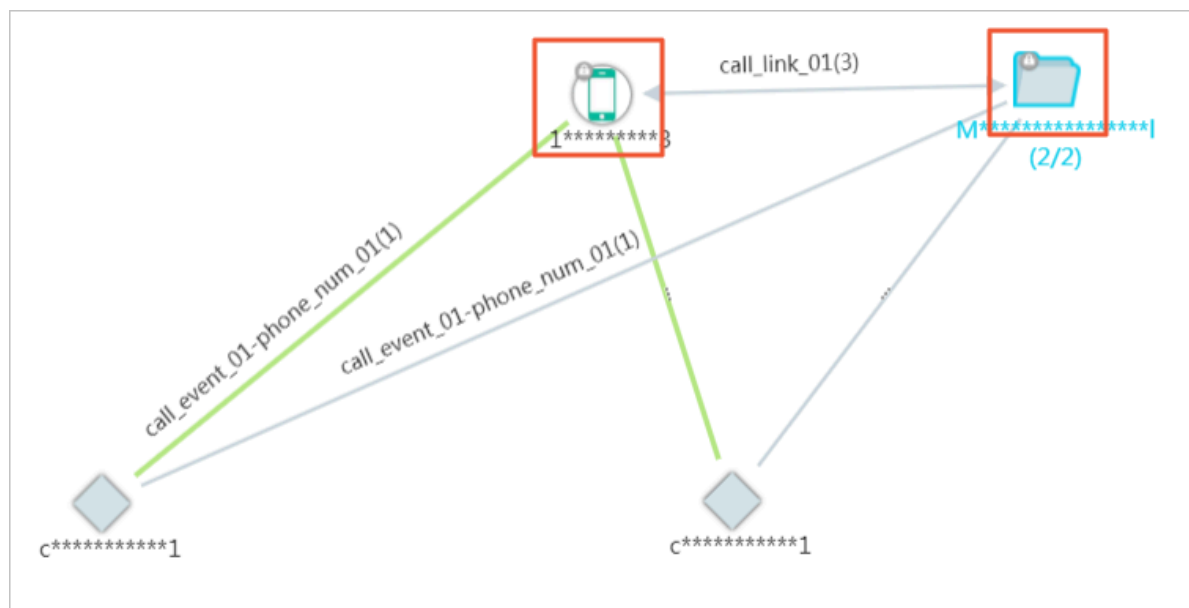
Method	Operation
Unlock nodes	<p>In the canvas, select one or more nodes, including merged nodes, and click the Unlock icon  in the toolbar to unlock the selected nodes or merged nodes.</p> <p>When a node is unlocked, the gray lock at the top left of the node disappears.</p> <div>  <b>Note:</b>                      If you split a locked merged node, the split nodes are automatically unlocked.                 </div>

Figure 7-46: Lock nodes



### 7.12.10 Network analysis

Network Analysis can be used to analyze node relationships from multiple perspectives, such as location precedence, closeness, and activity frequency.

#### Prerequisites

You have obtained an account and a password with the permission to perform network analyses.

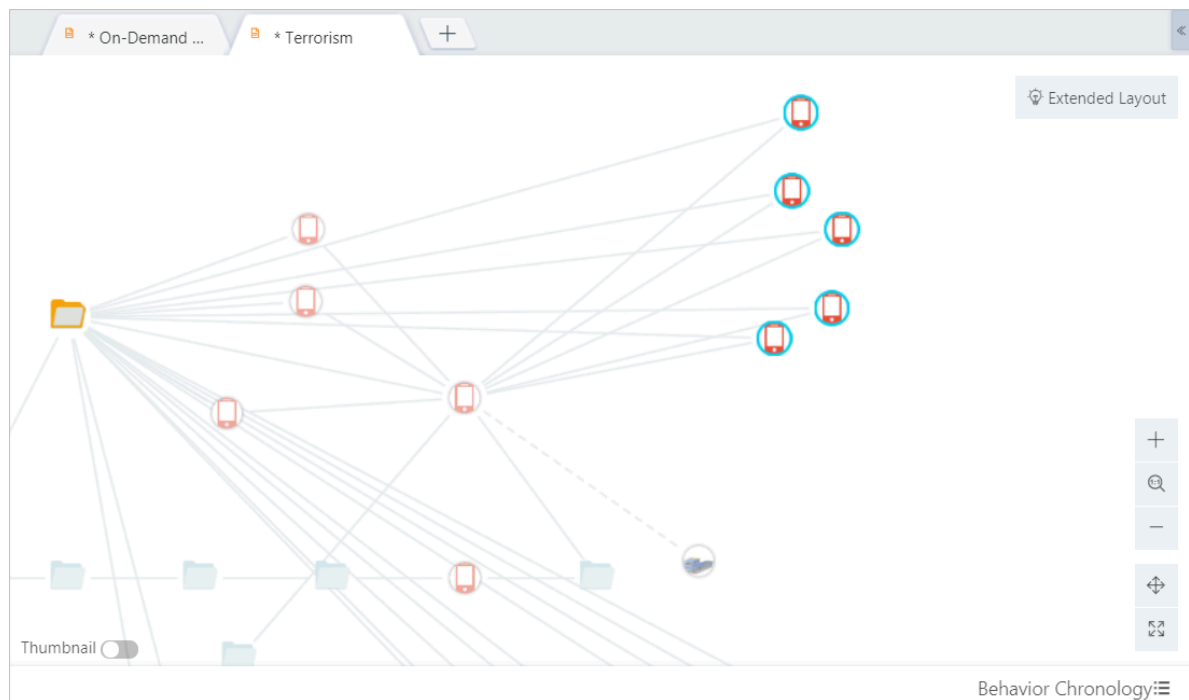
#### Context

The features of network analysis are described as follows:

- **Closeness:** Searches the key nodes that are most closely related to other parts of the network and checks the statuses of these nodes.
- **Location precedence:** Searches the bridging nodes that control the information flow.
- **Activity frequency:** Searches objects that are more active and more frequently interacted with other objects.
- **Data flow direction:** Searches the flow of relational data between objects in the network. It is only valid for directed relationships.

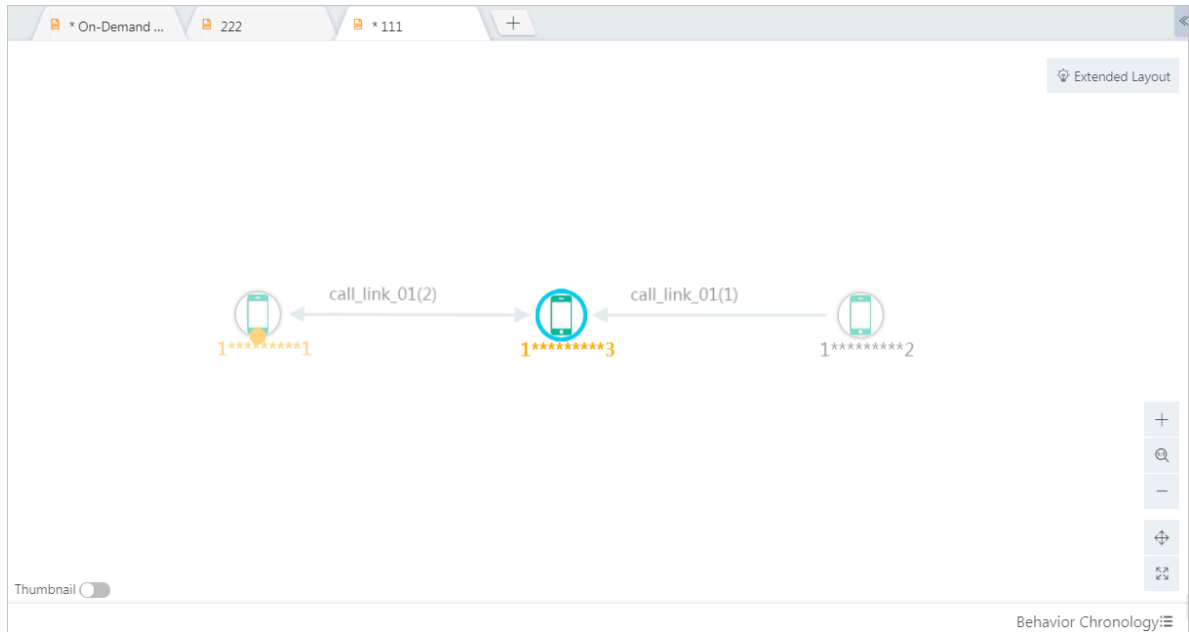
## Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis, and select one or more object nodes in the graph area.



- 3. Example of location precedence analysis: In the toolbar, choose Network Analysis > Location Precedence, as shown in [Figure 7-47: Location precedence analysis](#).**

Figure 7-47: Location precedence analysis



Other analysis examples are shown in [Figure 7-48: Closeness analysis](#), [Figure 7-49: Activity frequency analysis](#), and [Figure 7-50: Data flow direction analysis](#).

Figure 7-48: Closeness analysis

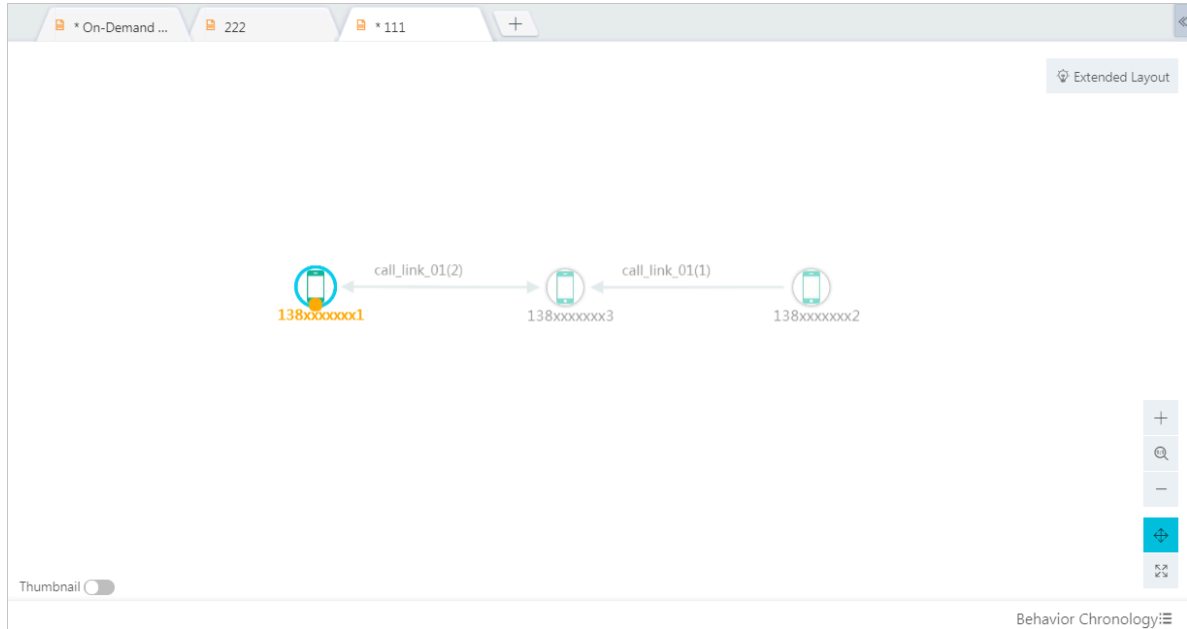


Figure 7-49: Activity frequency analysis

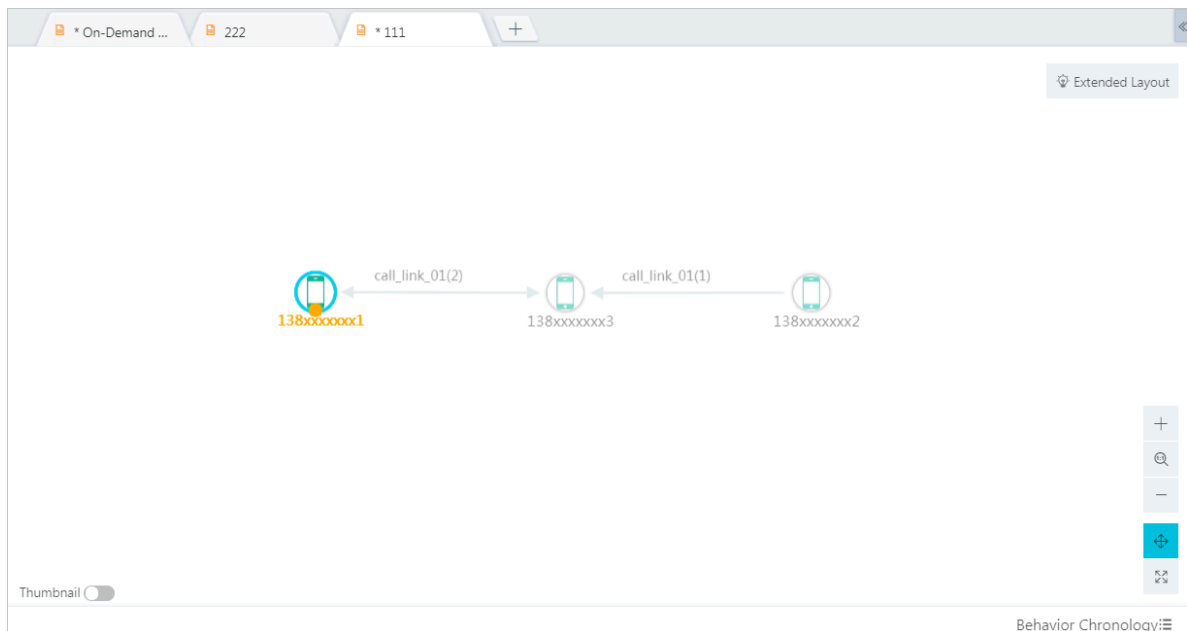
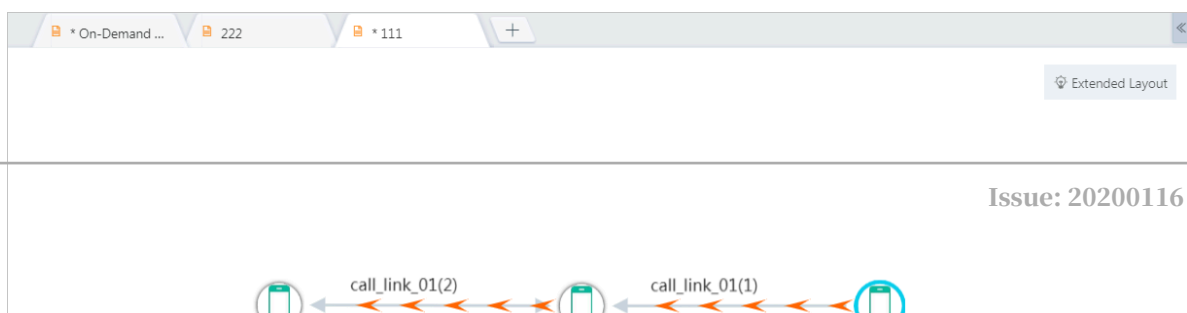


Figure 7-50: Data flow direction analysis





## 7.12.11 Closed-loop mining

Closed-loop mining allows you to check one or more nodes for continuous closed-loop links. This feature can be used to analyze cases including cash advance cases.

### Prerequisites

- You have obtained the account and password with the permission to perform graphic operations.
- An analysis file already exists. For more information about how to create a analysis, see [Create analyses](#).
- One or more node objects already exist. For more information about how to create a node, see [Add a node](#).

### Context

A closed loop refers to a link that starts from a node and ends at the same node. The following closed loops are available: the third-degree closed loops, fourth-degree closed loops, fifth-degree closed loops, and the sixth-degree closed loops.

The following example shows that a credit card cash advance will eventually return to the cashier regardless of the direction of the fund flow. For example, A is the cashier, and both B and C are coordinators. The fund flow is typically  $A > B > C > A$ , which forms a closed loop.

This topic describes loop steps by analyzing the third-degree closed loop of phone calls.

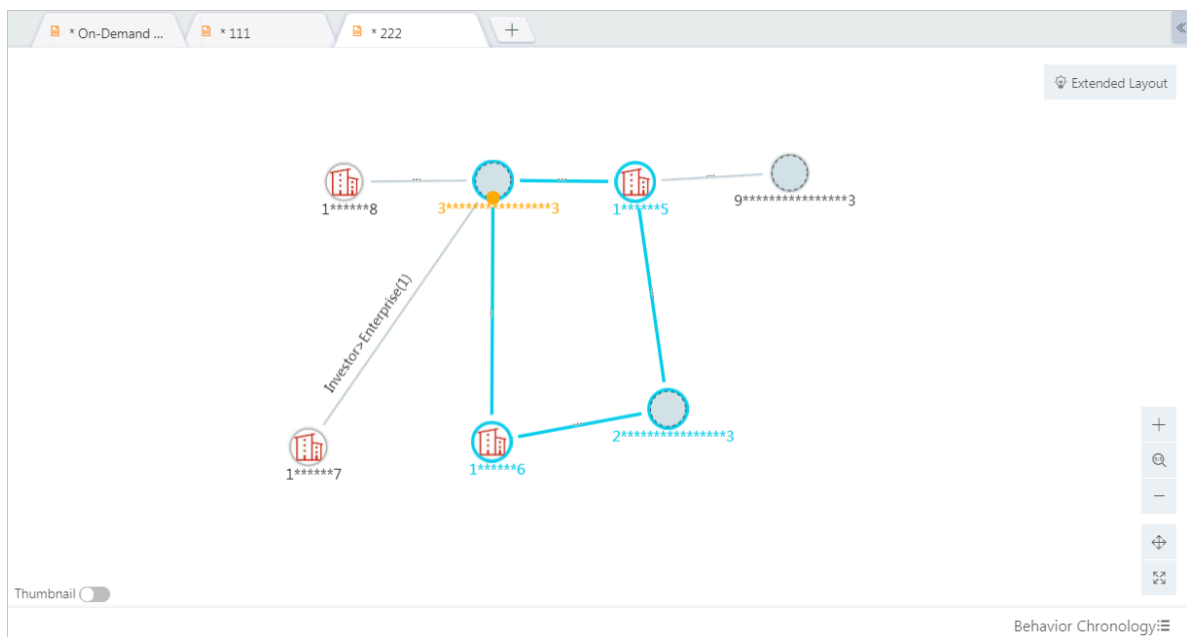
### Procedure

1. [Log on to Analytics Workbench](#).
2. Create an analysis and perform related analyses, or open an analysis file that already has analysis results.
3. Select a node in the analysis result, and choose Closed-Loop > 3-Degree Closed Loop in the toolbar.

4. In the dialog box that appears, specify **Directionality** and the link, and then click **OK**.

**Directionality** has the following options:

- **Undirected:** Analyzes both directional and undirectional closed-loop links on the specified node.
- **Directed:** Analyzes only the directional closed-loop links on the specified node.



## 7.12.12 Layouts

In Graph Analytics, you can easily analyze the content in multiple layouts.

Prerequisites

You have obtained an account and password with Layout permissions.

Background

Supported layouts are as follows.

Layout	Description
Matrix Layout	Objects are arranged in a matrix structure to help you sort and organize information during the analysis process.
Circle Layout	Objects are organized and evenly arranged in a circle to display their relationships. This layout helps you sort information and analyze the key nodes during the analysis process.

Layout	Description
Horizontal Layout	Objects are arranged along a horizontal line to help you analyze the information from a horizontal perspective.
Vertical Layout	Objects are arranged along a vertical line to help you analyze the information from a vertical perspective.
Force-Directed Layout	<p>The force-directed layout is used to visualize complex networks. All edges are more or less of equal length and there are as few intersecting edges as possible. With nodes and the weights of edges defined in advance, the force-directed layout positions the nodes automatically according to the principle that a higher weight leads to shorter distance. This procedure is convenient for you to tell how close nodes are from each other.</p> <p>This is a global layout where all object nodes and links in the graph area of Graph Analytics are calculated.</p>
Hierarchical Layout	Objects are arranged in a tree structure. The hierarchical layout is used for family trees and the organizational structures of enterprises.

#### Procedure

1. [Log on to Analytics Workbench.](#)
2. You can create a new analysis and produce the analysis results. Or, you can open an analysis file that already has the analysis results.
3. Take the matrix layout as an example. In the toolbar, choose Layout > Matrix Layout, as shown in [Figure 7-51: Matrix layout.](#)



#### Note:

We recommend that you choose a suitable layout for your business, so that you can easily view the analysis data.

The matrix layout can be used with the Merge Nodes feature, as shown in [Figure 7-52: Merge nodes.](#) This operation can sort and merge the analysis results when you

hand large amounts of information. The merged result is shown in [Figure 7-53](#):

*Merged result.*

Other layouts are shown in [Figure 7-54: Circle layout](#), [Figure 7-55: Horizontal layout](#), [Figure 7-56: Vertical layout](#), [Figure 7-57: Force-directed layout](#), and [Figure 7-58: Hierarchical layout](#).

Figure 7-51: Matrix layout

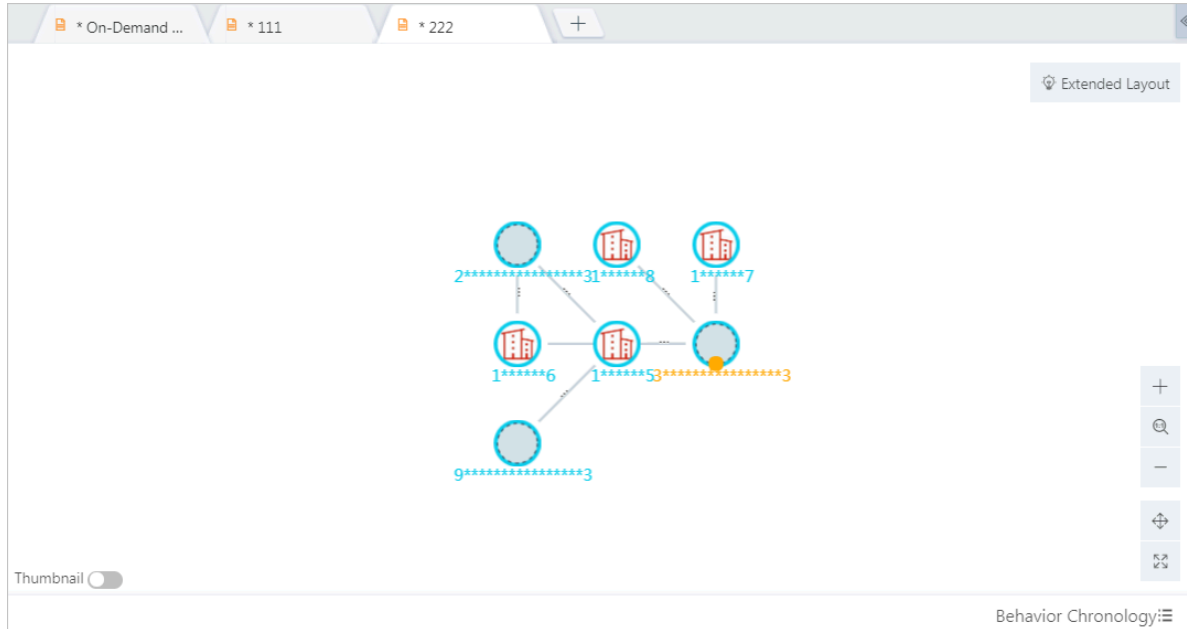


Figure 7-52: Merge nodes

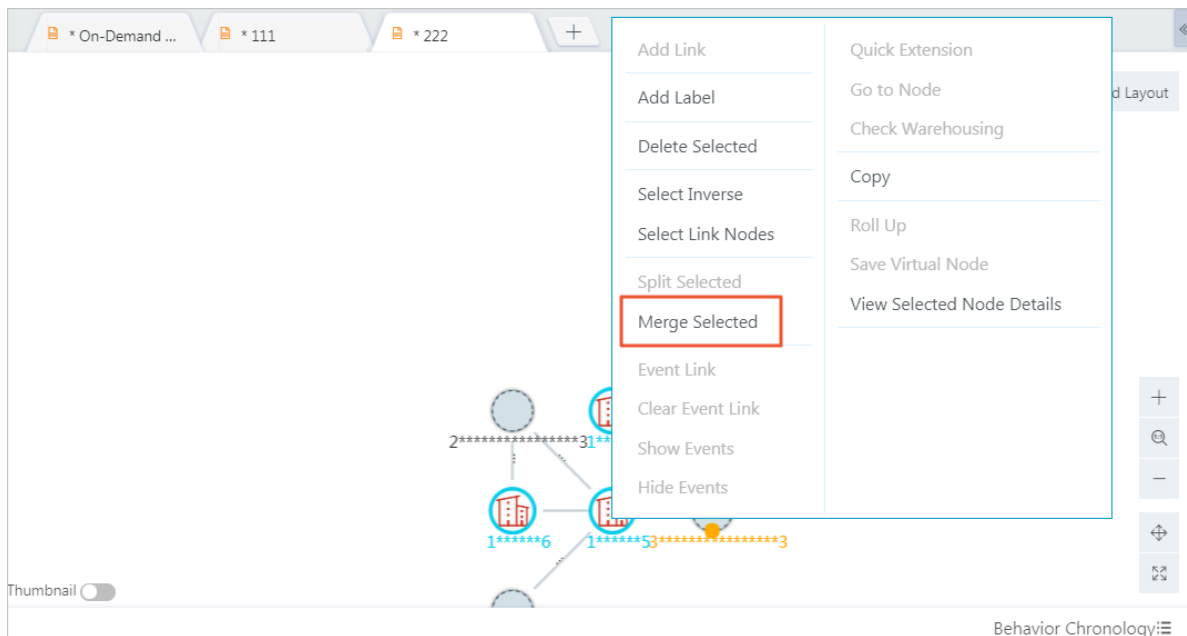


Figure 7-53: Merged result



## 7.12.13 Flag and unflag nodes

When you analyze a large number of object nodes, you can highlight the key object node by adding a small red flag to the node.

### Prerequisites

An object node already exists. For more information about how to add a node, see [Add a node](#).

### Context

You can perform flag operations on single nodes and merged nodes.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Click an existing analysis file to open the file on the Graph page, or click Create Analysis to create an analysis file and add nodes.
3. Perform the flag and unflag operations as follows.

Operation	Procedure
Flag	Select the target nodes or merged nodes, and then click Flag in the top navigation bar. A red flag is displayed on the selected node or merged node.
Unflag	Select the target nodes or merged nodes, and then click Unflag in the top navigation bar. The red flag displayed on the selected node or merged nodes disappears.

## 7.12.14 Labels

### 7.12.14.1 Label types

You can use labels to identify the content, category, and other properties of the node, which is easy for you and other users to search and locate nodes. In Graph Analytics, you can add label content to the node objects in the graph area. Labels are divided into system labels and user labels.

You can attach a system label or a user label to each node object.

- **System labels:** labels attached to the node objects by the system based on algorithms.

The system labels are displayed on the left side of the nodes. You cannot delete or modify the labels or click the Like button on the system label.

- **User labels:** labels attached to objects by the user.

The user labels are displayed on the right side of the nodes. The user labels with certain permissions can be deleted and liked by corresponding users.

## 7.12.14.2 User labels

Graph Analytics supports four types of user labels. The visible ranges and operations for labels vary by type.

### User label types

- **Public**

After the analysis is shared, all people can see the label and click the Like button on this label.

- **Only Me**

This label is visible only to the person who added the label. After the analysis is shared, other people cannot see this label.

- **Only for This Analysis**

This label is visible only in the current analysis, and the Like button of the label can only be clicked in the current analysis. After a node with this label has been copied to another analysis, this label becomes invisible.

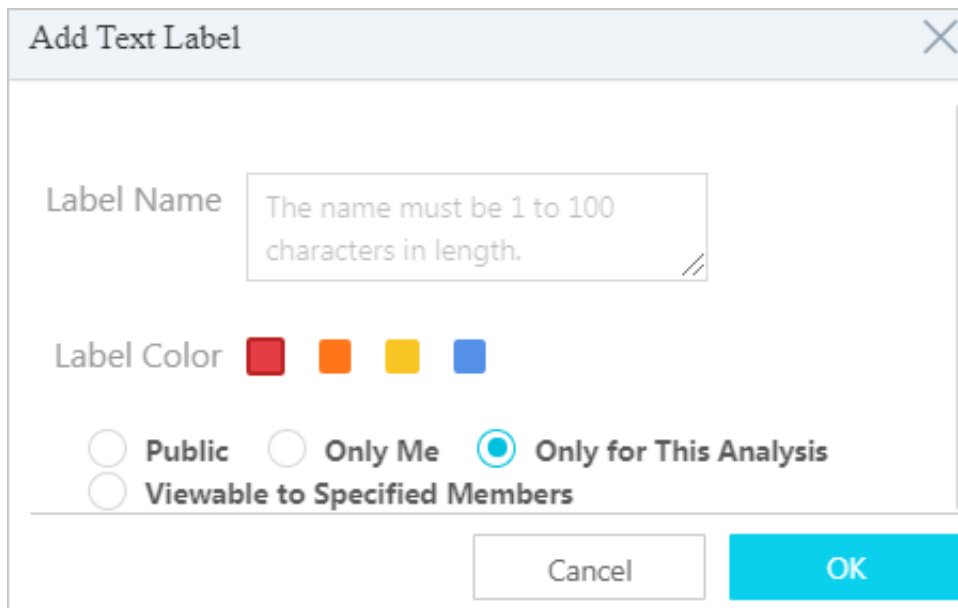
- **Viewable to Specified Members**

This label can only be seen and liked by the users that are specified when the analysis is shared.

## User label colors

**You can add user labels of different colors. By default, four colors are provided to differentiate the labels: red, yellow, blue, and green, as shown in [Figure 7-59: User label colors](#).**

Figure 7-59: User label colors

A screenshot of a dialog box titled "Add Text Label" with a close button (X) in the top right corner. The dialog contains a "Label Name" text input field with a placeholder message: "The name must be 1 to 100 characters in length." Below this is a "Label Color" section with four colored squares: red, orange, yellow, and blue. Underneath the color squares are four radio button options: "Public", "Only Me", "Only for This Analysis" (which is selected), and "Viewable to Specified Members". At the bottom right of the dialog are two buttons: "Cancel" and "OK".

### 7.12.14.3 Add user labels

**When some nodes are hard to understand, you can classify and describe these nodes by adding labels. This helps you and other users to understand these nodes quickly and easily.**

#### Prerequisites

**Make sure that you have obtained an account and a password with the permission to perform graphic operations.**

#### Context

**Based on the visible range, labels can be divided into four types: Public, Only Me, Only for This Analysis, and Viewable to Specified Members.**



#### Note:

**You cannot add labels to merged nodes.**

#### Procedure

1. [Log on to Analytics Workbench](#).



2. Open an existing analysis file or create an analysis.

3. You can use one of the following methods to add a label:

- Select one or more nodes in the graph area, right-click a selected node, and then select Add Label. Set the parameters in the dialog box that appears.
- Select one or more nodes on the Details tab of the right-side pane, click the Text Note icon in the top of the pane, and then set the parameters in the dialog box that appears.

The parameters are described in [Table 7-41: Parameters and descriptions](#).

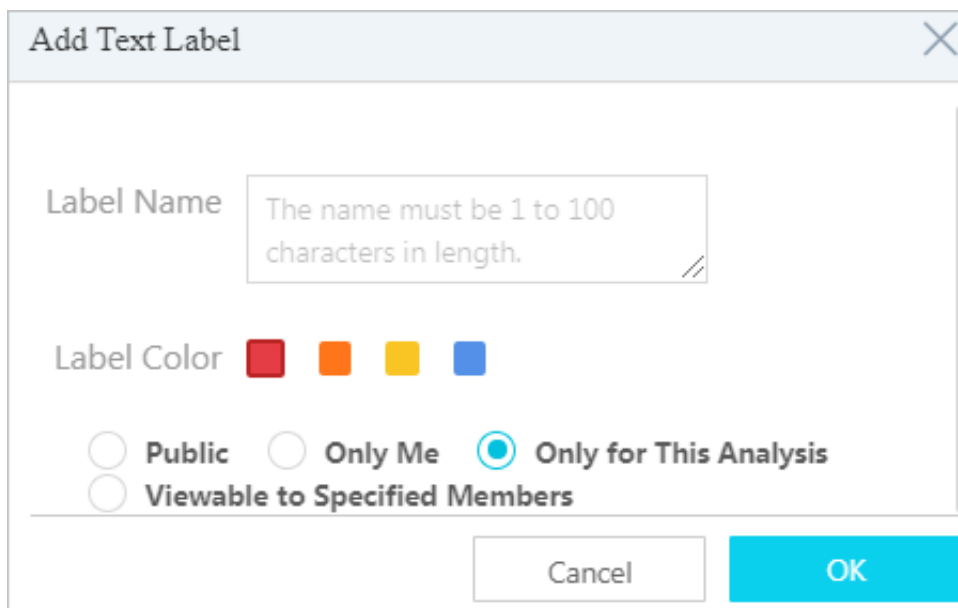


Table 7-41: Parameters and descriptions

Parameter	Description
Label Name	The note or description of the node, which helps you and other users understand the node. The label name must be from 1 to 20 characters in length.
Label Color	The color of the label displayed in the graph area for easy identification. Four colors are supported: red, orange, yellow, and blue.

Parameter	Description
Visible range options	<ul style="list-style-type: none"> <li>• <b>Public:</b> Users who use the node can see the label and click the Like button of the label.</li> <li>• <b>Only Me:</b> The label is visible only to the person who added the label. Other users cannot see the label when they add this node to the graph area.</li> <li>• <b>Only for This Analysis:</b> You can see the label and click the Like button of the label only in the current analysis. This label is invisible after this node is added to or appears in another analysis.</li> <li>• <b>Viewable to Specified Members:</b> Only the specified users can see the label and click the Like button of the label.</li> </ul> <p>When you select Viewable to Specified Members, you must specify users as needed.</p>

4. Click OK. A success message is displayed after the label has been added.

#### 7.12.14.4 View labels

In Graph Analytics, you can view labels by type and easily analyze the object nodes.

##### Prerequisites

There are labels visible to you in the current analysis file.

##### Procedure

1. [Log on to Analytics Workbench.](#)
2. Open an existing analysis file or create a new analysis. You can switch the display mode of the labels based on your needs.

Display mode	Operation
All Labels	In the toolbar, click All Labels. The system labels of each node and the labels visible to you in the current analysis are displayed in the graph area.
My Labels	In the toolbar, click the label icon and select My Labels. All labels created by you for the displayed node are shown in the graph area.
Hide Labels	In the toolbar, click the label icon and select Hide Labels. All labels in the graph area are hidden.

### 7.12.14.5 Click likes and delete likes

When you agree on a label that was added by another user to a node, you can click the Like button for the label.

#### Prerequisites

- Make sure that you have obtained an account and a password with the permission to perform label-related operations.
- Make sure that other users have added labels that are visible to you.

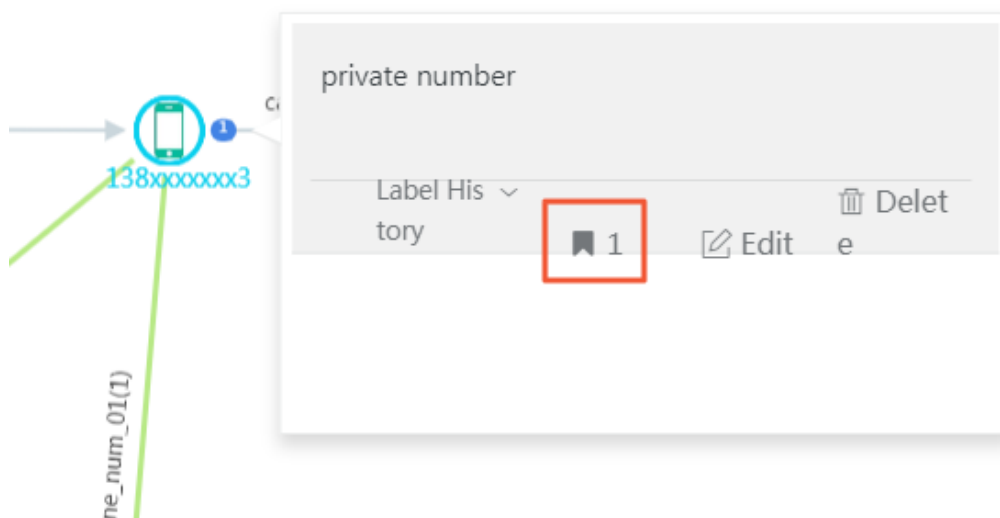
#### Context

Only labels that are Public, Only Me, and Viewable to Specified Members can be liked. By default, every label has one like. When the last like of a label is canceled, the label is removed from the node.

After you like a label, the number of likes increases by 1, and the Like button  1 changes to the Undo Like button  1. After you undo a like, the number of likes decreases by 1.


#### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create an analysis, and add or locate a node that has user labels that are visible to you.
3. Click the label number next to the node, and click a label name. The detailed information about the specified label is displayed.



4. If you agree on the label content, click the Like button  1. A success message is displayed after the operation is completed.

## What's next

If you think that a label you previously liked does not match the node, you can click the Undo Like button  to undo the like.

### 7.12.14.6 Edit user labels

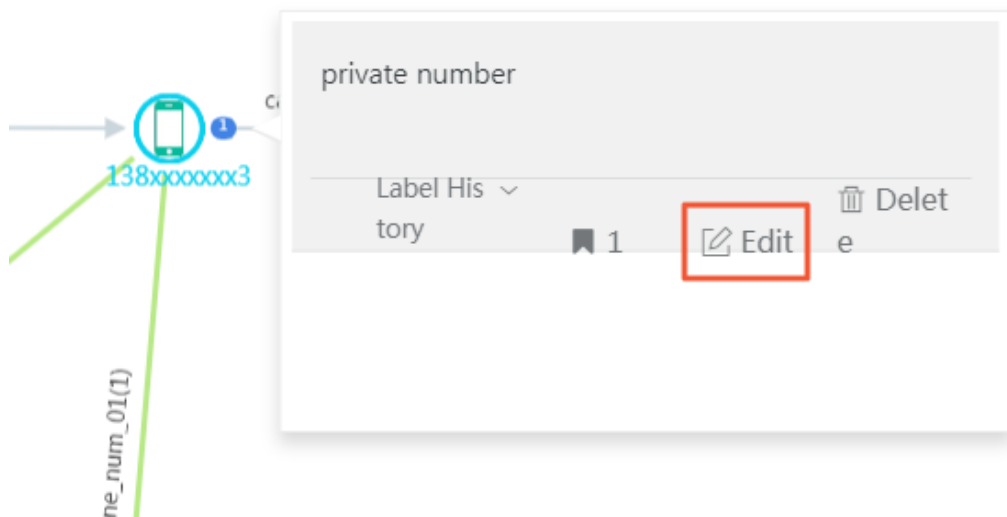
If the name, color, or visible range of a user label is unacceptable, you can edit the user label.

## Prerequisites

- You have obtained the account and password with the permission to perform label-related operations.
- Other users have created user labels that are visible to you.

## Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create an analysis, and add or locate a node that has user labels that are visible to you.
3. Click the label number next to the node, click a label name, and view the label details.



4. Click Edit, and re-set the parameters in the dialog box that appears.

For more information about the parameter settings, see [Table 7-41: Parameters and descriptions](#) in [Add user labels](#).

5. Click OK.

## 7.12.14.7 Delete user labels

You can delete unnecessary user labels.

### Prerequisites

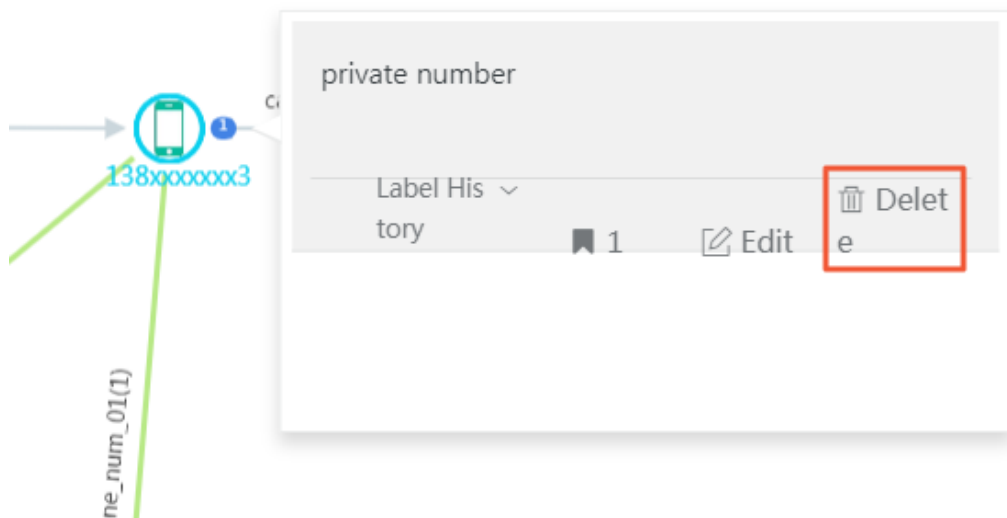
You have obtained the account and password with the permission to perform label-related operations.

### Context


If the label is created by you and the number of likes is 1, you can cancel the like, and the number of likes becomes 0. When the number of likes for a label becomes 0, the system automatically deletes this label.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create an analysis, and add or locate a node that has user labels that are visible to you.
3. Click the label number next to the node, and click a label name to view the detailed information about the label.



4. Click the Delete icon, and click OK in the dialog box that appears. A message is displayed, indicating that the label has been deleted successfully.

If the label is created by you and currently has only one like, you can click the delete icon  to delete the like, and the number of likes becomes 0. The system will automatically delete this label.


## 7.12.15 Save analysis

After the analysis is modified, you must save the modifications before you close the analysis. Graph Analytics provides a screenshot analysis function. When an analysis is saved, Graph Analytics generates a global screenshot for the analysis graph and saves the screenshot to the server.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created a new analysis or modified an existing analysis.

### Procedure

1. [Log on to Analytics Workbench](#).
2. Open an existing analysis file or create a new analysis.
3. Optional: Perform multiple operations on an existing analysis or a new analysis in the graph area, such as adding nodes, setting layouts, and performing analyses.
4. The following may occur when you click the Save icon  in the toolbar.

Operation	Description
Save existing analyses	Saves the analysis content in the graph area by the original name and the original path.
Save new analyses	If you are creating a new analysis, you need to set File Name and Folder of the analysis in the Save Analysis dialog box that appears, and click OK to save the analysis content in the graph area.

## 7.12.16 Print graph areas

Analytics Workbench enables you to print the content of an analysis from different perspectives. You can export the content as a paper document, or save the content as a PNG image.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- Make sure that the system has connected to a printing device.

### Procedure

1. [Log on to Analytics Workbench.](#)
2. Open an existing analysis, or create a new analysis and have the results analyzed.
3. You can print the analysis content or save the analysis content as an image.

Print area	Operation
Print the full graph	<ol style="list-style-type: none"><li>a. In the toolbar, choose Print &gt; Print Full Graph. The print settings page appears.</li><li>b. Set the print parameters, and then click Print to print all the analysis content in the graph area.</li></ol>
Print the visible area	<ol style="list-style-type: none"><li>a. In the toolbar, choose Print &gt; Print Visible Area. The print settings page appears.</li><li>b. Set the print parameters, and then click Print to print the analysis content in the visible area.</li></ol>
Print the selected subgraph	<ol style="list-style-type: none"><li>a. In the toolbar, choose Print &gt; Print Selected. The print settings page appears.</li><li>b. Set the print parameters, and then click Print to print the analysis content of the selected nodes in the graph area.</li></ol>
Save the full graph as an image	In the toolbar, choose Print > Save Full Graph as Image. Save the analysis content as an image in the PNG format.

### 7.12.17 Share analyses

You can use Graph Analytics to share your current analysis with a specific user or group to perform a collaborative analysis. The shared users can view the shared analysis file after they log on to Analytics Workbench.

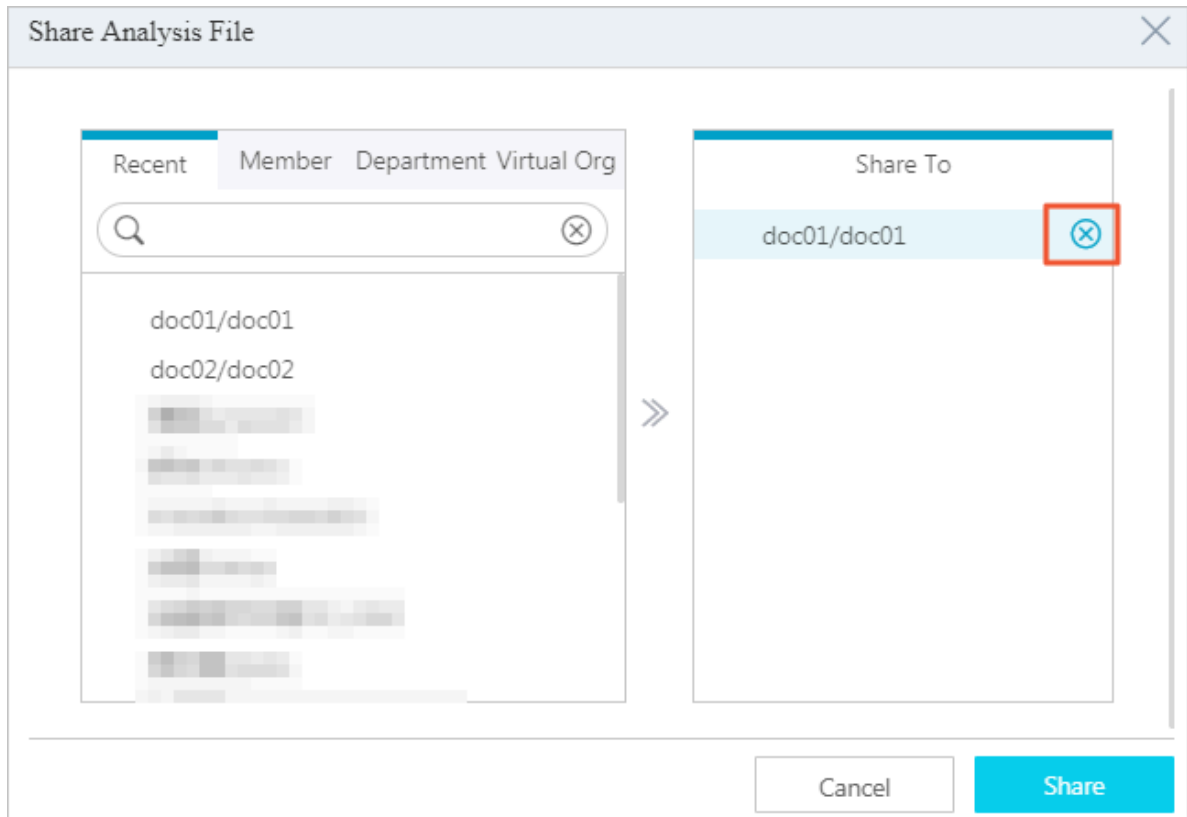
#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Analytics Workbench.](#)
2. Open an existing analysis or create a new analysis.

3. In the toolbar, click the Share icon and set the parameters in the dialog box that appears.



You can select the shared members individually by using the search and positioning feature. You can also select the shared members by department.

Members selected for sharing will be displayed in the Share To list on the right side. When the mouse pointer is moved over the member, you can see a Delete icon. You can click the icon to delete the current member.

4. After you have specified the shared members, click Share, and the system informs you that you have shared the analysis successfully.

### What's next

The user you shared files with can choose File Center > Shared with Me in the top navigation bar to view and operate on the shared analysis files.

After a member receives a shared analysis, the system automatically creates a directory with the same name as the source analysis on the Shared with Me page. By default, the directory has two files: the initial file and the automatically merged file.



## 7.12.18 Behavior chronology

### 7.12.18.1 Details

Details are used to display the link and event details of an object.

#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

#### Procedure

1. [Log on to Analytics Workbench](#).
2. Click an analysis file to open the file in the Graph area.
3. You can open the Behavior Chronology pane by using the following methods.

Method	Procedure
Method 1	Select an object in the Graph area, and click Behavior Chronology in the lower-right corner. The Details area is displayed by default.
Method 2	Select a link in the Graph area, and the Behavior Chronology pane appears automatically. The Details area is displayed by default.

4. On the left side of the Behavior Chronology pane, select a link or an event to view its details.

The screenshot displays the Behavior Chronology interface. At the top, a graph shows three nodes representing phone numbers (138xxxxxxx1, 138xxxxxxx3, 138xxxxxxx2) connected by links labeled 'call\_link\_01(2)', 'call\_link\_01(1)', and 'call\_event\_01-phone\_num\_01(1)'. Below the graph, a table titled 'Export Data' provides details for the selected link 'call\_link\_01'.

caller_num	callee_num	Uploaded By	Upload Type	Uploaded At	Edit
138xxxxxxx1	138xxxxxxx3	System	System	System	<a href="#">Edit</a>
138xxxxxxx3	138xxxxxxx1	System	System	System	<a href="#">Edit</a>
138xxxxxxx2	138xxxxxxx3	System	System	System	<a href="#">Edit</a>

5. Click Export to export the lists of behavior details for all links to an Excel file.

### 7.12.18.2 Behavior analysis

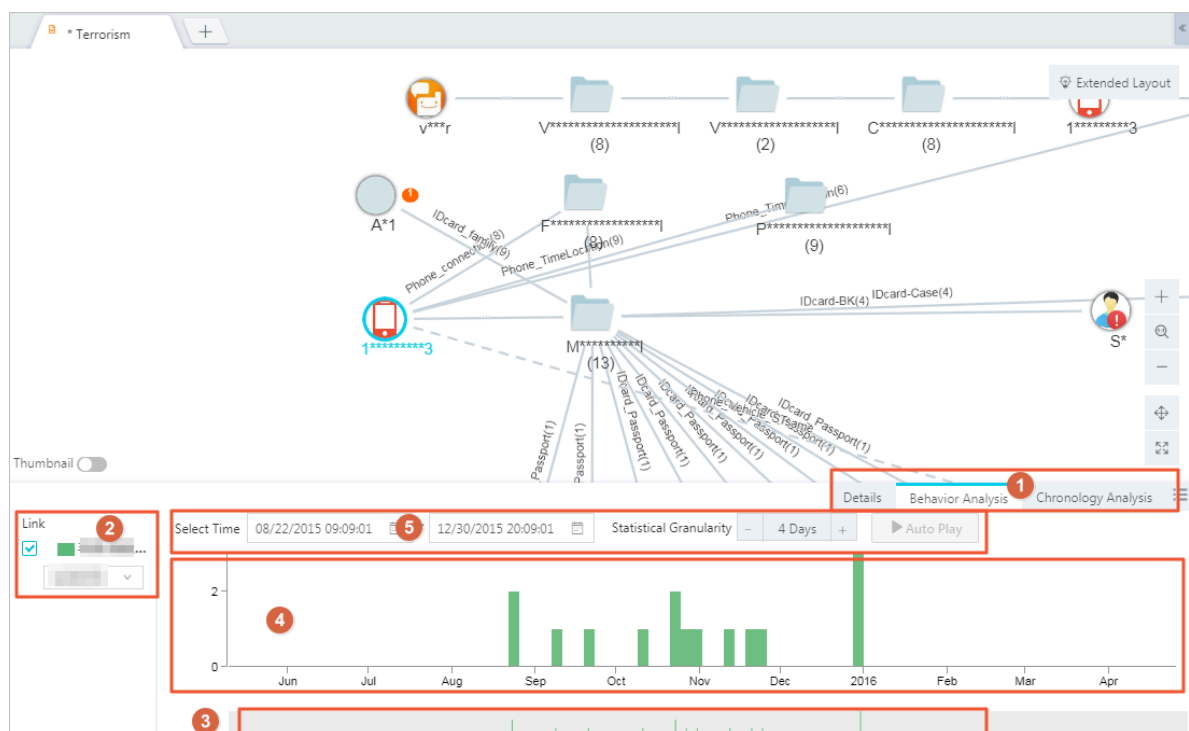
**This feature allows you to display and analyze the link data with time properties on the Graph page.**

## Prerequisites

**Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.**

## Procedure

1. *Log on to Analytics Workbench.*
2. Click an analysis file to open the file in the Graph area.
3. In the Graph area that appears, select an object for the behavior analysis.
4. Click the Behavior Chronology icon in the lower-right corner. In the Behavior Chronology pane that appears, click the Behavior Analysis tab.



**The behavior analysis area is described as follows.**

Table 7-42: Description of the behavior analysis area

Area	Description
Area 1	You can switch to the Behavior Analysis tab.
Area 2	You can filter the links to be analyzed.

Area	Description
Area 3	You can filter data using the thumbnail. You can locate the range through the thumbnail at the bottom, and select the middle part to move the thumbnail.
Area 4	<p>Move your mouse pointer to the column chart. Links that occurred at the current time point and the number of times the links have occurred are displayed.</p> <p>Click the column chart. The links and objects correlated to these links will be highlighted in the Graph area.</p> <p>Click and hold down the left mouse button and drag to select a time range. All links in the time range and objects correlated to these links will be highlighted in the Graph area.</p>
Area 5	<p>You can set the filter time range and the statistical granularity:</p> <ul style="list-style-type: none"> <li>• Set the start time and the end time for time-based filtering.</li> <li>• Click the + icon or the – icon to increase or decrease the statistical granularity.</li> </ul>

5. For more information about how to analyze the behaviors of an object in the Behavior Analysis area, see [Table 7-42: Description of the behavior analysis area](#).

### 7.12.18.3 Chronology analysis

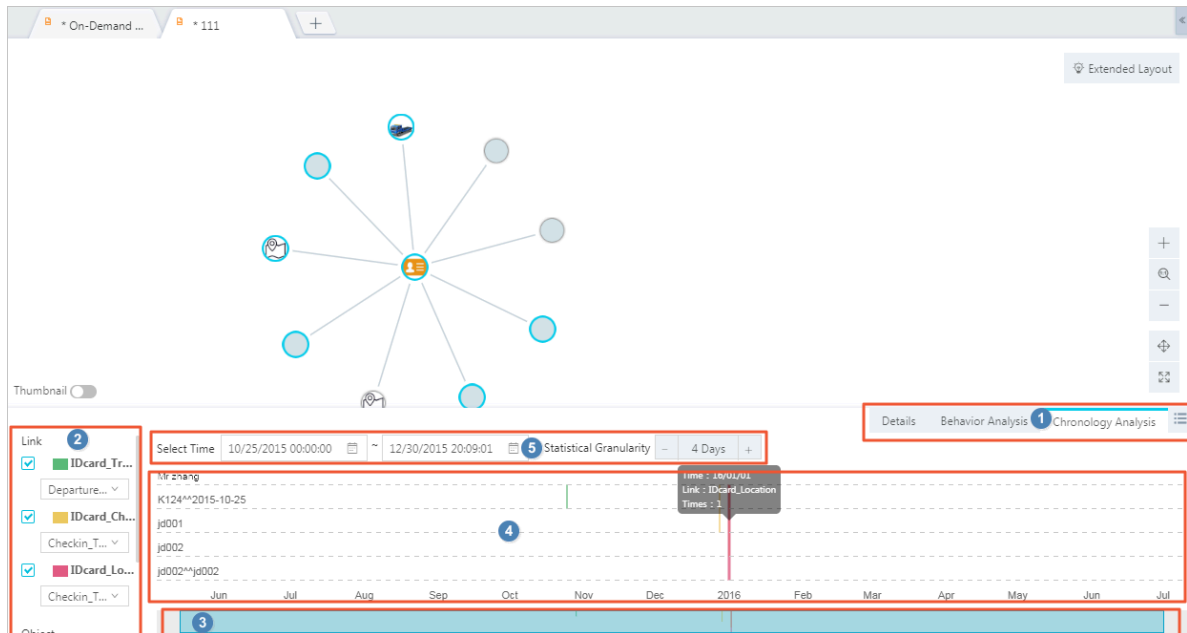
The chronology analysis shows the details of each event based on time.

#### Prerequisites

Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

1. [Log on to Analytics Workbench](#).
2. Click an analysis file to open the file in the Graph area.
3. In the Graph area, select the object and link to perform a chronology analysis.

4. Click the Behavior Chronology icon in the lower-right corner. In the Behavior Chronology pane that appears, click the Chronology Analysis tab.



By default, only five objects can be displayed for an analysis in the chronology analysis area.

The chronology analysis area is described as follows:

Table 7-43: Description of the chronology analysis area

Area	Description
Area 1	You can switch to the Chronology Analysis tab.
Area 2	You can select a link and an object. You can analyze a maximum of five objects at the same time.
Area 3	You can filter data using the thumbnail. You can locate the range through the thumbnail at the bottom, and select the middle part to move the thumbnail.

Area	Description
Area 4	<p>When you move the mouse pointer to a specific chronology line, the details of the chronology appear.</p> <p>When you click the chronology line, the links and the objects involved in these links will be highlighted in the Graph area.</p> <p>Click and hold down the left mouse button and drag to select a time range. All links in the time range and objects correlated to these links will be highlighted in the Graph area.</p>
Area 5	<p>You can set the filter time range and the statistical granularity:</p> <ul style="list-style-type: none"><li>• Set the start time and the end time for time-based filtering.</li><li>• Click the + icon or the - icon to increase or decrease the statistical granularity.</li></ul>

5. For more information about how to analyze the behaviors of an object in the Behavior Analysis area, see [Table 7-43: Description of the chronology analysis area](#).

## 7.12.19 Property statistics

### 7.12.19.1 Details


Typically, when you perform an analysis in Graph Analytics, a large number of objects will be involved. You may need to highlight the objects that are important to you. The properties and statistics area displays data with multiple properties. You can highlight the key objects on the Graph page.

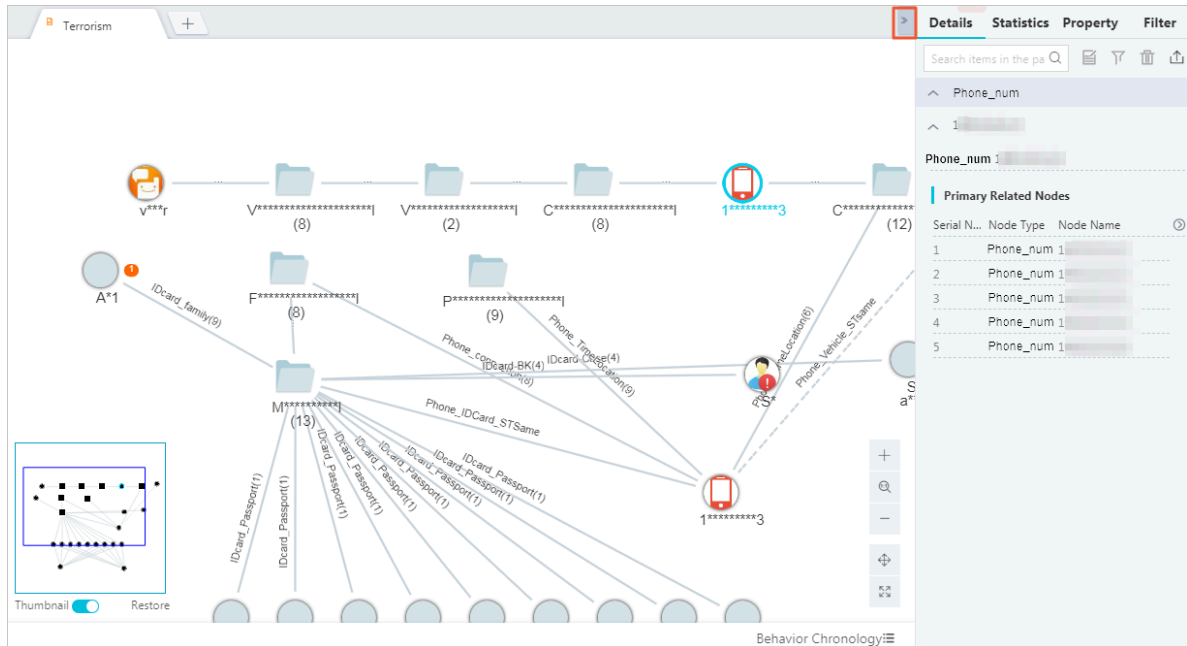
#### Prerequisites

You have obtained an account and a password with the permission to perform network analyses.

#### Procedure

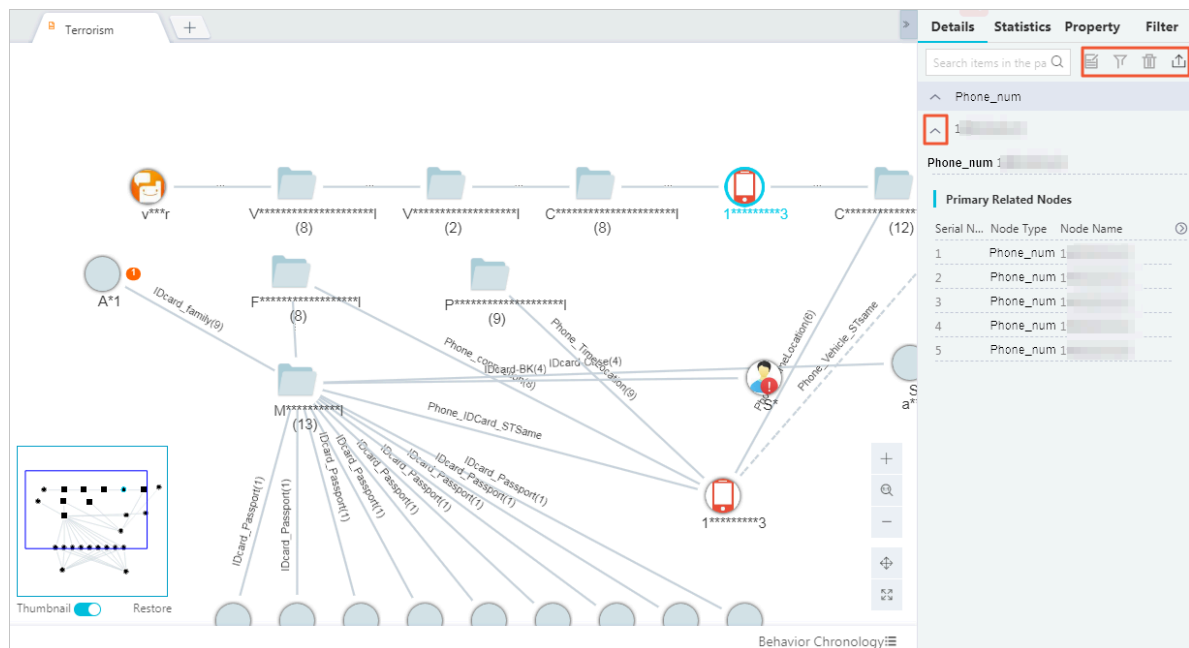
1. [Log on to Analytics Workbench](#).
2. Click an analysis file to open the file in the Graph area.
3. Select one or more objects or links in the Graph area.

- Click the  icon in the right side of the graph area to display the properties and statistics. By default, the Details tab is displayed.



The Details tab displays the basic information of the selected object, including the object type, object icon or avatar, object ID, object properties, and the correlated nodes.

**5. You can perform the following operations in the Details tab.**



Operation	Procedure
Highlight key objects	When you select objects in the right-side pane, the objects, links, and events in the graph area are masked, except for the selected objects. You can press the Control key to select multiple objects.
View object details	<p>When you select nodes or links in the main graph area, the details page on the right side shows all the property information of the selected nodes and links. You can press the Control key to select multiple nodes or links.</p> <p>If you select a link in the graph area, the details of the objects involved in the link are displayed.</p>
Add text notes	Add text notes or labels. For more information, see <a href="#">Add user labels</a> .
Select	Select the node in the graph area.
Delete	Delete the selected object in the graph area.
Export	Export the information of the selected object to an Excel file.

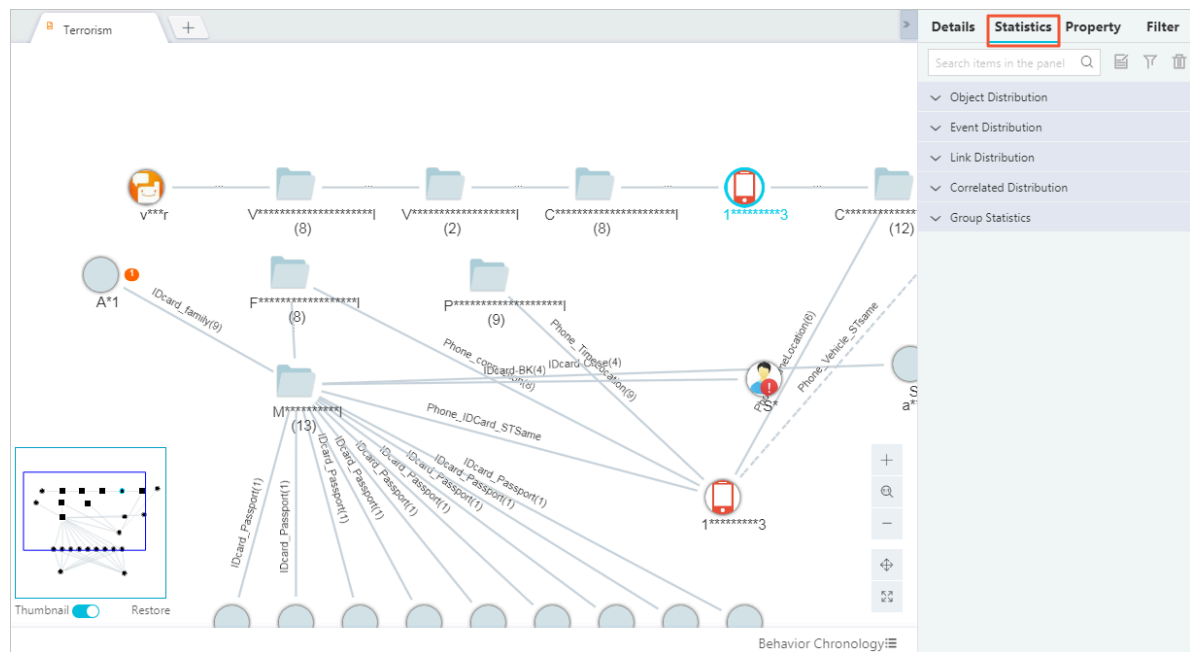
### 7.12.19.2 Statistics

**The Statistics tab displays the statistics of the selected objects or links.**

## Procedure

1. *Log on to Analytics Workbench.*
2. **Select one or more nodes or links in the graph area.**
3. **Click the right arrow in the right side of the graph area to display the properties and statistics.**
4. **Switch to the Statistics tab, as shown in *Figure 7-60: Statistics tab*.**

Figure 7-60: Statistics tab

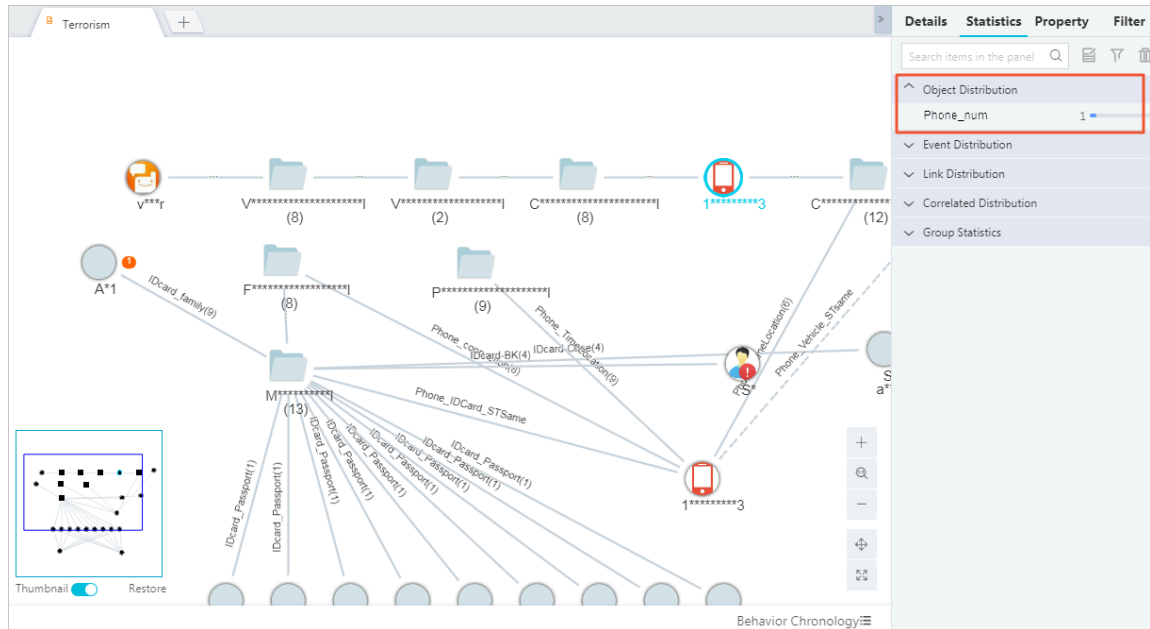




5. Click the up or down arrow to hide or show the properties of the selected object or link.

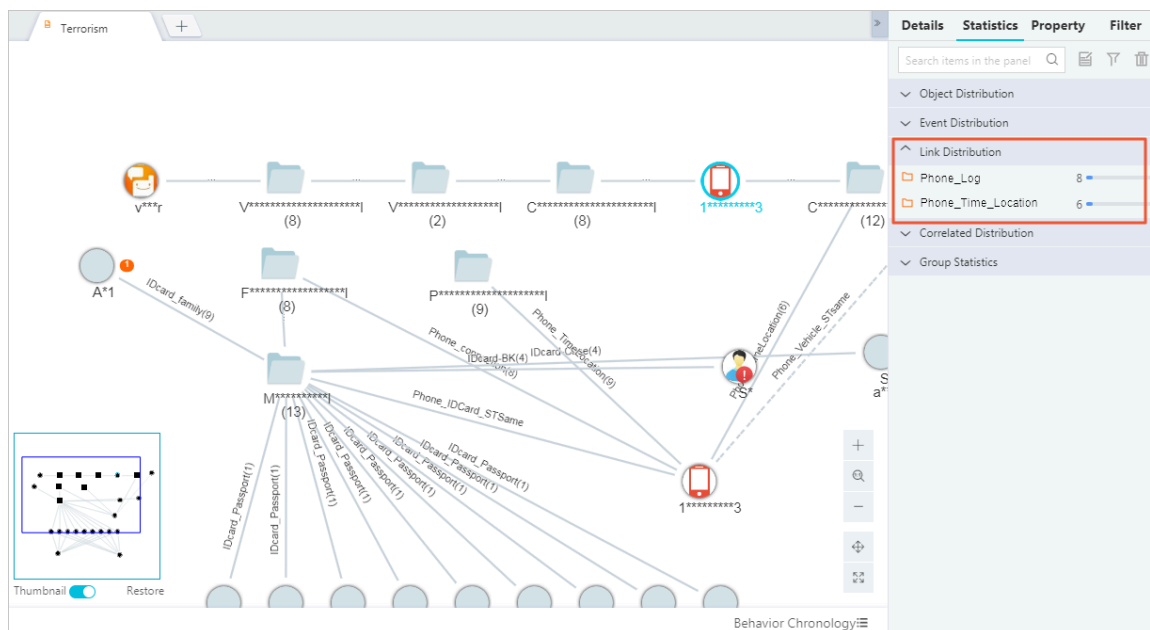
- View the object distributions, as shown in [Figure 7-61: Object distributions](#).

Figure 7-61: Object distributions



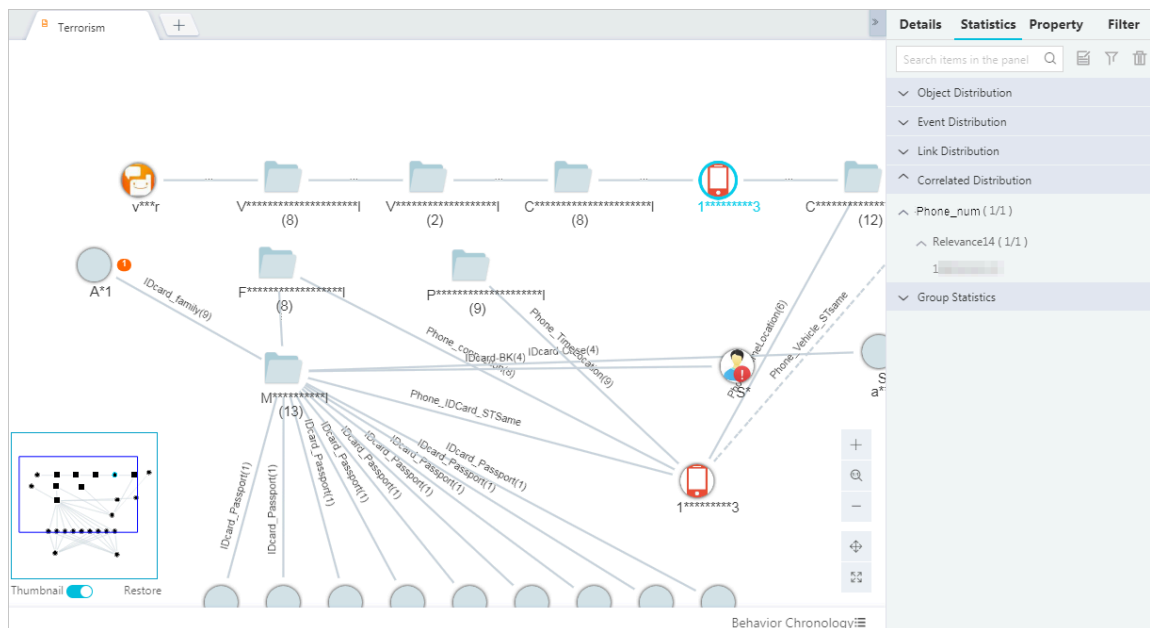
- View the link distributions, as shown in [Figure 7-62: Link distributions](#).

Figure 7-62: Link distributions



- View correlation distributions, as shown in [Figure 7-63: Correlation distributions](#).

Figure 7-63: Correlation distributions

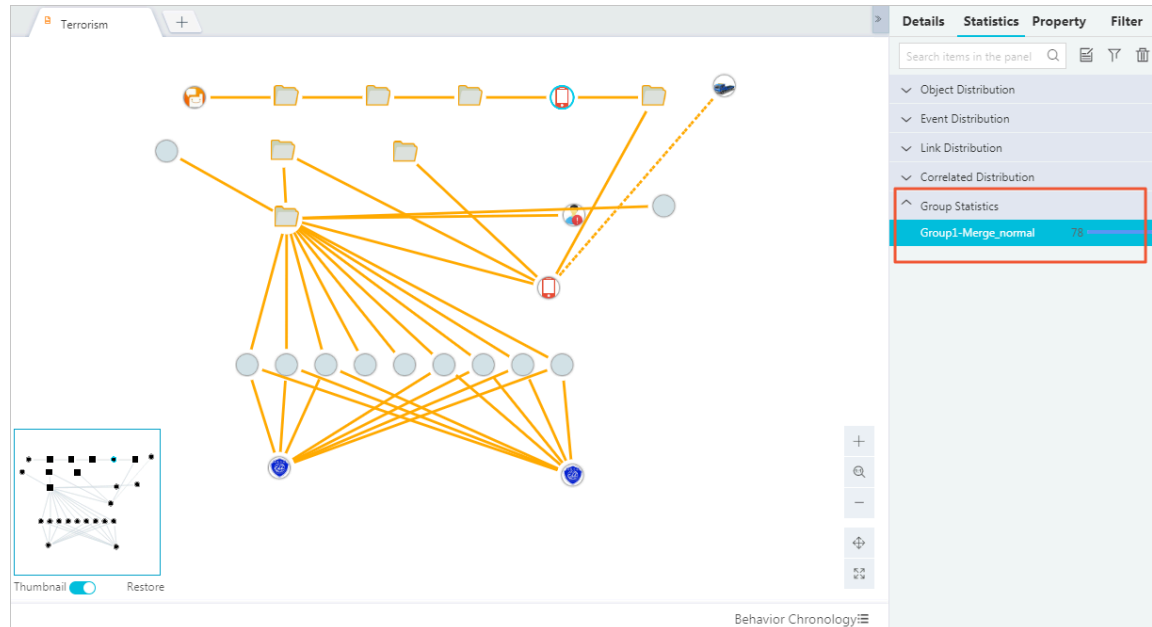


- View the group analysis, as shown in [Figure 7-64: Group analysis](#).

Group statistics allows you to analyze group distributions in Graph Analytics. Group statistics is typically used to perform a group analysis. After you enter multiple nodes, Graph Analytics analyzes the interrelations between these nodes, provides the analysis results, and displays the group distribution. A

group consists of multiple object nodes, with any two object nodes connected topologically. Nodes within a merged node are connected topologically.

Figure 7-64: Group analysis



- In group statistics, all isolated nodes form a group are called isolated nodes.
- The list in group statistics displays the number of nodes and the labels of the nodes that have the highest correlated degree in each group, excluding the group of isolated nodes. Group of isolated nodes is displayed at the top of the list, while other groups are listed in a descending order according to the number of nodes contained in the group.



**Note:**

Group statistics cover all nodes in Graph Analytics, regardless of your selection of nodes during the analysis process.

On the Statistics page, you can perform the following operations for the selected content:

- **Text Note:** You can add text notes. For more information, see [Add user labels](#).
- **Selected:** You can select the node in the graph area.
- **Delete:** You can delete the selected object in the graph area.

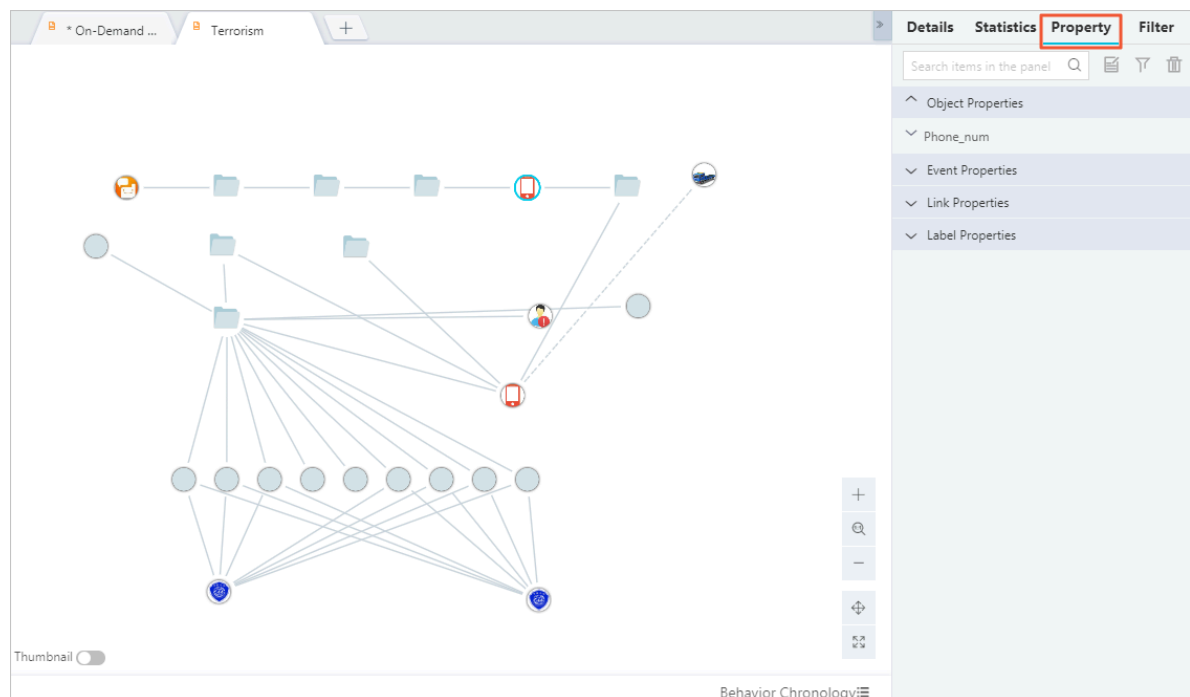
### 7.12.19.3 Property information

The Property tab in the right-side pane of the Graph page displays the object or link information and the label information of the selected objects or links.

#### Procedure

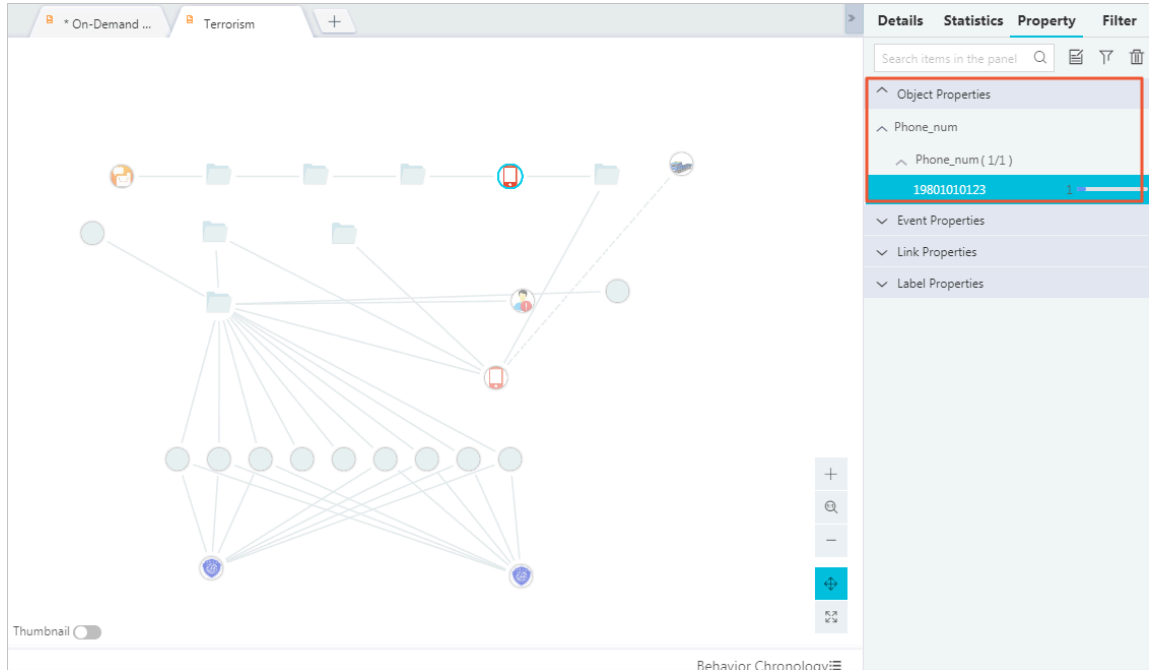
1. [Log on to Analytics Workbench](#).
2. Select one or more nodes or links in the graph area.
3. Click the right arrow in the right side of the graph area to display the properties and statistics.
4. Click the Property tab. The Property tab appears, as shown in [Figure 7-65: Property tab](#).

Figure 7-65: Property tab

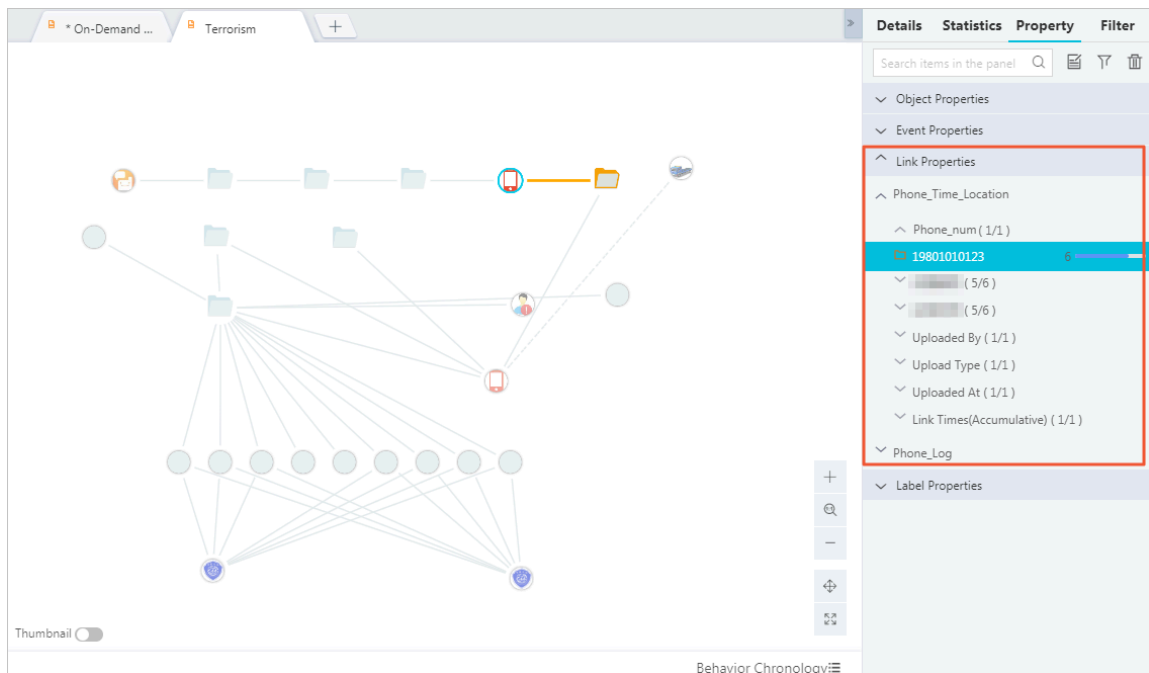


5. Click the up or down arrow to hide or show the properties of the selected object or link.

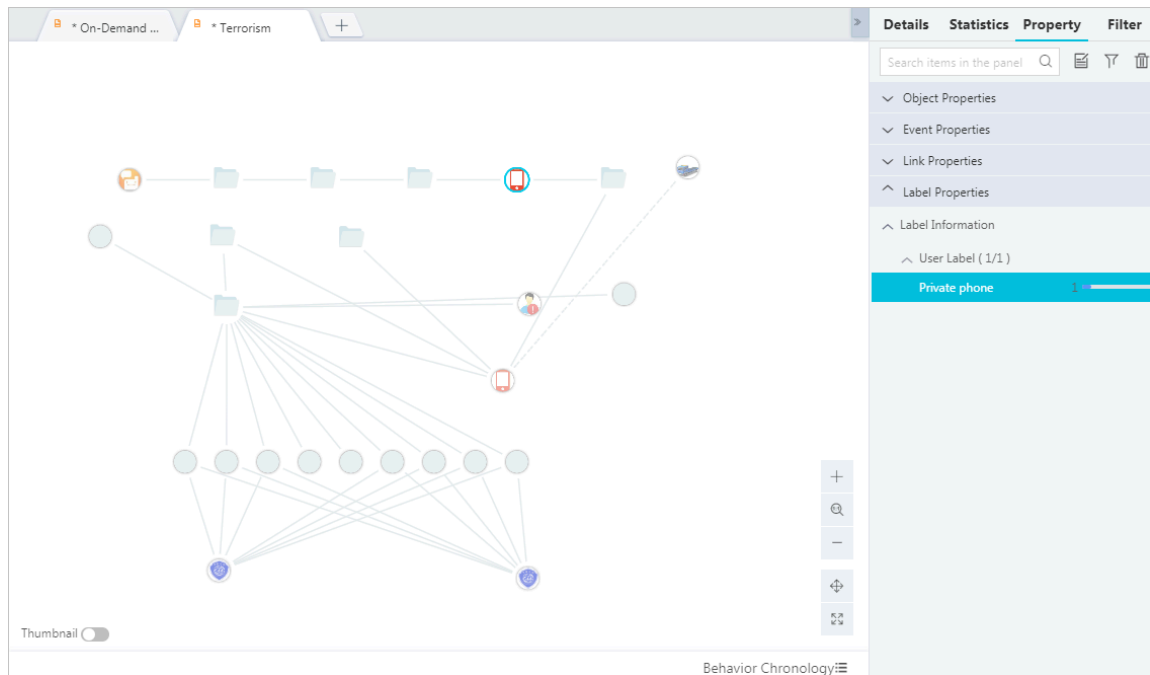
- View object properties.



- View event properties.
- View link properties.



- View label properties.



On the Property tab, you can perform the following operations for the selected content:

- **Text Note:** You can add text notes. For more information, see [Add user labels](#).
- **Selected:** You can select the node in the graph area.
- **Delete:** You can delete the selected object in the graph area.

#### 7.12.19.4 Secondary filtering

You can use the secondary filtering feature to filter objects, links, or events in the canvas. In a complex analysis, you can filter out irrelevant objects, links, or events to keep the content simple and concise.

##### Context

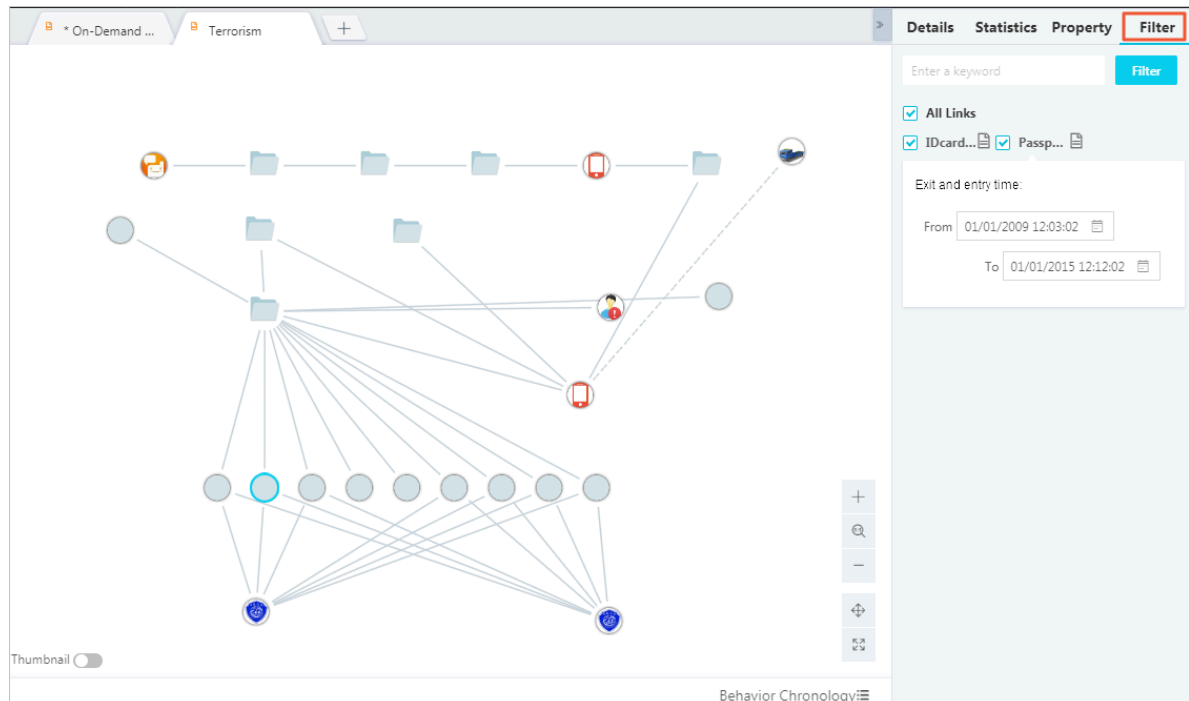
In many service scenarios where data processing is complex, the canvas analysis may become very complex after a few analysis extensions. It is essential to perform secondary filtering before you make further analyses and judgments.

##### Procedure

1. [Log on to Analytics Workbench](#).
2. Click the arrow-pointing-right icon in the right of the graph area to show the properties and statistics.

3. Click the Filter tab, and switch to the Filter page, as shown in [Figure 7-66: Entry to the secondary filtering](#).

Figure 7-66: Entry to the secondary filtering



4. Enter a keyword to be filtered.

Related parameters are described as follows:

- **Time type:** maximum and minimum values of the time property. You can adjust the value range, for example, departure time.
- **Numeric type:** maximum and minimum numerical values. You can adjust the value range, for example, age.
- **Dictionary type:** enumerated values. You can delete some of the enumerated values.
- **Character string:** used for searching. Fuzzy search is supported.

## 7.13 File Center

### 7.13.1 View and manage all analyses

You can see all the personal analysis files and shared analysis files of the current user. The files are arranged in the order of creation time.

#### Prerequisites

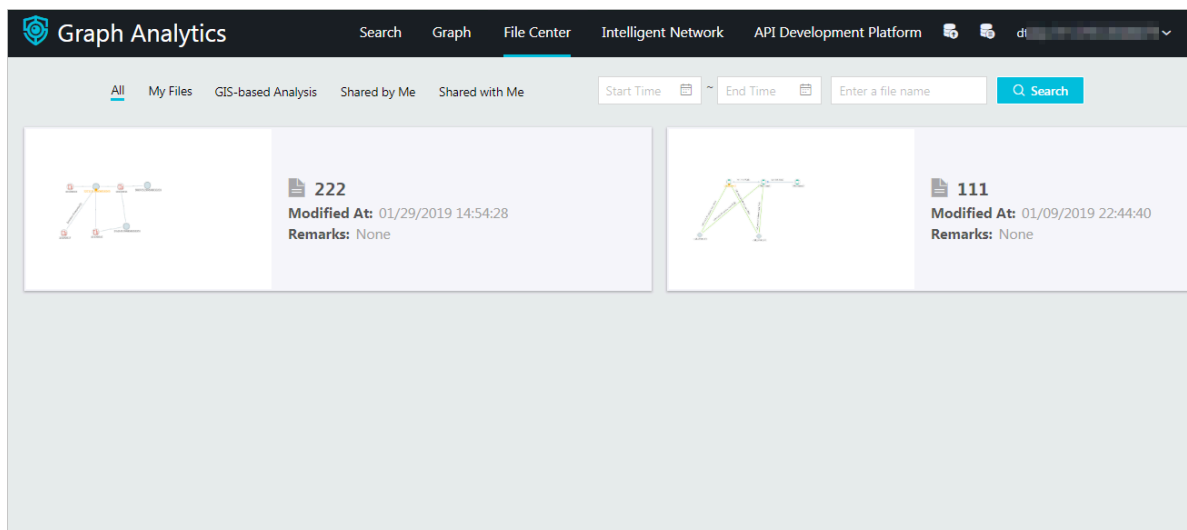
The current user has created and saved analysis files, or has received shared analysis files.

#### Procedures

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click **File Center**, and then click the **All** tab. The **All** tab appears.

The **All** page displays all the shared files and personal files.



When you handle a large number of analysis files, you can use the **Search** feature to find the target files.



3. On the **All** page, you can perform the following operations on each analysis:

Method	Description
Open an analysis file	Double-click an analysis file to open the file in the graph area.
Delete an analysis file	Select an analysis that you created or a shared file that you received, click the Delete icon in the lower-left corner, and click Delete in the dialog box that appears.



Method	Description
Rename an analysis file	<div> <b>Note:</b> This operation is only valid for analysis files created by you.</div> <p>Select an analysis, and click the Rename icon in the lower-left corner. In the dialog box that appears, enter a new name and click OK.</p>
Change sharing permissions	<div> <b>Note:</b> This operation is only valid for files that have been shared by the current user.</div> <p>Select a shared file, and click the Change Sharing Permissions icon in the lower-left corner. In the dialog box that appears, reset the shared members and click Share.</p>

### 7.13.2 View and manage your files

You can view all your personal folders and analysis files in the order of creation time. You can perform operations on the folders and analysis files, such as add, delete, view, and modify.

#### Prerequisites

Make sure that you have created and saved an analysis file.

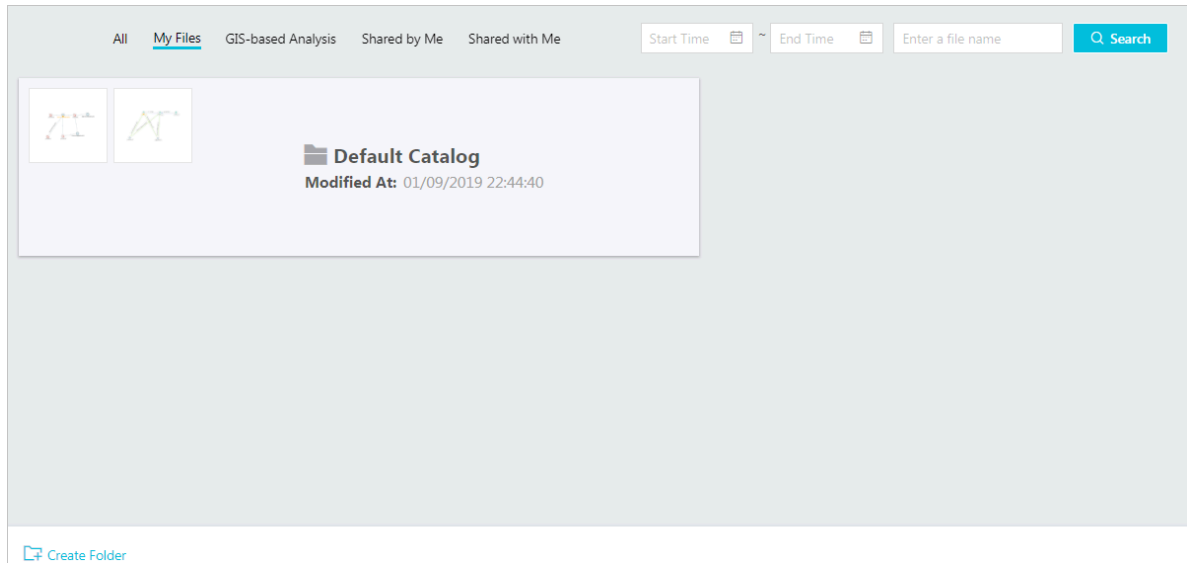
#### Procedure

1. [Log on to Analytics Workbench.](#)

2. In the top navigation bar, click File Center and select the My Files tab. The My Files page appears.

The My Files page displays all folders that contain analysis files that have been saved.

You can use the Search feature to handle a large number of folders or analysis files.



3. On the My Files page, you can perform the following operations.

Operation	Description
Create a folder	Click Create Folder in the lower-left corner of the page. In the dialog box that appears, enter a folder name, and then click OK.
Rename a folder	Select the folder to be renamed, and click Rename in the lower-left corner of the page. In the dialog box that appears, enter a new name, and then click OK.
Delete a folder	Select the folder to be deleted, and click Delete in the lower-left corner of the page. In the message that appears, click Delete.
Rename an analysis file	<ol style="list-style-type: none"> <li>Double-click a folder to open the folder.</li> <li>Select an analysis file to be renamed, and click Rename in the lower-left corner of the page. In the dialog box that appears, rename the analysis file, and then click OK.</li> </ol>

Operation	Description
Delete an analysis file	<ol style="list-style-type: none"><li>Double-click a folder to open the folder.</li><li>Select an analysis file to be deleted, and click Delete in the lower-left corner of the page. In the message that appears, click Delete.</li></ol>
Move an analysis file	<ol style="list-style-type: none"><li>Double-click a folder to open the folder.</li><li>Select the analysis file to be moved, and click Move in the lower-left corner of the page. In the dialog box that appears, select the target folder, and click OK.</li></ol>
Share an analysis file	<ol style="list-style-type: none"><li>Double-click a folder to open the folder.</li><li>Select the analysis file to be shared, and click Share in the lower-left corner of the page. In the dialog box that appears, select the members you want to share the file with, and click Share.</li></ol>

### 7.13.3 My shared items

#### 7.13.3.1 Overview

Graph Analytics allows you to share personal analyses with others. You can share personal ideas and experiences with other users and combine their intelligence and experiences to achieve collaboration and build a better team.

The initiator and the collaborators are the main roles in the collaboration and sharing process. A collaboration proceeds as follows.



After a member receives a shared analysis, the system automatically creates a directory with the same name as the source analysis on the Shared with Me page. By default, the directory has two files: the initial file and the automatically merged file.

Graph Analytics allows you to manage shared files, including deleting files, renaming files, and managing version history.

**Note:**

Only initiators have the permissions to delete and rename shared files.

### 7.13.3.2 View and manage shared files


The Shared by Me page displays all the files shared by the current user in the order of time when the files were created. You can delete folders and change sharing permissions. You can also merge shared files and delete version files.

#### Prerequisites

You have shared an analysis file. For more information, see [Share analyses](#).

#### Context

On the Shared by Me page, you can perform the following operations.

Operation	Description
Delete a shared folder	After the sharer deletes the shared folder, the files will also be deleted from members to which the files are shared.
Modify sharing permissions	After the sharer modifies the sharing permissions, the permissions of the original sharing members are revoked, and the shared files are also deleted.
Delete a shared file	 <b>Note:</b> You cannot delete the initial file or the automatically merged file that exist by default.
Manually merge files	You can merge multiple published versions in a shared folder to form a new version. Only published file versions can be merged. After the files are merged, a manually merged file is generated. If the published versions are different, use the union of the versions.

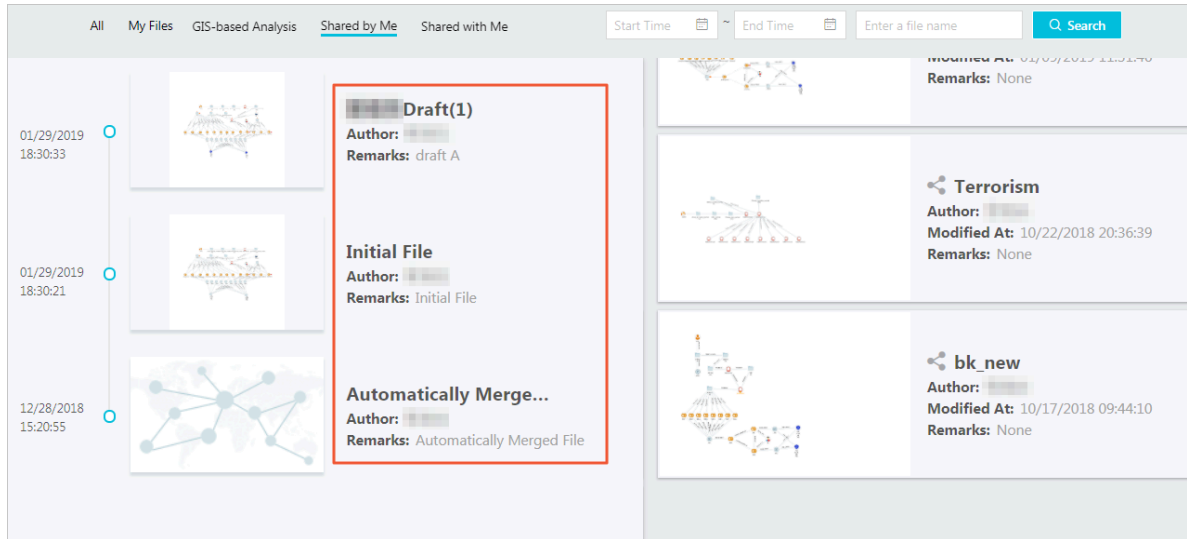
#### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click File Center, and then click the Shared by Me tab.  
The Shared by Me page appears.

The Shared by Me page displays all folders that the current user has shared. Each folder is a shared item. It contains multiple draft files, one initial file,

one automatically merged file, multiple manually merged files, and multiple published version files.

When a large number of folders or files are displayed, you can use the search function to query the target.



3. On the Shared by Me page, you can perform the following operations.

Operation	Description
Delete a shared folder	Select a shared folder, click the Delete icon in the lower-left corner, and click Delete in the dialog box that appears.
Modify sharing permissions	Select a shared folder, and click the Change Sharing Permissions icon in the lower-left corner. In the dialog box that appears, reselect the members to whom you want to share the folder, and click Share.
Delete a shared file version	<p>a. Double-click the shared folder, and select a file. You can delete draft files, manually merged files, and published version files.</p> <p>b. Click the Delete icon in the lower-left corner, and click Delete in the dialog box that appears.</p>
Manually merge files	Select two or more published version files, and click the Merge Files icon in the lower-left corner. In the dialog box that appears, enter the description and click OK.

### 7.13.3.3 Edit a shared file

The sharer and shared members can edit shared files, including initial files and automatically merged files. After a file is edited and saved, a draft file will be generated. Only the current user can view the draft.

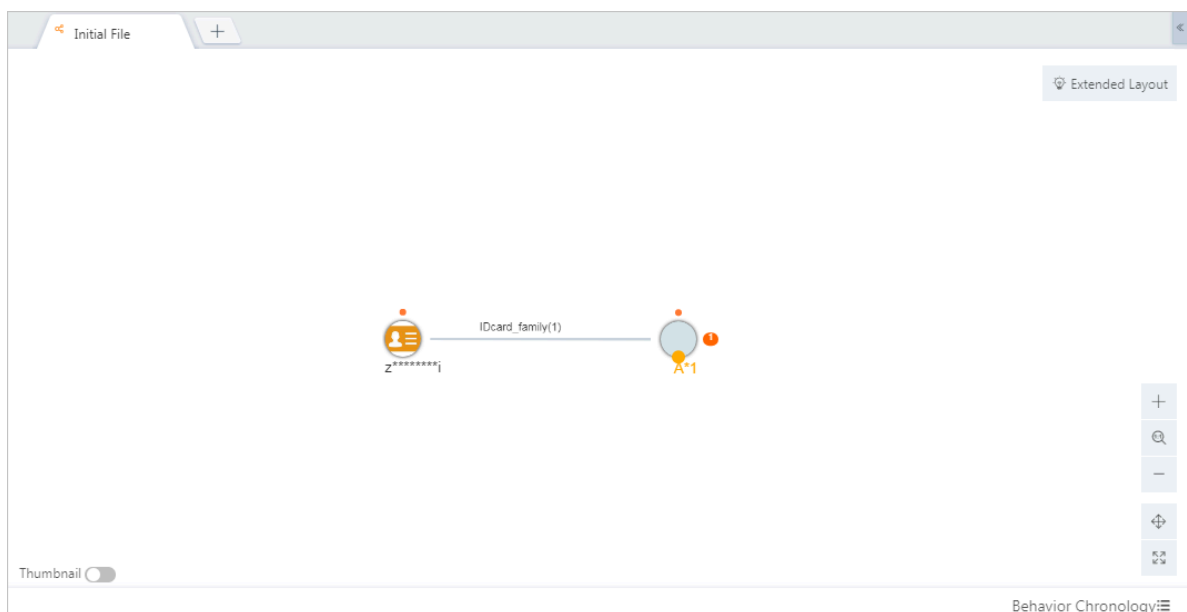
#### Prerequisites

You have shared an analysis file. For more information, see [Share analyses](#).

#### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click **File Center**, and then click the **Shared by Me** tab. The **Shared by Me** page appears.
3. Double-click a shared folder, and then double-click a shared file in the folder, such as the initial file or the automatically merged file, to open the shared file in the graph area.

Each object node in the initial file has a red dot above it to identify itself as the initial node. Subsequently extended object nodes do not have this red dot.

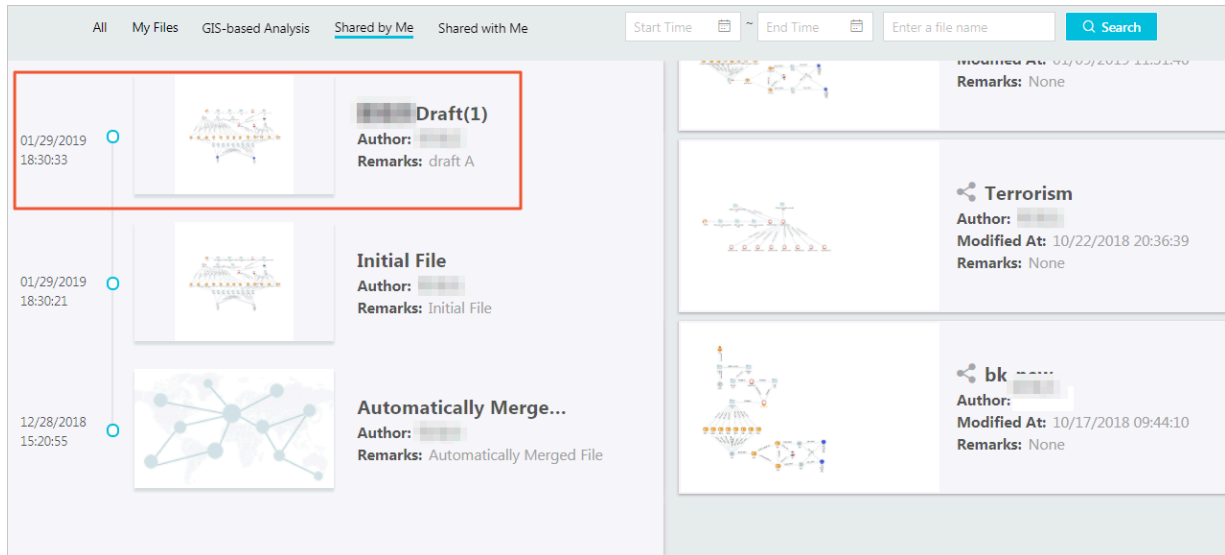


4. Perform analysis or layout operations on the shared file as needed.
5. After the analysis results are shown, click the **Save** icon in the toolbar. In the dialog box that appears, enter the **Draft Statement** and click **Save**.

**Draft Statement:** Enter the draft statement. The statement must be 1 to 200 characters in length.

## Result

After a draft of a shared file is saved, the draft file is displayed in the directory of the shared file. However, the draft is visible only to the current user.



- **Location:** above the initial file.
- **Name format:** username of the current user + draft + (number). For example, test123draft(1).
- **Author:** the user who saved this draft.
- **Description:** displayed below the draft file name.
- **Time:** the time when the draft was created is displayed on the left side of the file.

### 7.13.3.4 Publish a version

You can edit a shared analysis file and publish a new version. The published analysis file can be viewed by all users with whom the file is shared. However, only the sharer and the publisher can delete the file.

## Prerequisites

You have shared an analysis file. For more information, see [Share analyses](#).

## Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click File Center, and then click the Shared by Me tab. The Shared by Me page appears.

3. Double-click a shared folder, and double-click a shared file in the folder, such as the initial file or the automatically merged file, to open the shared file in the graph area. Perform analysis or layout operations on the shared file as needed.
4. After the analysis results are shown, click the Publish icon in the toolbar and set the parameters in the dialog box that appears.

The parameters are described in [Table 7-44: Publish parameters](#).

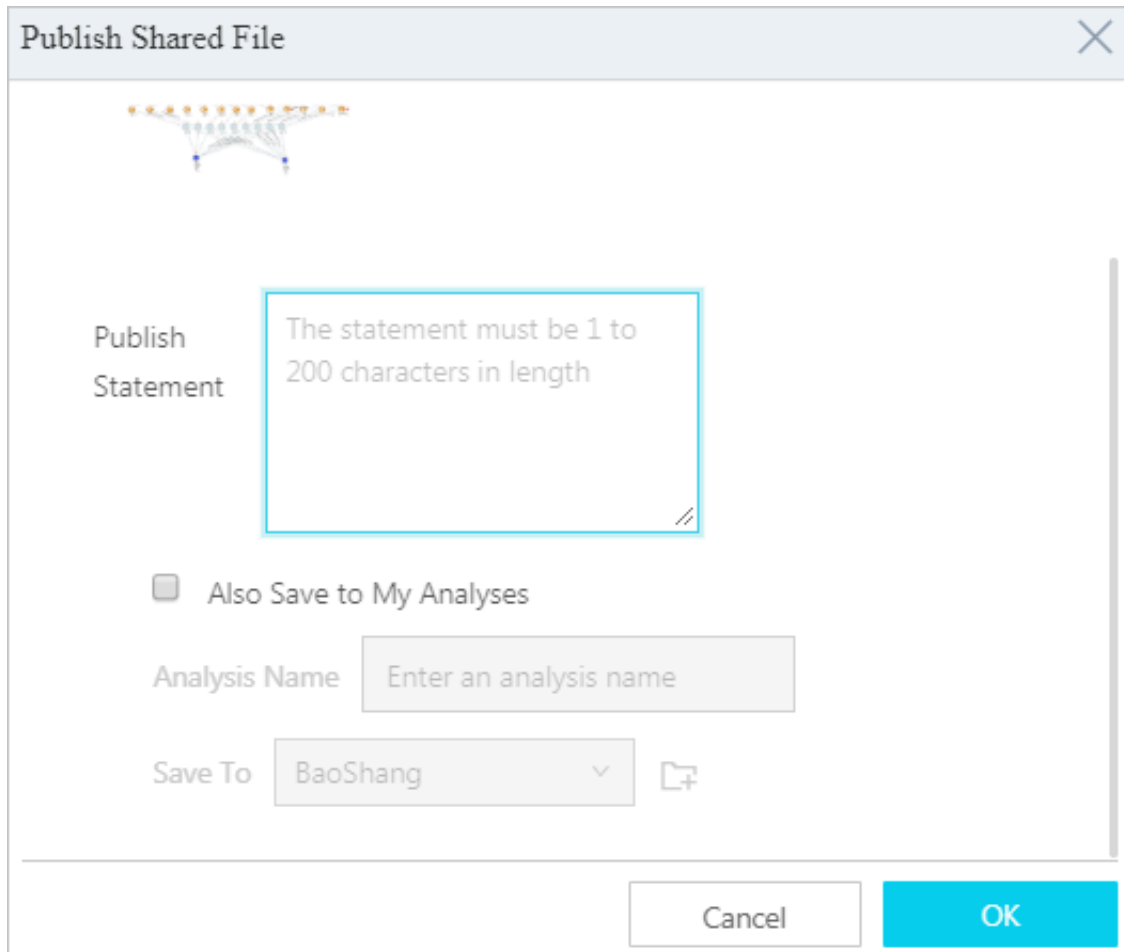


Table 7-44: Publish parameters

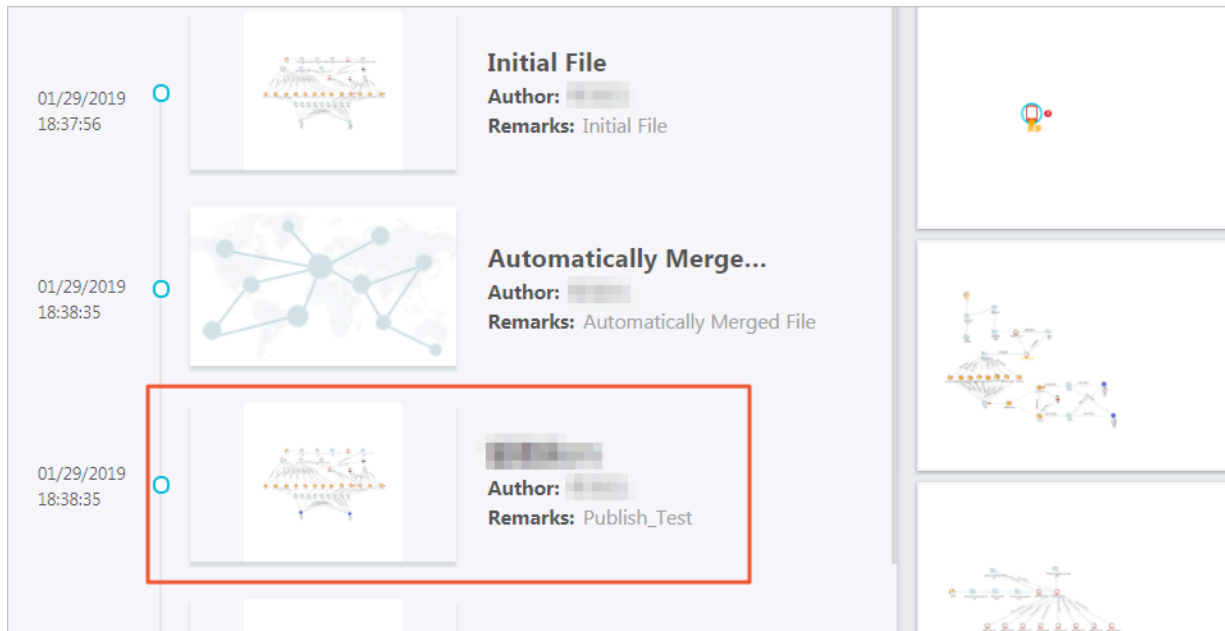
Parameter	Description
Publish Statement	Enter the publish statement. The statement must be 1 to 200 characters in length.
Also Save to My Files	After you select this checkbox, the analysis file is displayed on the My Files tab page. You must enter the analysis name and the directory of the analysis file.

5. Click OK.

## Result



After a new version of a shared analysis is published, the published version file is displayed in the directory of the shared analysis file.



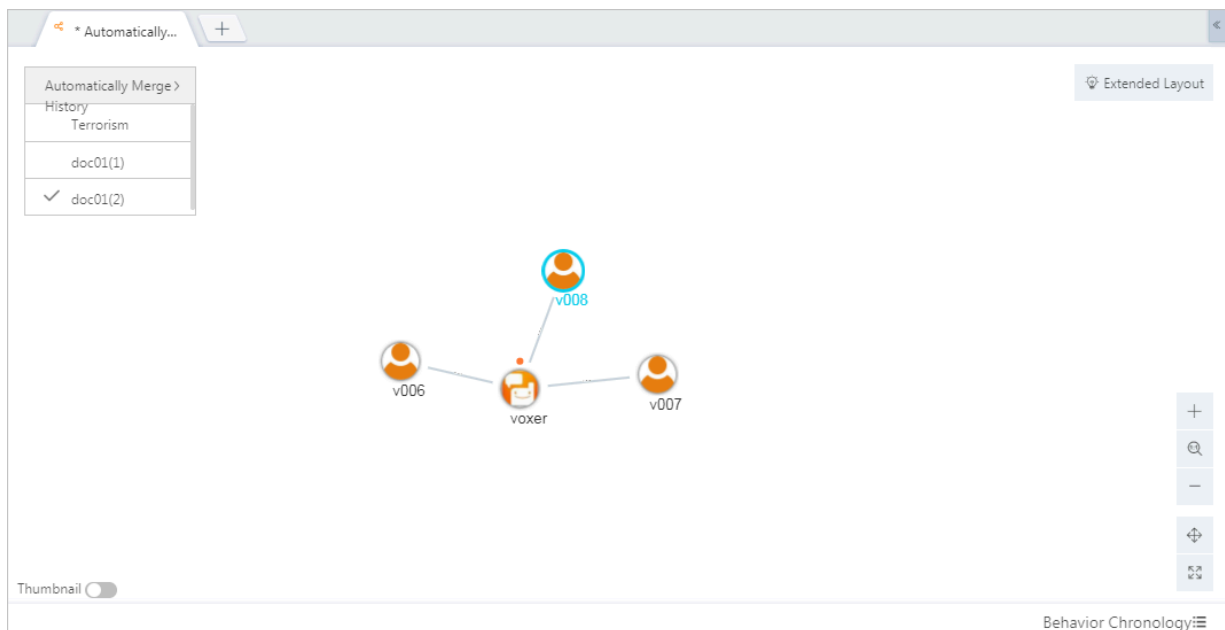
- **Location:** below the automatically merged file or the manually merged files.
- **Name format:** username + (version number). For example, test123(1).
- **Author:** the user who published the file.
- **Description:** displayed below the file name.
- **Time:** the time when the file was generated is displayed on the left side of the file.

### 7.13.3.5 Automatically merge files

When a new version is published, both the shared members and the sharer can use the auto merge feature to merge the new version with the earliest version (the initial file).

After a new version is published, the automatically merged file is updated, as shown in [Figure 7-67: Automatically merge files](#).

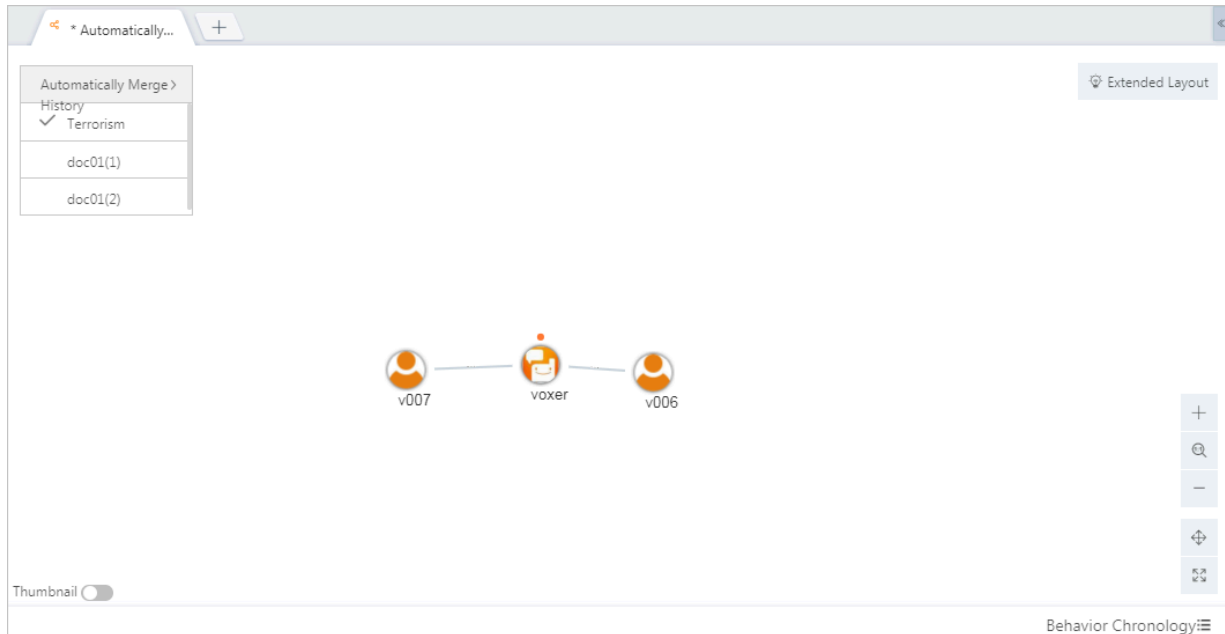
Figure 7-67: Automatically merge files



The versions are listed in the upper-left corner of the graph area. By default, the file of the latest version is displayed.

**You can select an earlier version. Select a version in the Auto Merge History list to view the file of an earlier version, as shown in [Figure 7-68: Select an earlier version](#).**

Figure 7-68: Select an earlier version



### 7.13.4 View and manage received shared files

**You can view and modify the analysis files shared by other users on the Shared with Me page. However, you cannot delete these files.**

#### Prerequisites

**You have received a shared file.**

#### Context

**You can edit shared files, delete drafts, and publish new versions on the Shared with Me page.**

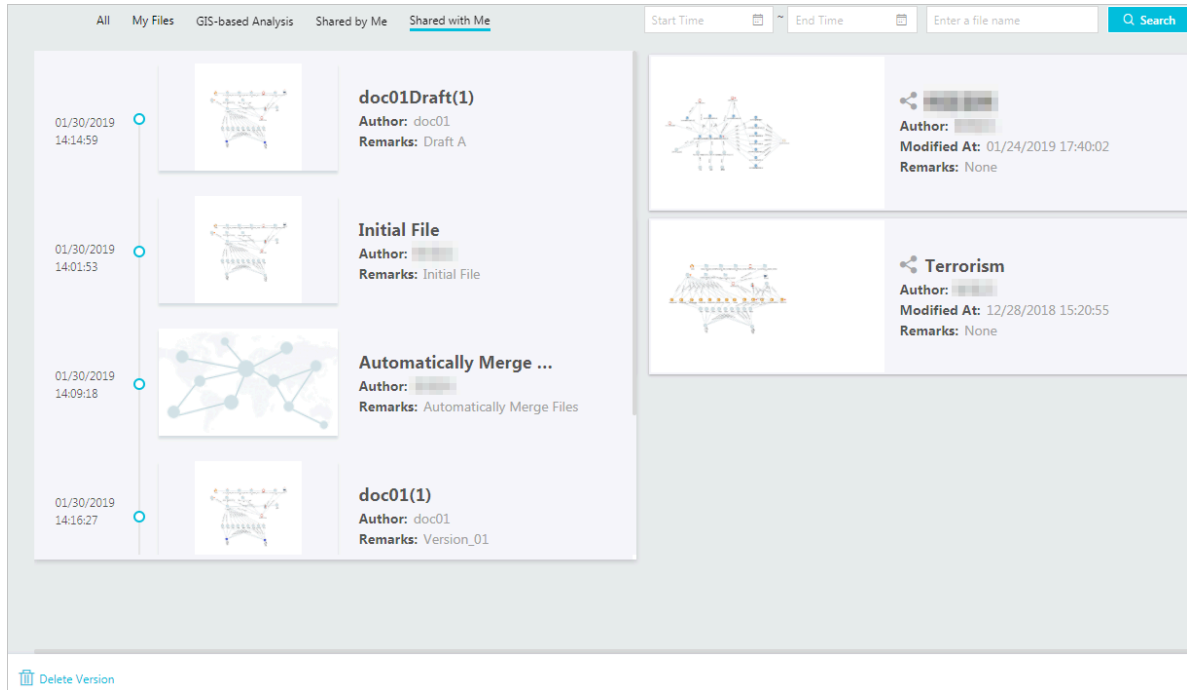
#### Procedure

1. [Log on to Analytics Workbench](#).
2. **In the top navigation bar, click File Center, and then click the Shared with Me tab. The Shared with Me page appears.**

**The Shared with Me page displays all shared folders received by the current user. Each folder is a shared item. It contains multiple draft files, one initial file,**

one automatically merged file, multiple manually merged files, and multiple published version files.

When a large number of folders or files are displayed, you can use the search function to query the target.



3. On the Shared with Me page, you can perform the following operations.

Operation	Description
Edit a shared file	Double-click an initial file, an automatically merged file, or a published version file to open the file in the graph area. Edit and save the file, and a draft file is generated.
Delete a draft	Select the draft to be deleted, click the Delete Version icon in the lower-left corner, and click Delete in the dialog box that appears.
Publish a version	See <a href="#">Publish a version</a> .

## 7.14 Intelligent Network

## 7.14.1 Intelligent Network overview

**This topic describes related concepts of Intelligent Network and how to use Intelligent Network.**

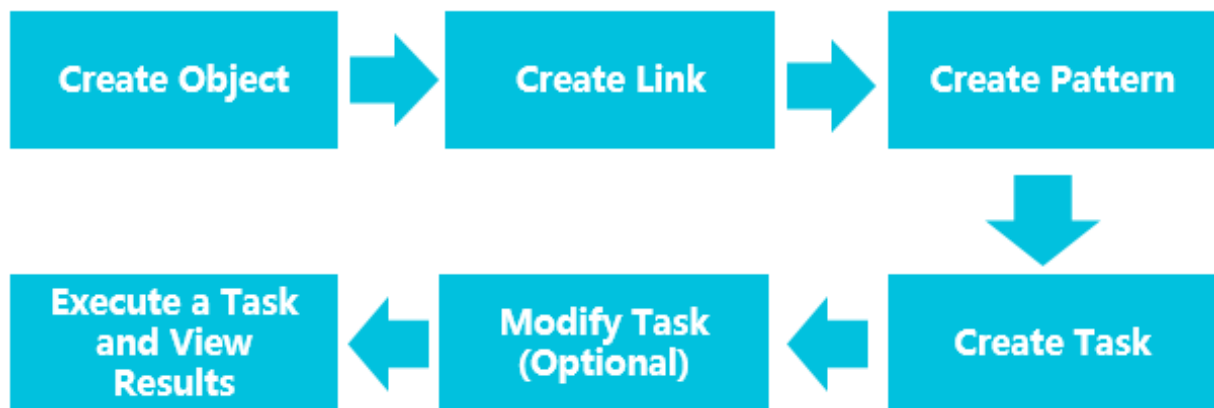
### Concepts

**Intelligent Network allows you to query subgraphs with the same graph structure as a task specified in a predefined pattern.**

**Intelligent Network involves the following concepts:**

- **Pattern:** the relationship graph structure model that is predefined in Intelligent Network. Patterns are divided into private patterns and public patterns.
  - **Private pattern:** Only administrators and creators can use private patterns to create private tasks. Private patterns can be set to public patterns, but this is an irreversible operation.
  - **Public pattern:** All users can use public patterns to create public or private tasks. Public patterns cannot be set to private patterns.
- **Task:** created based on the pattern and used to query data with the same graph structure as the task in the data source. After a task is created based on a pattern, the task has the same settings as the pattern. You can modify the graph structure and filter conditions of the task. Tasks are divided into private tasks and public tasks.
  - **Private task:** Only administrators and creators can use private tasks. Private tasks created based on public patterns can be set to public tasks, but this is an irreversible operation.
  - **Public tasks:** All users can use public tasks. Public tasks cannot be set to private tasks.

Procedure to use Intelligent Network



## 7.14.2 Patterns

### 7.14.2.1 Create patterns

A pattern is the relationship graph structure model predefined in Intelligent Network. It is the basis of creating a task. Before you use the intelligent network to query a sub-graph, you must use related objects and links to predefine a pattern.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created an object and a first-degree link correlated with the object. For more information about how to create an object and a first-degree link, see [Create an object](#) and [Create a first-degree link](#).

#### Context

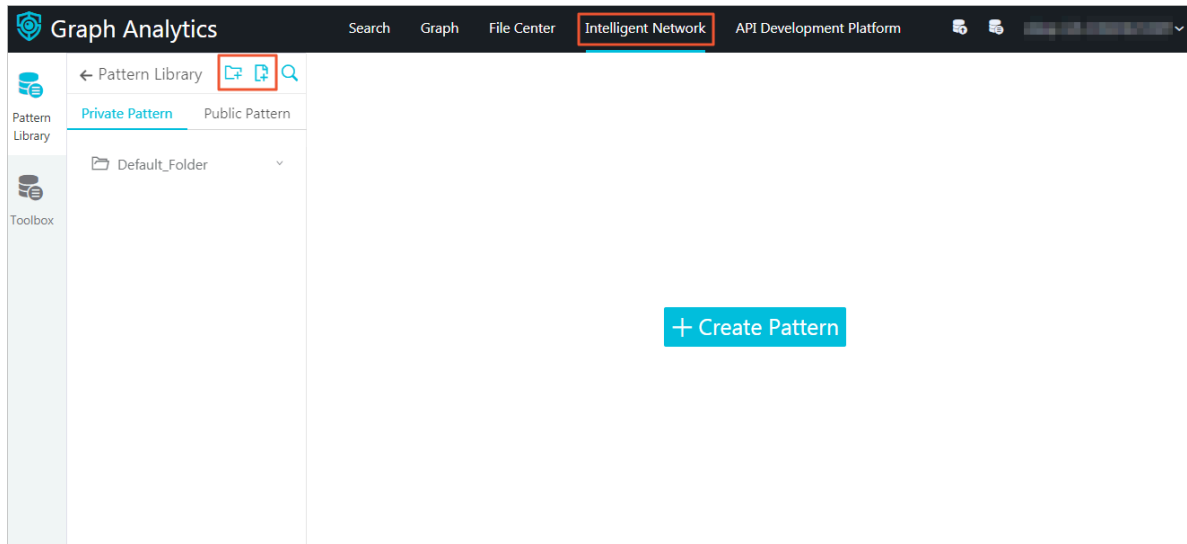
You can set the filter conditions in the Property tab and the Global Conditions tab based on your requirements.


- Property displays the properties of objects and links that can be used as query conditions. The Conditional Query properties and properties in Accumulative Statistics Settings will be displayed on the Property tab page.
- Global Conditions are based on the number and time type properties in a link. Global conditions take effect on the entire pattern. To configure Global Conditions, you must name the relevant number or time type properties on the Property tab.


The following example shows a transfer pattern (A>B>C>A). The global condition is that the transfer from A to B is made earlier than the transfer from B to C.

## Procedure



1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.




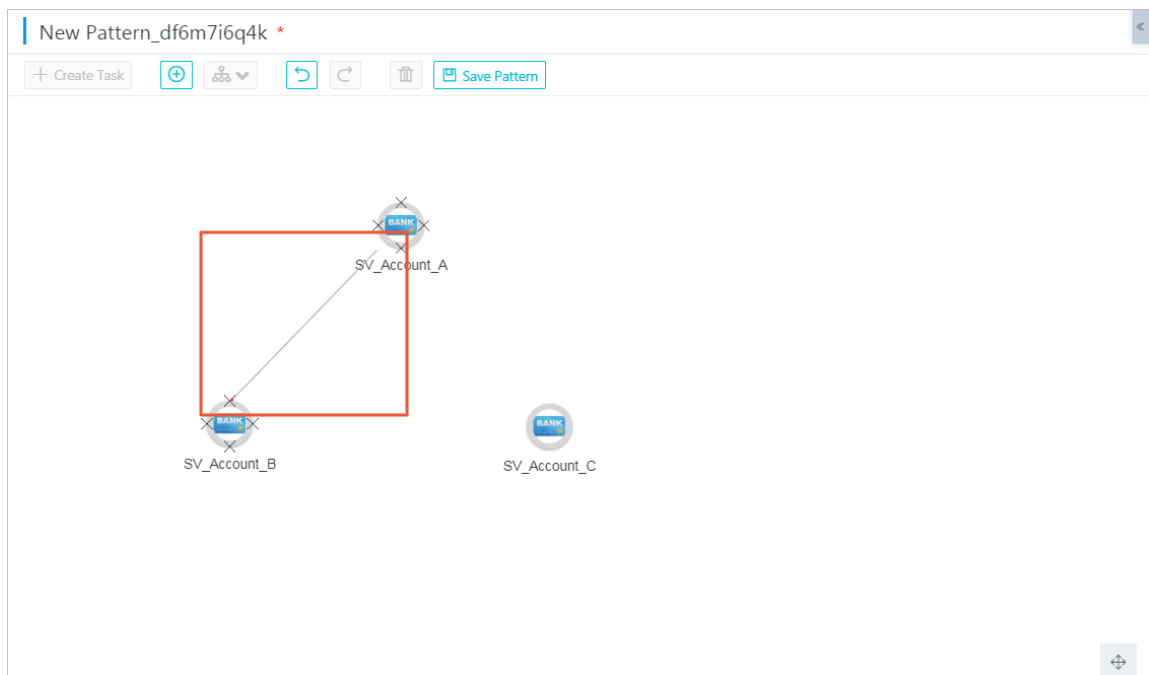
3. You can create a new folder to save the new pattern separately.
  - a) In the Pattern Library navigation pane, click the  icon (Create Folder).
  - b) Specify the Directory Name and the pattern type as needed.


- c) Click OK.
4. In the Pattern Library navigation pane, click the  icon (Create Pattern) or Create Pattern in the blank area on the right side.

## 5. Configure the relationship graph structure model of a pattern.

If an incorrect operation occurs, you can click the  icon (Undo) and the  icon (Redo) to quickly restore to a specific state.



- For the new pattern New pattern\_XXX, click the  icon in the toolbar, or click **Toolbox** in the left-side navigation pane to open the tool box.
- Drag the objects one by one from the toolbox to the right side of the page to add object nodes to New mode\_XXX.
- Right-click the object node and select **Change Node Name** to set the name of the object node.
- Move your mouse pointer to the source object. When four cross signs appear around the object, move your mouse pointer to a cross sign and press the left mouse button. Then, drag the mouse pointer to the target object. When four cross signs appear around the target object, release the left mouse button.

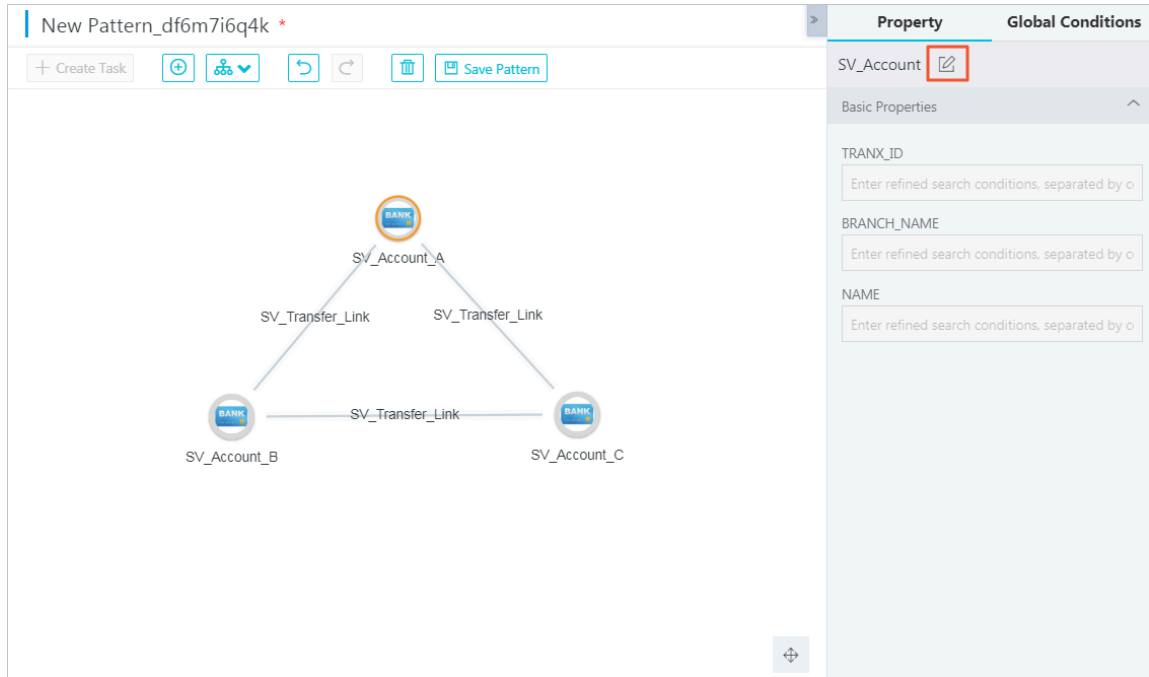




- In the **Add Link** dialog box that appears, select the link type and specify the link logic.
- Click **OK**.
- Optional: To handle multiple object nodes with complex relationships, you can select the corresponding object node and click the  icon to select an appropriate layout.



## 6. Configure the filter conditions in the Property tab.

- Click the  icon in the upper-right corner, and then click Property.
- Select an object. On the Property tab, click the  icon to set the filter conditions for the object.



- After you have configured the preceding parameters, click the  icon to save the Property configurations of the object.
- Select a link. On the Property tab, click the  icon to set the filter conditions and direction of the link.

The screenshot displays the 'New Pattern\_df6m7i6q4k' window. On the left, a graph pattern is shown with three nodes: SV\_Account\_A, SV\_Account\_B, and SV\_Account\_C, each marked with a 'BANK' icon. They are connected by three edges, all labeled 'SV\_Transfer\_Link'. One edge (A to B) is light blue, another (A to C) is dark blue, and the third (B to C) is grey. The top toolbar includes buttons for '+ Create Task', a plus icon, a minus icon, a refresh icon, a save icon, and a 'Save Pattern' button. On the right, the 'Property' tab is active, showing a list of properties for 'SV\_Transfer\_Link'. The 'Direction' section has four radio button options, with 'SV\_Account\_C -> SV\_Account\_A' selected. The 'Basic Properties' section lists several properties: 'transaction\_serial\_num', 'transaction\_date', 'transaction\_time', 'transaction\_organization\_name', and 'transaction\_accout\_name'. Each property has a text input field for refined search conditions. The 'transaction\_date' field is marked with an 'FX' icon, indicating it is a global condition.


**On the Property page of a link, properties marked with FX are the basis of Global Conditions.**

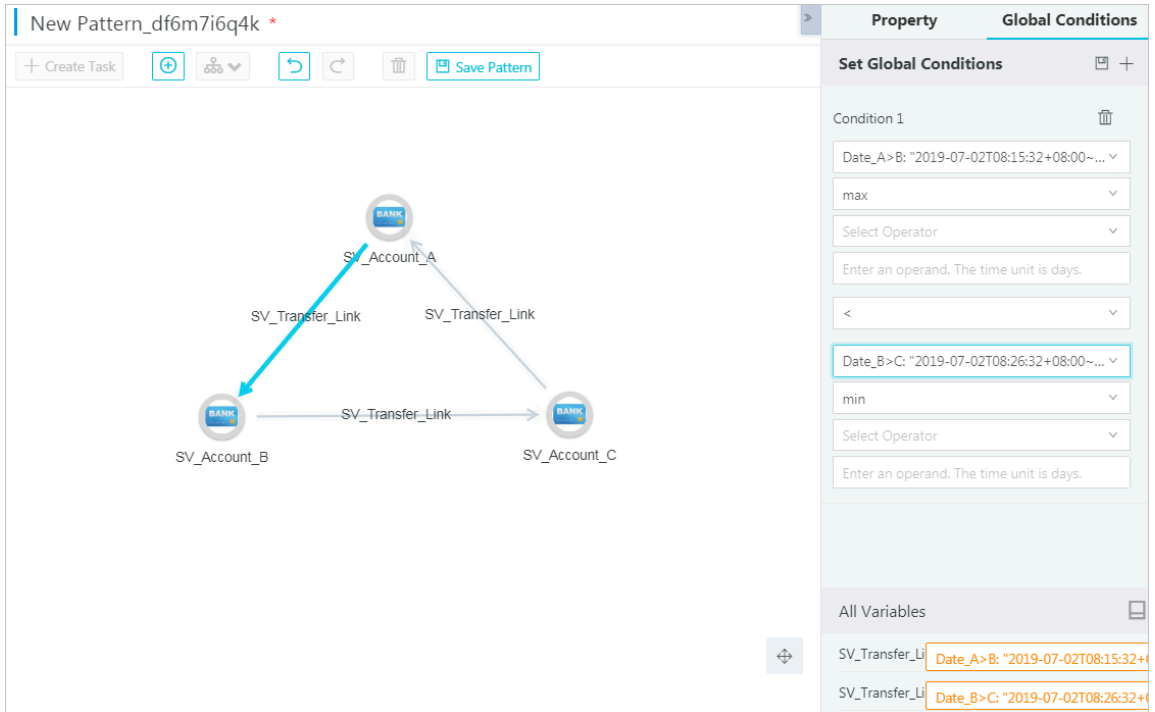
- e) To set the global conditions based on a property marked with FX, click FX to name the property.

**In this example: the Transaction Date property of the link between A and B is named as Date A>B. The Transaction Date property of the link between B and C is named as Date B>C.**

- f) After you have configured the preceding parameters, click the  icon to save the Property configurations of the link.

## 7. Configure filter conditions on the Global Conditions tab.

- Click the  icon in the upper-right corner, and then click Global Conditions.
- If you have not set the global conditions, click [Click here to configure settings.](#)



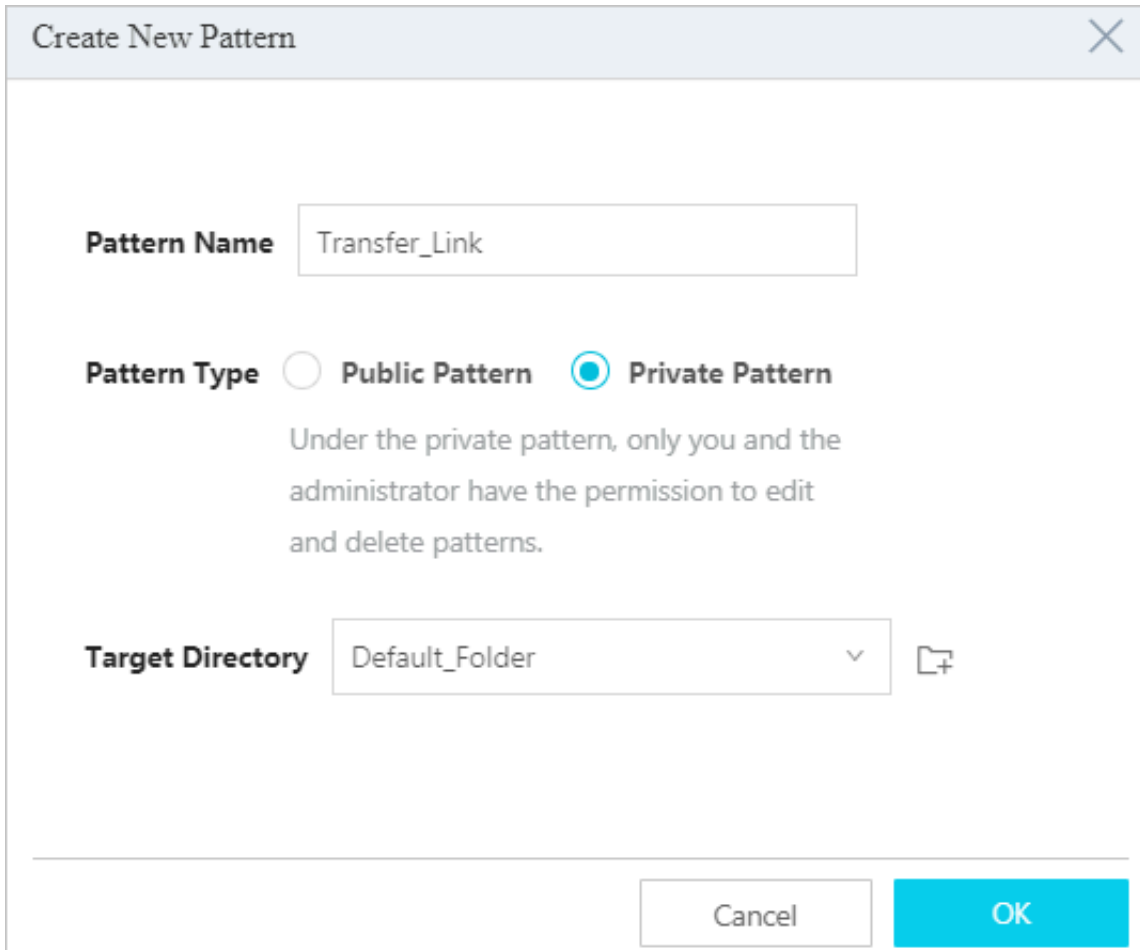
The screenshot displays the 'Global Conditions' configuration interface. The main workspace shows a graph with three bank nodes (SV\_Account\_A, SV\_Account\_B, SV\_Account\_C) connected by transfer links (SV\_Transfer\_Link). The right sidebar is titled 'Global Conditions' and contains a 'Set Global Conditions' section. It shows two conditions: 'Date\_A > B: 2019-07-02T08:15:32+08:00~...' and 'Date\_B > C: 2019-07-02T08:26:32+08:00~...'. Below these, there is an 'All Variables' section with a list of variables including SV\_Transfer\_Li.

On the Global Conditions tab, configure the global conditions based on the link properties that are marked with FX in the Property tab. The global condition in this example is that the transfer from A to B is made earlier than the transfer from B to C.


- After you have configured the preceding parameters, click the  icon to save the configurations on the Global Conditions tab.

8. Click Save Pattern. Set the parameters in the Create Pattern dialog box that appears.

If you cannot find a proper folder in Target Directory, you can click the  icon to create a folder.



The 'Create New Pattern' dialog box contains the following fields and options:

- Pattern Name:** A text input field containing 'Transfer\_Link'.
- Pattern Type:** Two radio buttons. 'Public Pattern' is unselected, and 'Private Pattern' is selected (indicated by a blue dot).
- Private Pattern Note:** A text block stating: 'Under the private pattern, only you and the administrator have the permission to edit and delete patterns.'
- Target Directory:** A dropdown menu showing 'Default\_Folder' with a downward arrow. To the right of the dropdown is a folder creation icon (.
- Buttons:** 'Cancel' and 'OK' buttons at the bottom right.

9. Click OK.

### 7.14.2.2 View patterns

In Graph Analytics, you can view all the patterns within your permissions and details of each pattern, such as the relationship graph structures and tasks created based on the pattern.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



**Note:**

**Only administrators and creators can view the private patterns.**

- You have created a pattern. For more information about how to create a pattern, see [Create patterns](#).

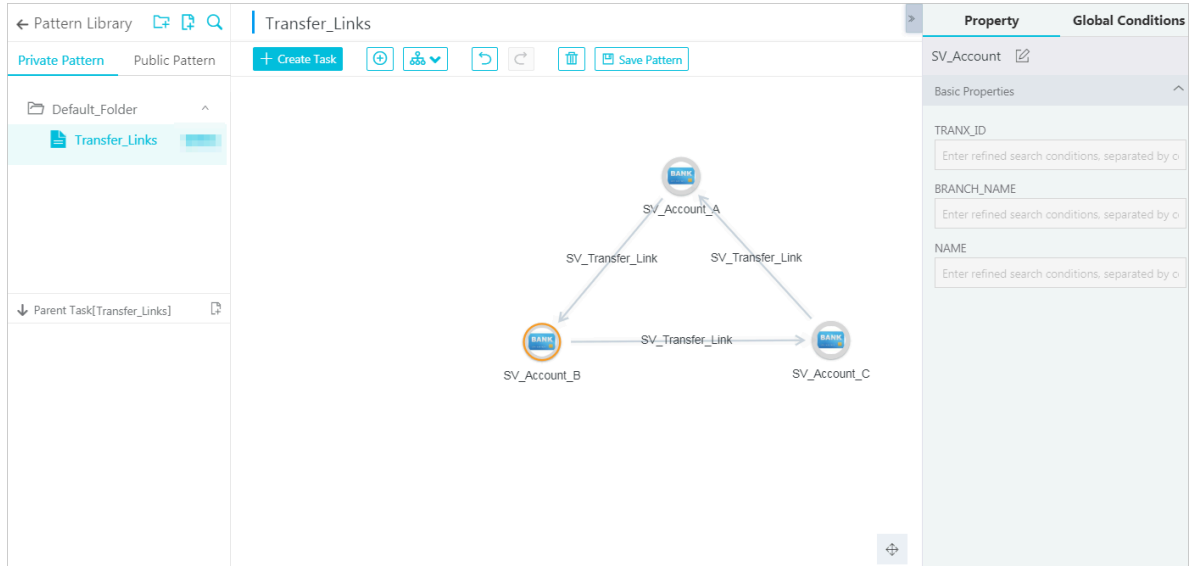
## Context

You can view the following pattern information:


- Relationship graph structures of a pattern
- Tasks created based on a pattern
- Filter conditions of a pattern

## Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, click Private Pattern, and then click a folder to view the private patterns in the folder.  
Click Public Pattern, and then click a folder to view the private patterns in the folder.
4. Click a pattern to view its details.



Operation	Procedure
View the relationship graph structures	Click a pattern to view the relationship graph structures on the right side of the page.
View tasks created based on a pattern	Click a pattern. All tasks created based on this pattern are displayed under the Pattern Library tab.

Operation	Procedure
View the filter conditions of a pattern	In the relationship graph structure, select an object or link, and click the  icon in the upper-right corner to display the Property tab and the Global Conditions tab. You can view the filter conditions of objects or links and the global filter conditions of the pattern.

### 7.14.2.3 Modify patterns

You can modify a pattern when its relationship graph structure and filter conditions are not suitable for the business scenarios. Modifying a pattern does not affect the tasks that are created based on the pattern.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



**Note:**

Only administrators and creators can modify private patterns.

- You have created a pattern. For more information about how to create a pattern, see [Create patterns](#).

#### Context

In Graph Analytics, you can modify the following aspects of a pattern:

- Modify the pattern name
- Modify the filter conditions
- Modify the object node name
- Modify the link type
- Add an object node
- Add a link
- Delete an object node
- Delete links

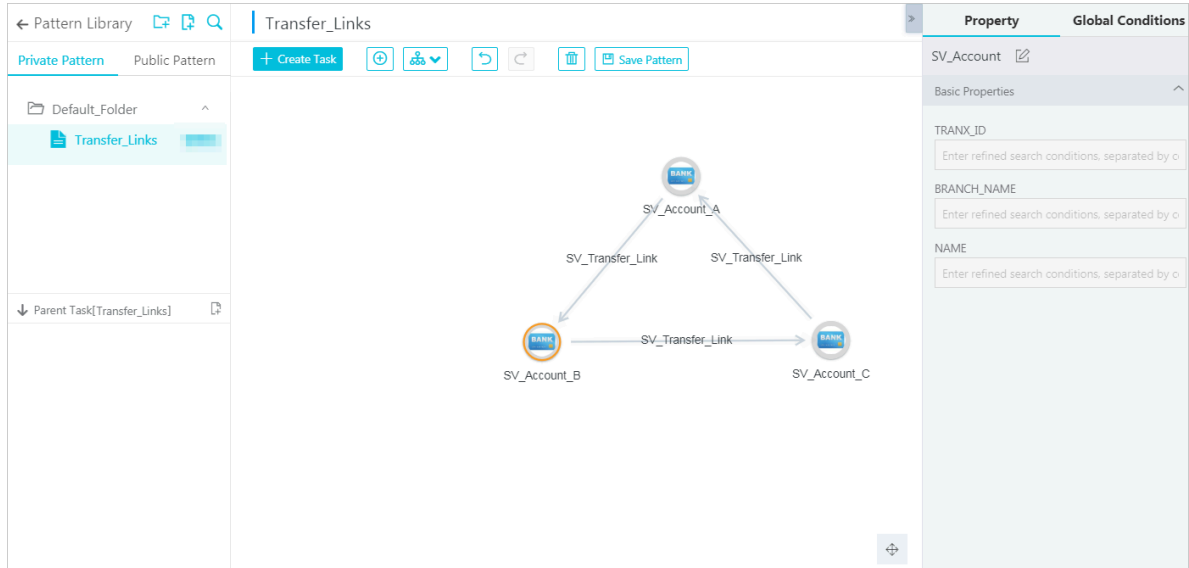


**Note:**


After you modify a pattern, the existing tasks that are created based on this pattern are not affected. The modifications will be applied when you create a new task based on the modified pattern.




## Procedure

1. [Log on to Analytics Workbench.](#)
2. In the top navigation bar, click **Intelligent Network**.
3. In the **Pattern Library** navigation pane, select a pattern in **Private Pattern** or **Public Pattern**.



4. Modify the pattern as needed.

Operation	Procedure
Modify the pattern name	<ol style="list-style-type: none"> <li>In the <b>Pattern Library</b> navigation pane, click <b>Private Pattern</b> or <b>Public Pattern</b> and select a pattern to be modified. Right-click the pattern and select <b>Rename</b>.</li> <li>Reset the name, and then press <b>Enter</b>. A message is displayed, indicating that the operation is successful.</li> </ol>
Modify the filter conditions	Click the  icon in the upper-right corner, and set the filter conditions in <b>Property</b> and <b>Global Conditions</b> . For more information, see <a href="#">Create patterns</a> .
Modify the object node name	<ol style="list-style-type: none"> <li>Right-click the object node to be modified, and select <b>Change Node Name</b>.</li> <li>In the dialog box that appears, set the new name of the object node, and then click <b>OK</b>.</li> </ol>
Modify the link type	<ol style="list-style-type: none"> <li>Right-click the link to be modified, and select <b>Set Link Type</b>.</li> <li>In the dialog box that appears, select the link type, specify the link logic, and then click <b>OK</b>. You can select multiple link types.</li> </ol>

Operation	Procedure
Add an object node	<ol style="list-style-type: none"> <li>Click the  icon in the toolbar, or click Toolbox on the left side of the page.</li> <li>Drag the objects one by one from the toolbox to the right side of the page to add object nodes.</li> <li><i>Add a link to a new object node.</i></li> </ol>
Add a link	<p>After you have deleted a link or added a new object node, you need to add a link for the object node.</p> <ol style="list-style-type: none"> <li>Move your mouse pointer to the source object. When four cross signs appear around the object, move your mouse pointer to a cross sign and press the left mouse button. Then, drag the mouse pointer to the target object. When the four cross signs appear around the target object, release the left mouse button.</li> <li>In the Add Link dialog box that appears, select the link type and specify the link logic.</li> <li>Click OK.</li> </ol>
Delete object nodes or links	<div>  <b>Note:</b> After an object node is deleted, the links related to the object node are also deleted.         </div> <p><b>Method one:</b></p> <ol style="list-style-type: none"> <li>Select an object node or link, and click the  icon in the toolbar. Or, you can press the Delete key. You can click-and-drag the mouse to box select multiple object nodes.</li> <li>In the message box that appears, click Delete Selected.</li> </ol> <p><b>Method two:</b></p> <p>Right-click an object node or link, or any of the object nodes in the box selection, and then select Delete Selected to directly delete the object node or link.</p>

- After you have configured the preceding parameters, click Save Pattern in the toolbar to save the pattern.



## 7.14.2.4 Set private patterns to public patterns

In Graph Analytics, you can set private patterns to public patterns, but this is an irreversible operation.

### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



#### Note:

Only administrators and creators can set private patterns to public patterns.

- You have created a private pattern. For more information about how to create a pattern, see [Create patterns](#).

### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, click Private Pattern, right-click a private pattern, and then select Set as Public Pattern.
4. In the dialog box that appears, configure relevant parameters.

The parameters are described in [Table 7-45: Parameter configurations for setting a private pattern to a public pattern](#).

Table 7-45: Parameter configurations for setting a private pattern to a public pattern

Parameter	Description
Publish the dependent task?	<p>Valid values are as follows:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> Private tasks created based on this private pattern are also set to public tasks.</li><li>• <b>No:</b> Private tasks created based on this private pattern remain private tasks. After this private pattern is set to a public pattern, these tasks will be converted to private tasks that are created based on a public pattern. You can manually set these tasks as public tasks when necessary.</li></ul>

Parameter	Description
Target Directory	<p>Set the target directory in Public Pattern to receive the private pattern.</p> <p>When the private pattern is set as a public pattern, it will be moved from the directory in Private Pattern to the specified directory in Public Pattern.</p>

5. After you have configured the preceding parameters, click OK. A message is displayed, indicating that the operation is successful.

### 7.14.2.5 Delete a pattern

You can delete a pattern that is no longer used. Deleting a pattern does not affect the tasks that are created based on the pattern.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



#### Note:

Only administrators and creators can delete private patterns.

- You have created a pattern. For more information about how to create a pattern, see [Create patterns](#).

#### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, click Private Pattern or Public Pattern and select a pattern to be deleted. Right-click the pattern and select Delete.
4. In the message box that appears, click OK.

### 7.14.3 Tasks

#### 7.14.3.1 Create a task

A task is created based on a pattern. It can be used to query the data with the same graph structure as the task in the data source.


#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.
- You have created a pattern. For more information about how to create a pattern, see [Create patterns](#).

## Context

For a task created based on the pattern, the objects, links, and filter conditions of the task are inherited from the pattern by default, which are exactly the same as the pattern. To modify these configurations, see [Modify a task](#).

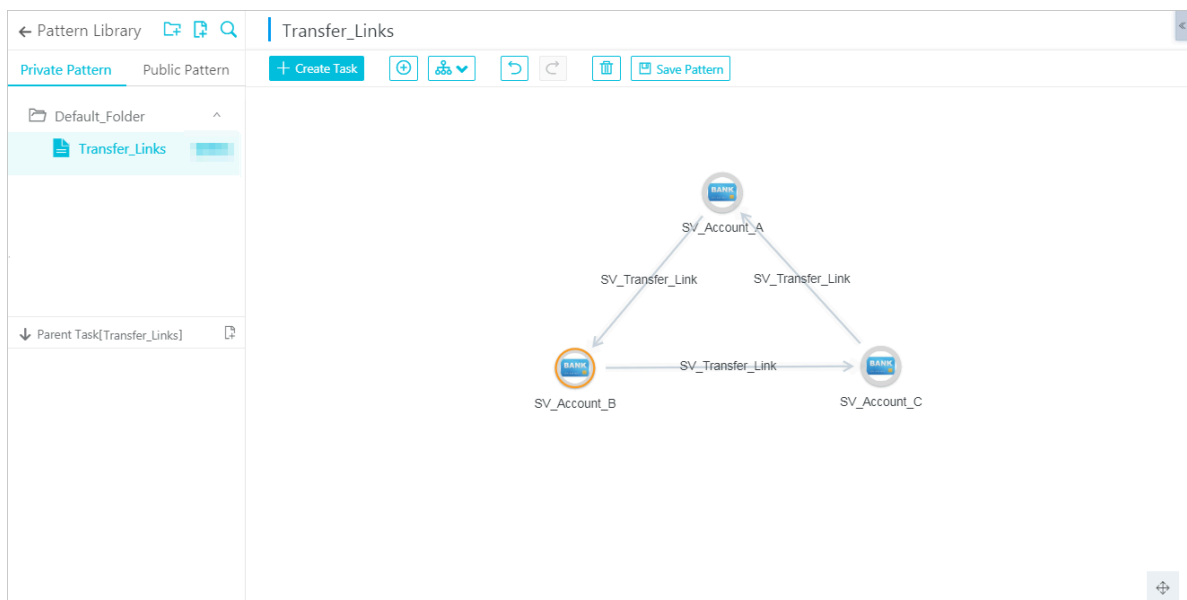
## Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, select a pattern in Private Pattern or Public Pattern. Click Create Task on the right side of the page or click the  icon (Create a Task).



### Note:

**You can only create private tasks based on a private pattern. You can create public tasks or private tasks based on a public pattern.**



**4. Configure the parameters on the Create Task dialog box that appears.**

The parameters are described in [Table 7-46: Parameter configurations for creating a task](#).

**Create Task**

**Task Name**

**Affiliated Pattern** Transfer\_Links

**Task Type** ☐ Public Task ☒ Private Task

For private tasks, only you and the administrator have the permission to edit and delete tasks.

Table 7-46: Parameter configurations for creating a task

Parameter	Description
Task Name	The user-defined task name.
Affiliated Pattern	The pattern that you have selected. It is automatically entered and cannot be modified.
Task Type	When the selected pattern is public, you can set the task type to Public Task or Private Task. When the selected pattern is private, the default task type is Private Task and cannot be modified.

**5. Click OK.**

After you have configured the preceding parameters, a message appears, indicating that the task has been created. The objects, links, and filter conditions

of the task are inherited from the pattern by default, which are exactly the same as the pattern.

### 7.14.3.2 Check the task

You can view tasks based on patterns. You can select a specific pattern and view all the tasks that are created based on the pattern. You can also view the detailed information about these tasks, such as the relationship graph structures and filter conditions of the tasks.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



**Note:**

Only administrators and creators can view the private tasks.

- You have created a task. For more information about how to create a task, see [Create a task](#).

#### Context

Tasks are divided into public tasks and private tasks. A lock icon is displayed before a private task, but no lock icon is displayed before a public task. A task can have three statuses: executed, not executed, and failed. These three statuses are represented by different icons.

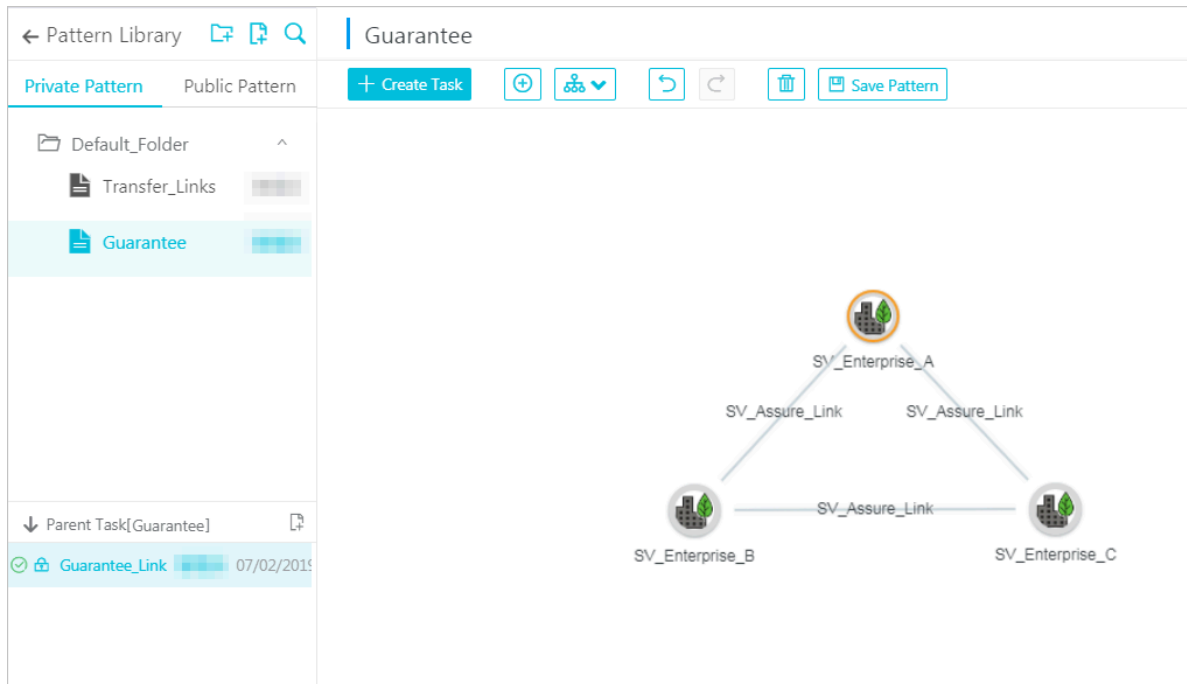
You can view the following task information:

- The task type and task status
- Relationship graph structures of a task
- Filter conditions of a task
- Primary keys of the object in the task

#### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.

**3. In the Pattern Library navigation pane, select a pattern in Private Pattern or Public Pattern and view the tasks that are created based on the pattern.**



**The task types and statuses are described as follows:**

- **Tasks are divided into public tasks and private tasks. A lock icon is displayed before a private task, but no lock icon is displayed before a public task.**
- **A task can have three statuses: executed, not executed, and failed. These three statuses are represented by different icons.**

**4. Click a task and view the detailed information about the task as follows.**

Guarantee\_Link

Run Global Task

SV\_Enterprise\_A

SV\_Assure\_Link

SV\_Enterprise\_B

SV\_Assure\_Link

SV\_Enterprise\_C

SV\_Assure\_Link

SV\_Enterprise

Basic Properties

ID

name

Result


Task Status: Success Operator: admin Duration: 1 s Matching Results: 4 Start Time: 07/02/2019 15:16:35 End Time: 07/02/2019 15:16:36

Guarantee\_Link152\_0

search content

	SV_Enterprise_A	SV_Enterprise_B	SV_Enterprise_C	Actions
<input type="checkbox"/>	27002	27003	27004	Go to Graph   Go to Map
<input type="checkbox"/>	27003	27004	27008	Go to Graph   Go to Map
<input type="checkbox"/>	27005	27006	27007	Go to Graph   Go to Map

Go to Graph Go to Map Selected 0/4

Operation	Procedure
View the relationship graph structures of a task	Click a specific task to view the relationship graph structures on the right side of the page.
View the filter conditions of a task	Select an object or a link, and click the  icon in the upper-right corner. The Property and Global Conditions tabs are displayed. You can view the filter conditions of an object or a link and the global conditions of tasks.
View the primary keys of objects in a task	Right-click an object and select Set Primary Keys to view the primary keys that have been set for this object.

**7.14.3.3 Modify a task**

For a task that is created based on a pattern, the objects, links, and filtering conditions of the task are inherited from the pattern by default, which are exactly the same as the pattern. When some configurations of a task do not match the scenario, you can modify the task accordingly.

**Prerequisites**

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.

**Note:**

Only administrators and creators can modify private tasks.

- You have created a task. For more information about how to create a task, see [Create a task](#).


## Context

In Graph Analytics, you can modify the following aspects of a task:


- Modify the filter conditions
- Modify the object node name
- Modify the link type
- Add an object node
- Add a link
- Delete an object node
- Delete a link



## Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, select a pattern in Private Pattern or Public Pattern. Select the task to be modified.
4. Modify a task as shown in the following operations.

Operation	Procedure
Modify the filter conditions	You can click the  icon in the upper-right corner. You can set the filter conditions for a task in the Property and Global Conditions tabs that appear. For more information, see <a href="#">Create patterns</a> .
Set the object primary key	<ol style="list-style-type: none"><li>a. Right-click the object node to be modified, and select Set Primary Keys.</li><li>b. In the dialog box that appears, set the primary keys of the object node, and click OK. You can set multiple primary keys.</li></ol>



Operation	Procedure
Modify the object node name	<ol style="list-style-type: none"> <li>Right-click the object node to be modified, and select <b>Change Node Name</b>.</li> <li>In the dialog box that appears, set the new name of the object node, and then click <b>OK</b>.</li> </ol>
Modify the link type	<ol style="list-style-type: none"> <li>Right-click the link to be modified, and select <b>Set Link Type</b>.</li> <li>In the dialog box that appears, select the link type, specify the link logic, and then click <b>OK</b>. You can select multiple link types.</li> </ol>
Add an object node	<ol style="list-style-type: none"> <li>Click the  icon in the toolbar, or click <b>Toolbox</b> on the left side of the page.</li> <li>Drag the objects one by one from the toolbox to the right side of the page to add object nodes.</li> <li><i><a href="#">Add a link to a new object node.</a></i></li> </ol>
Add a link	<p>After you have deleted a link or added a new object node, you need to add a link for the object node.</p> <ol style="list-style-type: none"> <li>Move your mouse pointer to the source object. When four cross signs appear around the object, move your mouse pointer to a cross sign and press the left mouse button. Then, drag the mouse pointer to the target object. When the four cross signs appear around the target object, release the left mouse button.</li> <li>In the <b>Add Link</b> dialog box that appears, select the link type and specify the link logic.</li> <li>Click <b>OK</b>.</li> </ol>

Operation	Procedure
Delete object nodes or links	<div> <b>Note:</b> After an object node is deleted, the links related to the object node are also deleted.</div> <p><b>Method one:</b></p> <ol style="list-style-type: none"><li>Select an object node or link, and click the  icon in the toolbar. Or, you can press the Delete key. You can click-and-drag the mouse to box select multiple object nodes.</li><li>In the message box that appears, click Delete Selected.</li></ol> <p><b>Method two:</b></p> <p>Right-click an object node or link, or any of the object nodes in the box selection, and then select Delete Selected to directly delete the object node or link.</p>

5. Click Run Global Task to save the modification configuration.



**Note:**

After a task is modified, you must execute the task again. Otherwise, the modifications cannot be saved.

#### 7.14.3.4 Execute the task and view the result

After the task is created or modified, you can execute the task to query sub-graphs that have the same graph structure as the task. The task execution results are displayed in a list. You can also view the execution results in the graph area.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.




**Note:**

Only administrators and creators can execute private patterns.






- You have created a task. For more information about how to create or modify a task, see [Create a task](#) and [Modify a task](#).

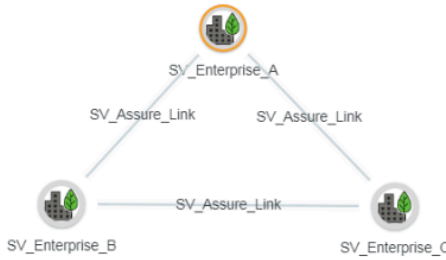
#### Procedure

- [Log on to Analytics Workbench](#).

2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, select a pattern in Private Pattern or Public Pattern. Select the task to be executed.
4. Click Run Global Task in the top toolbar.
5. In the right-side area, click the  icon to view the execution results.

Guarantee\_Link

Run Global Task     



Result

Task Status: **Success** Operator: admin Duration: 1 s Matching Results: 4 Start Time: 07/02/2019 15:16:35 End Time: 07/02/2019 15:16:36

Guarantee\_Link152\_0

	SV_Enterprise_A	SV_Enterprise_B	SV_Enterprise_C	Actions
<input type="checkbox"/>	27002	27003	27004	<a href="#">Go to Graph</a>   <a href="#">Go to Map</a>
<input type="checkbox"/>	27003	27004	27008	<a href="#">Go to Graph</a>   <a href="#">Go to Map</a>
<input type="checkbox"/>	27005	27006	27007	<a href="#">Go to Graph</a>   <a href="#">Go to Map</a>

[Go to Graph](#) [Go to Map](#) Selected 0/4 1

6. View the task execution results in the graph area.



**Note:**

The task result is displayed as a temporary analysis file in the graph area. You can save and share the file and perform other operations. You can also extend the analysis.

Operation	Procedure
View the results of a single task in the graph area	Select a record and click Go to Graph. The result is displayed on the graph page, as shown in <a href="#">Figure 7-69: View the results of a single task in the graph area.</a>

Operation	Procedure
View the results of multiple tasks in the graph area	You can also select multiple records and then click Go to Graph. The results are displayed on the graph page, as shown in <a href="#">Figure 7-70: View the results of multiple tasks in the graph area</a> .

Figure 7-69: View the results of a single task in the graph area

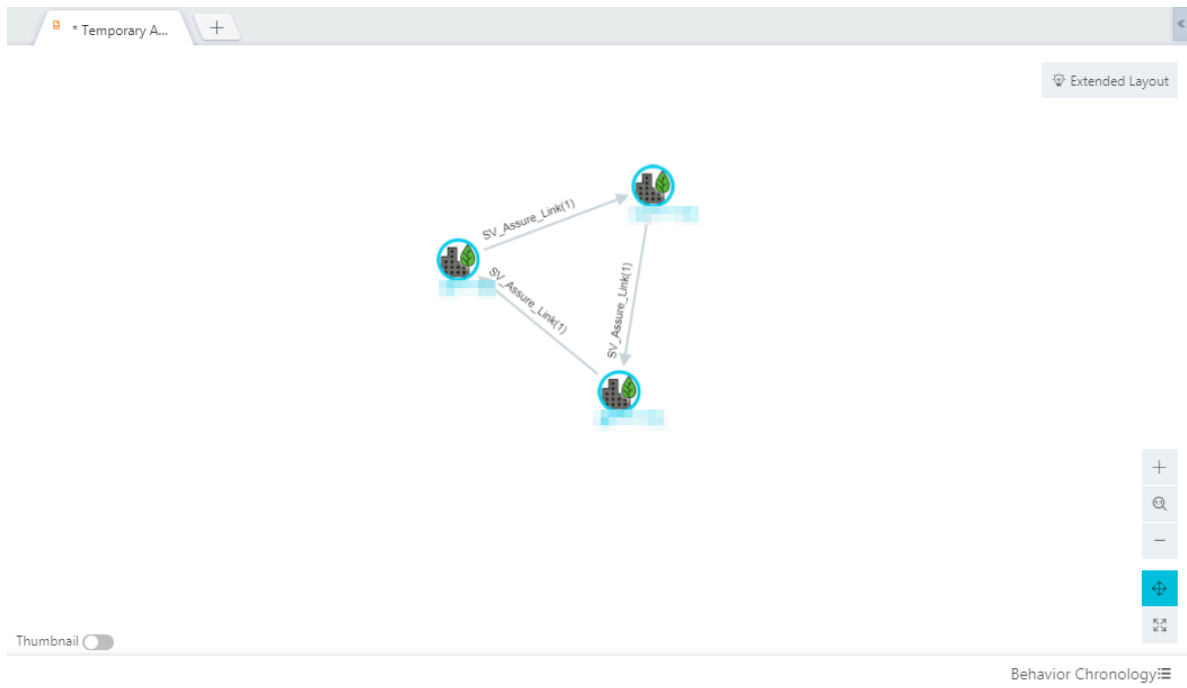
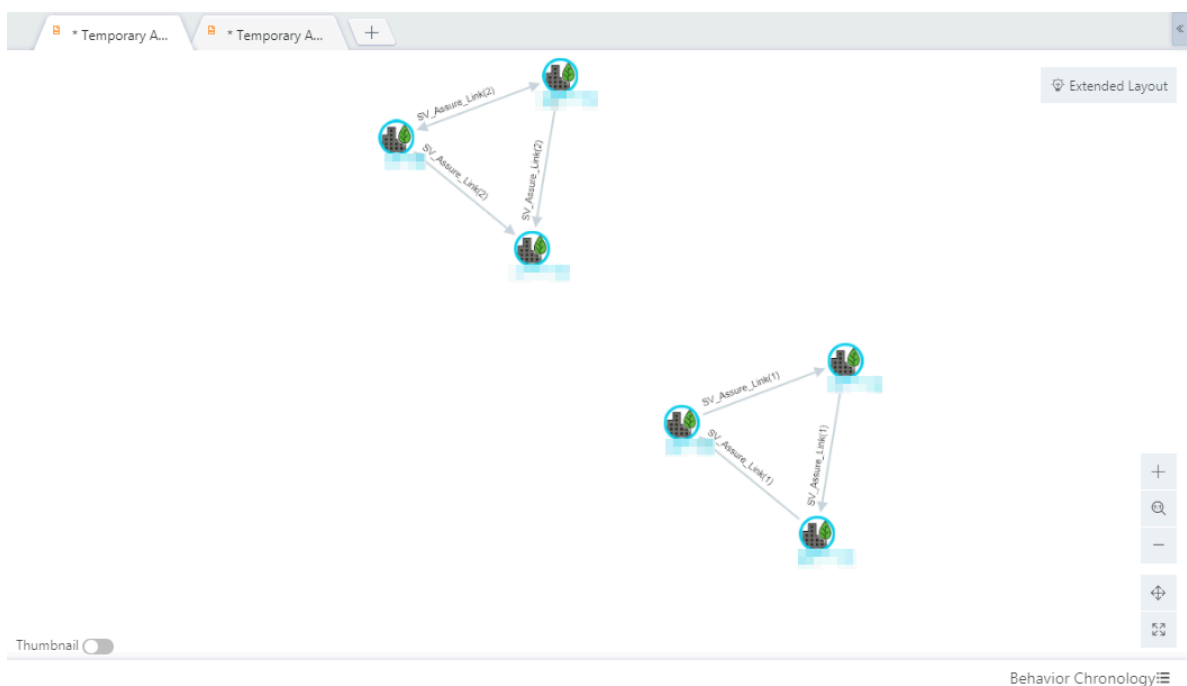


Figure 7-70: View the results of multiple tasks in the graph area



### 7.14.3.5 Set a private task as a public task

Private tasks created based on public patterns can be set to public tasks, but this is an irreversible operation.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



**Note:**

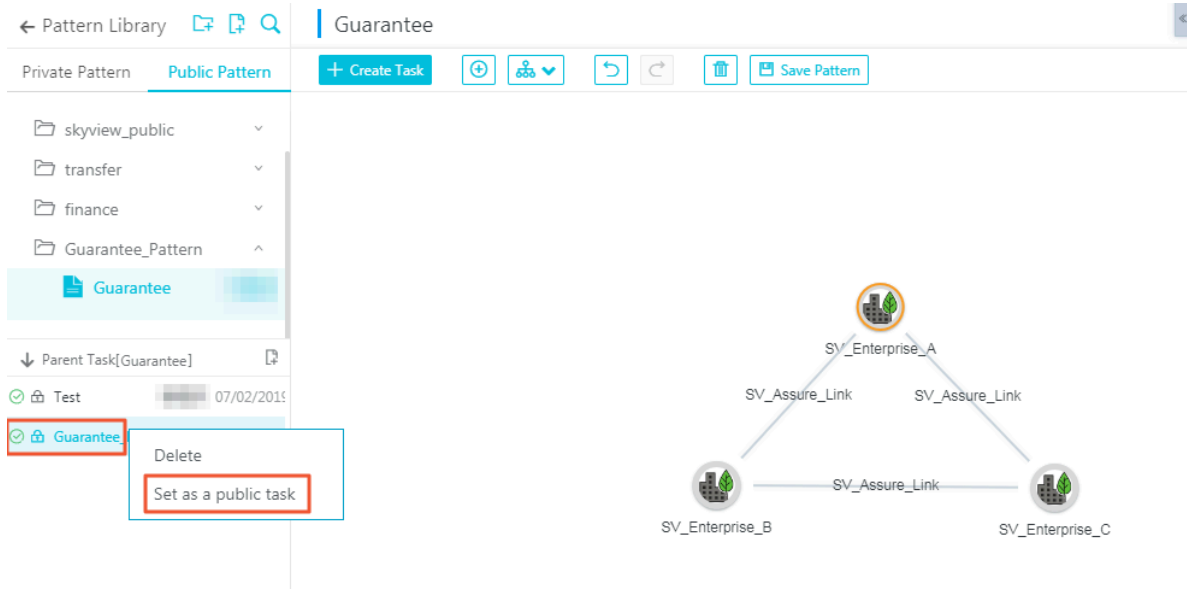
Only administrators and creators can set private patterns to public patterns.

- You have created a private task based on the public pattern. For more information about how to create a pattern, see [Create a task](#).

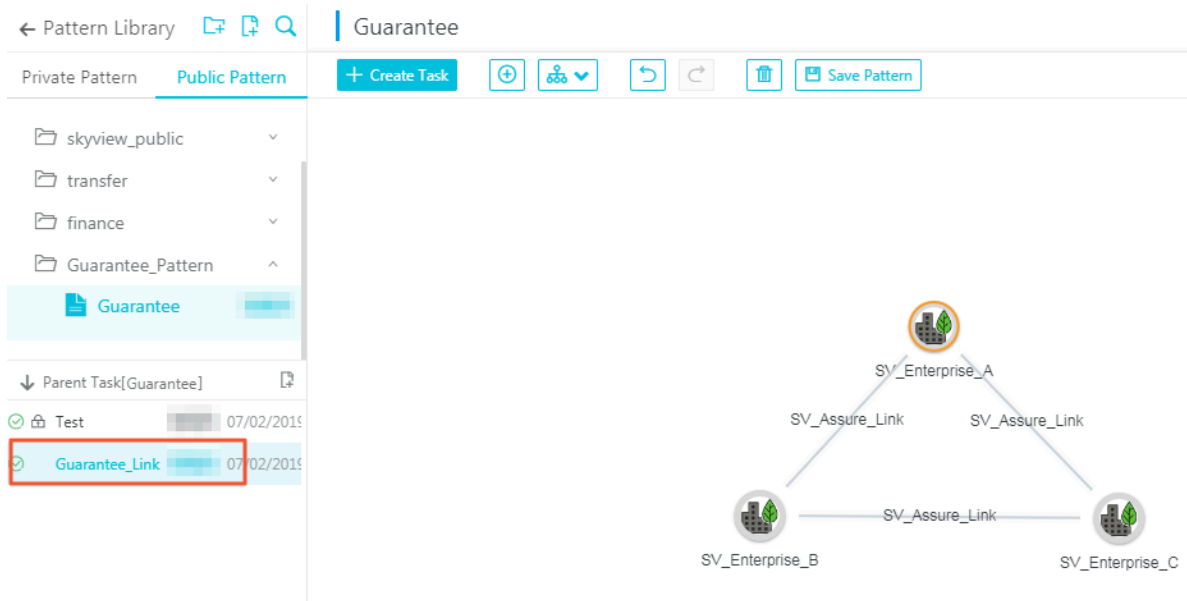
#### Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.

3. In the Pattern Library navigation pane, click **Public Pattern** . Right-click a private task that is created based on the pattern, and then select **Set as Public Task**.



After a private task is set to a public task, the lock icon in front of the task disappears. A message is displayed, indicating that the operation is successful.



### 7.14.3.6 Delete a task

You can delete a task when it is no longer needed.

#### Prerequisites

- Make sure that you have an account and password for Graph Analytics and are authorized to operate on Analytics Workbench and Administration Console.



**Note:**

Only administrators and creators can delete private patterns.

- You have created a task. For more information about how to create a pattern, see [Create a task](#).

## Procedure

1. [Log on to Analytics Workbench](#).
2. In the top navigation bar, click Intelligent Network.
3. In the Pattern Library navigation pane, select a pattern in Private Pattern or Public Pattern. Right-click the task to be deleted and then click Delete.



**Note:**

Deleted tasks cannot be restored. Exercise caution when you delete a task.

4. In the dialog box that appears, click OK.

## 7.15 Examples

### 7.15.1 Tax industry case studies

Tax inspectors used Graph Analytics to analyze abnormal information and uncovered a tax cheat group that was related to 13 companies.

#### Case background

On April 1, 2016, a clerk of an overseas trading platform was routinely checking goods being loaded and unloaded. When the clerk checked “Hangzhou Children's Products Co., Ltd.”, he was told that the goods had been shipped. The company exported goods such as children's products to overseas countries with a total volume of 40 million RMB. Apparently, this was a case of avoiding "loading inspection", and this case attracted the clerk's attention. The clerk used Graph Analytics to analyze the company and found a large-scale tax cheat group related to this company.

#### Data preparation



**Note:**

All the data in this case, including individual people's names, company names, places, and times, is purely fictitious.

Data resources used in this case include:

- Information of the drawer (the manufacturer), the customer (the domestic trader), the remitter, and the overseas trader.
- The relationships between the customer and the drawer, the customer and the remitter, and the customer and the overseas trader are required.

Connect to data sources and configure an OLEP model

Before you perform an analysis, you must connect the data source to Graph Analytics and build an OLEP model based on the data table in the data source.

1. [Log on to Administration Console of Graph Analytics](#). For more information about how to connect the data source to Graph Analytics, see [Create data sources](#).
2. Create objects based on the data table in this case, as shown in [Create OLEP models for tables](#).

The created objects are as follows.

Table 7-47: Objects mapped by the data table

Data table	Object
Drawer information table	<ul style="list-style-type: none"><li>• Drawer: Maps the drawer information.</li><li>• WIFI: Maps the WIFI used by the drawer.</li></ul>
Customer information table	<ul style="list-style-type: none"><li>• Contact phone number: Maps the telephone number of the customer.</li><li>• Customer: Maps the customer information.</li></ul>
Remitter information table	Remitter: Maps the remitter information.
Overseas trader information table	<ul style="list-style-type: none"><li>• Country: Maps the country where the overseas trader is located.</li><li>• Overseas trader: Maps the overseas trader information.</li></ul>



**3. Create a first-degree link mapping for the data table in this case, as shown in**

*Create OLEP models for tables.*

**The first-degree link is created as follows.**

Table 7-48: Links mapped by the data table

Data table	First-degree link	Source object	Target object
<b>Drawer information table</b>	<ul style="list-style-type: none"><li>• <b>Frequent logon WIFI: The link between the drawer and the frequent logon WIFI can be used to query both the drawer and the WIFI information.</b></li><li>• <b>Name of the drawer's company: The link between the drawer and the drawer's company can be used to query both the drawer and the company name.</b></li><li>• <b>Drawer's phone number: The link between the drawer and the drawer's phone number can be used to query both the drawer and the phone number of the drawer.</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Frequent logon WIFI: The drawer</b></li><li>• <b>Name of the drawer's company: The drawer</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Frequent logon WIFI: The WIFI</b></li><li>• <b>Name of the drawer's company: The name of the company</b></li></ul>

Data table	First-degree link	Source object	Target object
<b>Customer information table</b>	<ul style="list-style-type: none"> <li>Customer's phone number: The link between the customer and the customer's phone number can be used to query both the telephone number or the user who have used this phone number.</li> <li>Name of the customer's company: The link between the customer and the customer's company can be used to query both the customer or the company of the customer.</li> </ul>	<ul style="list-style-type: none"> <li>Customer's phone number: The customer</li> <li>Name of the customer's company: The customer</li> </ul>	<ul style="list-style-type: none"> <li>Customer's phone number : The phone number</li> <li>Name of the customer's company: The name of the company</li> </ul>
<b>Customer -drawer information table</b>	<b>Invoicing:</b> This link can be used to query both the customer and the drawer.	<b>Drawer</b>	<b>Customer</b>
<b>Customer -remitter information table</b>	<b>Remittance:</b> This link can be used to query both the customer and the remitter.	<b>Remitter</b>	<b>Customer</b>
<b>Customer -overseas trader information table</b>	<b>Purchase:</b> This link can be used to query both the overseas trader and the customer.	<b>Overseas trader</b>	<b>Customer</b>

Data table	First-degree link	Source object	Target object
Overseas trader information table	Country: This link can be used to query both the overseas trader and the country where the overseas trader is located.	Overseas trader	Country

4. For more information about how to configure the business information related to these objects and links, see [Configure object properties and business parameters](#) and [Configure link properties and business parameters](#).
5. Create multi-degree links as shown in [Create a multi-degree link](#).

The multi-degree links created in this case are as follows.

Table 7-49: Multi-degree links

Multi-degree link	Base links	Description
The drawer and the customer with the same name.	<ul style="list-style-type: none"><li>• Name of the customer's company.</li><li>• Name of the drawer's company.</li></ul>	Queries the drawer with the same name as the customer.

#### Analysis process

After you have connected the data source and configured the OLEP model, you can go to Analytics Workbench to analyze cases.

1. [Log on to Analytics Workbench](#).
2. On the Graph page, create a new analysis, and add Hangzhou Children's Products Co., Ltd as the node to start with.
3. Select the added node, and query the company's information on the right side of the page.

**Results:** This company is a manufacturer company instead of a trading company.

**Inferences made by analysts:** This company needs a domestic trader to export its goods.

4. Analysts used the link extension feature of Graph Analytics to query the downstream customer of this company, namely the domestic trader.

Select this company as a node, and click Link Extension in the toolbar. Set Link Type to Invoicing to query the downstream customer of this company.

**Results:** This company has been the drawer for three trading companies. These three companies are from the medical industry and the apparel industry, but somehow have become the drawee of a manufacturer engaged in children's products.

5. Analysts used the link extension feature to investigate these three customers to find the overseas traders that have built business relationships with these three customers.

Select these three customers, and click Link Extension. Set Link Type to Purchase to query the overseas traders that have built business relationships with these three customers.

**Results:** All these customers have their own overseas trader.

6. Analysts used the link extension feature to search the remitter of these three customers.

Select these three customers, and click Link Extension. Set Link Type to Remitter, and search for the remitters related to these three customers.

**Results:** The analysis results indicate that these three customers receive remittance from the same remitter. In this phase, the relationship network is downward-trend and looks like a symmetrical funnel, which is a typical abnormal pattern.

**Inferences made by analysts:** A manufacturer sells products to overseas buyers through three customers, but there is only one remitter. Whether these three overseas traders are in the same area or not, this network pattern is very suspicious.

**7. Analysts tried to query the locations of these three overseas traders.**

Select these overseas traders, and click Link Extension. Set Link Type to Country to query the locations of these three overseas traders.

**Results:** These three overseas traders are located in different countries.

**Inferences made by analysts:** These overseas traders are located in different countries, but the remittance is made by the same remitter. Given that, it is very likely that these three customers and the drawer have committed tax cheating.

**8. To obtain more information, analysts viewed the information cubes of these three customers.**

**Results:** Two of the customers are companies that provide both manufacturing and trading services.

**9. Following these clues, analysts queried the manufacturers of these two companies.**

Select these two companies and click Link Extension. Set Link Type to Drawer and customer with the same name to query the manufacturers of these two companies.

**Results:** These two companies share the same manufacturer (drawer).

**10 Analysts used Group Analysis to analyze the relationship between these two companies.**

Select these two nodes and their manufacturers, and choose Analyze > Group Analysis. Set Link Type to Invoicing to analyze the relationship between them.

**Results:** These two companies issue invoices to each other.

**Judgement made by the analysts:** Based on experience, this phenomenon is obviously abnormal.

**11 Analysts used Backbone Analysis to verify whether this network contains any key members.**

In the toolbar, choose Analyze > Backbone Analysis. Set the Backbone Node to Customer to analyze the backbone customers in the current network.

**Results:** The current relationship network contains two companies as key nodes. These two companies have taken up important positions in the current relationship network.

**12. Analysts used the link extension feature to search for the drawers of these two key members.**

**Select these two key members, and click Link Extension. Set Link Type to Invoicing to query the drawers of these two key members.**

**Results: Analysts found a group of drawers by analyzing one of the key members.**

**Inferences made by analysts: This company provides both manufacturing and trading services. It provides trading services for domestic manufacturers from various industries, including the electromechanics industry, electronics industry, moulding industry, and the food industry. This is a suspicious issue of tax cheating.**

**13. Analysts checked the behavior details of these drawers and the key member.**

**Select this key member and the group of drawers. Click Behavior Chronology, and then click the Details tab. Set Link to Invoicing to view the invoicing details.**

**Results: The proportion of sales and export volumes are both around 10%, which is relatively even.**

**Inferences made by analysts: The proportion of orders and exports are too even, which shows obvious human manipulation.**

**14. Analysts used Common Neighbors to check whether these drawers have shared objects, for example, WIFI.**

**Select the drawer nodes, and choose Analyze > Common Neighbors. Set Link Type to WIFI to check whether the drawers have a shared WIFI.**

**Results: These drawers often connect to WIFI using the same MAC address to log on to the trading platform.**

**Inferences made by analysts: This is an obvious abnormal phenomenon. It is very likely that these companies are operated by the same group of people.**

**15. Analysts used Common Neighbors to check whether the drawers and the customers have shared phone numbers.**

**Select all drawers and customer nodes, and choose Analyze > Common Neighbors. Set Link Type to Drawer TEL and Customer TEL to check whether the drawer and the customer have a shared contact number.**

**Results: A telephone number was shared by different drawers and customers.**

**Inferences made by analysts: Based on the abnormal information found in the relationship network, it is very likely that these groups are cheating on tax rebates.**

On-the-spot investigation

**According to the results returned from the on-the-spot investigation, all these manufacturers and traders are highly suspicious. The department concerned continued to investigate these groups, and uncovered a tax cheat group related with 13 enterprises. This group registered different roles on the trading platform, made fake invoices and transactions internally, and the amount of tax refunds reported to the trade platform reached 100 million RMB.**

## 7.16 FAQ

**1. Q: How can I log on to the system for the first time?**

**A: Contact the administrator to obtain the account and the initial password to log on to Graph Analytics. To keep your account secure, modify the password as prompted.**

**2. Q: Why does the system prompt an error message indicating incorrect user name and incorrect password when I was logging on to Graph Analytics?**

**A: If you are unable to log on to the system, check whether the account name and the password you entered are correct, and whether the password is entered in half-width characters, in the correct case, or has any space. If the error persists, contact the administrator.**

**3. Q: What is the default rule for the simple link extension (double-clicking a link to start a link extension)?**

**A: To help you analyze the information quickly, the administrator configures common links in Administration Console and synchronizes them to the double-**

click operation. These common links must be published by the administrator. If you need to add or remove a double-click link extension, contact the administrator.

4. Q: Why does the system prompt "a maximum of five nodes can be selected" when I select all nodes in the graph area to perform the link lookup analysis?

A: By default, Graph Analytics supports a maximum of five nodes in the link extension analysis. If you need to analyze more than five nodes, contact the administrator, and the administrator will assess the system scale and make adjustments.

5. Q: Why can't I delete a node by pressing `Delete` on the keyboard?

A: Currently, you cannot delete a node by pressing `Delete` on the keyboard. You can right-click the selected node and click `Delete` in the shortcut menu.

6. Q: How many steps can be undone or rolled back?

A: A maximum of 20 steps can be rolled back or undone.

7. Q: Why does the `Path Analysis` button turn gray after I select a node?

A: `Path Analysis` is available only after two nodes are selected.

8. Q: After I select a node, why do the `Group Analysis` button and the `Common Neighbors` button turn gray?

A: `Group Analysis` and `Common Neighbors` are available only after two or more nodes are selected.

9. Q: Can I configure a new link analysis method discovered by myself while using Graph Analytics?

A: Currently, links are configured by the administrator. If you need to add a new link analysis method, contact the administrator.

- 10.Q: Why is there no data available after I click `Behavior Analysis`, `Chronology Analysis`, and `Details`?

Q: You need to select a node in the graph area before performing these analyses or viewing behavior details.

- 11.Q: Why are statistics displayed on the right side when no node in the graph area is selected?

A: If no node is selected, statistics on all nodes are collected by default.



**12.Q: How to find the nodes that fall out of the scope of the canvas?**

**A: You can enable the thumbnail to locate the nodes, and move the nodes that fall out of the canvas to the visible area.**

**13.Q: Why are all layout buttons become gray and unavailable?**

**A: The layout buttons become available only after you select a node in the graph area.**

**14.Q: After I select all the nodes in the graph area, why does the system give no response after I click Hierarchical Layout?**

**A: In the Hierarchical Layout mode, you need to select one node as the starting point to view the hierarchical layout.**

**15.Q: Besides the drag button in the toolbar, can I use any shortcut operations to move the canvas?**

**A: You can press Space on the keyboard to move the canvas.**

## 8 Machine Learning Platform for AI

---

### 8.1 What is machine learning?

Machine learning is a process of using statistical algorithms to learn large amounts of historical data and generate an empirical model to provide business strategies.

Apsara Stack Machine Learning Platform for AI is a set of data mining, modeling, and prediction tools. It is developed based on MaxCompute (also known as ODPS).

Machine Learning Platform for AI supports the following functions:

- Provides an all-in-one algorithm service covering algorithm development, sharing, model training, deployment, and monitoring.
- Allows you to complete the entire procedure of an experiment either through the GUI or by running PAI commands. This function is typically intended for data mining personnel, analysts, algorithm developers, and data explorers.
- In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. Machine Learning Platform for AI allows you to call algorithms to decouple the applications and compute engines after you have deployed algorithm packages in MaxCompute clusters.
- Provides various algorithms and reliable technical support, providing more options to resolve service issues. In the Data Technology (DT) era, you can use Machine Learning Platform for AI to implement data-driven services.

Machine Learning Platform for AI can be applied in the following scenarios:

- Marketing: commodity recommendations, user profiling, and precise advertising
- Finance: loan delivery prediction, financial risk control, stock trend prediction, and gold price prediction.
- Social network sites (SNSs): microblog leader analysis and social relationship chain analysis.
- Text: news classification, keyword extraction, text summarization, and text analysis.
- Unstructured data processing: image classification and image text extraction through OCR.

- Other prediction cases: rainfall forecast and football match result prediction.

Machine learning can be divided into three types:

- **Supervised learning:** Each sample has an expected value. You can create a model and map input feature vectors to target values. Typical examples of this learning mode include regression and classification.
- **Unsupervised learning:** No samples have a target value. This learning mode is used to discover potential regular patterns from data. Typical examples of this learning mode include simple clustering.
- **Reinforcement learning:** This learning mode is complex. A system constantly interacts with the external environment to obtain external feedback and determines its own behavior to achieve a long-term optimization of targets. Typical examples of this learning mode include AlphaGo and driverless vehicles.

## 8.2 Quick start

### 8.2.1 Overview

This topic describes how to perform data preparation, data preprocessing, data visualization, algorithm modeling, model prediction and evaluation, online prediction, and DataWorks task scheduling to set up a machine learning experiment.

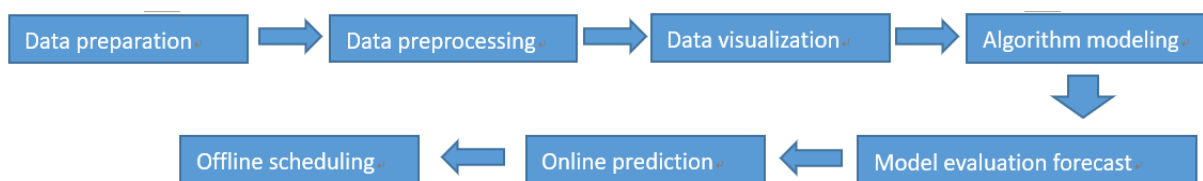


#### Note:

This document covers Apsara Stack Machine Learning Platform for AI, online model service, and deep learning framework. The online model service and deep learning framework are not basic functions of Apsara Stack Machine Learning Platform for AI and must be purchased separately.

For more information, see [Figure 8-1: Machine learning experiment creation flowchart](#).

Figure 8-1: Machine learning experiment creation flowchart



1. *Data preparation*

**Import target data into the Apsara Stack Machine Learning Platform for AI console.**

2. *Data preprocessing*

**Perform data processing, including SQL-based conversion, normalization, and standardization, to ensure that all data has the same dimensions.**

3. *Data visualization*

**Display data in charts to view the features of the data and the distribution of the values. This serves as the basis for model algorithm selection.**

4. *Algorithm modeling*

**Use machine learning algorithms to train data and ultimately build a model.**

5. *Model evaluation forecast*

**Make predictions from and evaluate the model, and use the prediction results to create business development strategies.**

6. *Online prediction*

**Use online prediction to deploy the generated model and adjust your business strategy based on the prediction results.**

7. *Offline scheduling*

**Deploy experiments in DataStudio and run them on a regular basis.**

## 8.2.2 Log on to the Apsara Stack Machine Learning Platform for AI console

**This topic describes how to log on to the Apsara Stack Machine Learning Platform for AI console.**

### Procedure

- In the left-side navigation pane, choose Big Data > Machine Learning.**
- On the page that appears, set Department and click PAI to go to the Apsara Stack Machine Learning Platform for AI console.**



**Note:**

**If this is the first time that you log on to the Apsara Stack Machine Learning Platform for AI console, you must perform the following steps:**

- a. Create a department.
- b. Create a project. Set Department to the department created in the previous step.
- c. You can assign permissions to users, as described in [add a custom role](#).
- d. Create a user. Assign one or more roles to users. The roles must be authorized to access MaxCompute. Set Department to the department created in step 1.
- e. Access the Dashboard page, and choose Big Data > MaxCompute in the left-side navigation pane. On the page that appears, create a MaxCompute task account and a MaxCompute project.
- A. Create an Apsara Stack account: Set Department to the department created in step 1.
- B. Create a MaxCompute project: Set Department to the department created in step 1, Dt\_project to the project created in step 2, and MaxCompute Task Account to the MaxCompute account you created.
- f. Create a DataWorks workspace. In the Advanced Settings section, set MaxCompute Project Name to the project created in step 1.

### 8.2.3 Data preparation

This topic describes how to import data into the Apsara Stack Machine Learning Platform for AI console for modelling.

#### Prerequisites

Make sure that you have created a MaxCompute project and imported table data into the project. You can download the data from <http://archive.ics.uci.edu/ml/datasets/Heart+Disease>.

#### Procedure

1. [Log on to the Apsara Stack Machine Learning Platform for AI console](#) and click Experiments in the left-side navigation pane.
2. On the Experiments page, right-click My Experiments and choose New Experiment from the shortcut menu. In the dialog box that appears, enter the experiment name and description. Click Create to go to the Components page.
3. In the Components list, click Data Source/Target, and drag and drop the Read MaxCompute Table onto the canvas.

4. Click the Read MaxCompute Table component and configure its parameters.  
Enter the MaxCompute table name in Table Name in the right-side configuration pane.
5. In the right-side parameter setting pane, click Column Information to view the column name, data type, and the first 100 rows of data in the input table.

## 8.2.4 Data preprocessing

This topic describes how to perform data preprocessing by using methods such as normalization, SQL scripts, and data splitting.

### Prerequisites

Before data preprocessing, make sure that you have completed [data preparation](#).

### Procedure

1. [Log on to the Apsara Stack Machine Learning Platform for AI console](#) and click Components in the left-side navigation pane.
2. In the Components list, click Tools. Drag and drop the SQL Script component onto the canvas. Click Data Preprocessing. Drag and drop the Normalization component onto the canvas, and connect the components.
3. Click the SQL Script component and click the right-side Parameters tab. In the SQL Script input box, enter the following SQL scripts to convert the features from string type to numeric type.

```
select age,
(case sex when 'male' then 1 else 0 end) as sex,
(case cp when 'angina' then 0 when 'notang' then 1 else 2 end) as
cp,
trestbps,
chol,
(case fbs when 'true' then 1 else 0 end) as fbs,
(case restecg when 'norm' then 0 when 'abn' then 1 else 2 end) as
restecg,
thalach,
(case exang when 'true' then 1 else 0 end) as exang,
oldpeak,
(case slop when 'up' then 0 when 'flat' then 1 else 2 end) as slop,
ca,
(case thal when 'norm' then 0 when 'fix' then 1 else 2 end) as thal
,
(case status when 'sick' then 1 else 0 end) as ifHealth
from ${t2};
```

4. Click the Normalization component and select all fields to normalize the numeric features to values ranging from 0 to 1.

5. Click Data Preprocessing. Drag and drop the Split component onto the canvas and set Split Ratio to 0.7.



**Note:**

This step splits data into two parts: 70% of the data is used as the model training set, and 30% of the data is used as the model prediction set.

## 8.2.5 Data visualization

This topic describes how to view the features and value distribution by using statistical analysis components.

### Prerequisites

Before data visualization, make sure that you have completed [data preprocessing](#).

### Procedure

1. [Log on to the Apsara Stack Machine Learning Platform for AI console](#) and click Components in the left-side navigation pane.
2. In the Components list, click Statistical Analysis. Drag and drop the Whole Table Statistics component onto the canvas. Connect the components, and click Run at the bottom of the canvas.
3. After the experiment stops running, right-click Whole Table Statistics and choose View Data from the shortcut menu. The analysis report is displayed.

## 8.2.6 Algorithm modeling

This topic describes how to perform feature training and generate models by using the machine learning components.

### Prerequisites

Before algorithm modeling, ensure that you have completed [data preprocessing](#) and learned the data characteristics and value distribution through [data visualization](#).

### Procedure

1. [Log on to the Apsara Stack Machine Learning Platform for AI console](#) and click Components in the left-side navigation pane.
2. In the Components list, choose Machine Learning > Binary Classification. Drag and drop the Binary Logistic Regression component onto the canvas, and connect the corresponding components and data streams.

3. Click the component, and select 13 feature columns from Training Feature Columns in the right-side Column Settings pane. All parameters use the default settings.
4. Click Run.
5. Click Models in the left-side navigation pane to view the generated model.

## 8.2.7 Model prediction evaluation

This topic describes how to use a model to make predictions and evaluate its results by using the prediction and evaluation components.

### Prerequisites

Before evaluating the prediction, make sure that you have completed [algorithm modeling](#) and generated a machine learning model from the experiment.

### Procedure

1. [Log on to the Apsara Stack Machine Learning Platform for AI console](#) and click Components in the left-side navigation pane.
2. In the Components list, click Machine Learning. Drag and drop the Prediction component onto the canvas, and connect the corresponding components and data streams.
3. Choose Machine Learning > Evaluation. Drag and drop the Binary Classification Evaluation component onto the canvas and connect the corresponding components and data streams.
4. Click Run in the upper-left corner of the canvas.  
  
During the running process, select a component and click the Developer Tool icon in the lower-right corner of the canvas to view the running status of the component.
5. Right-click the Binary Classification Evaluation component and choose View Evaluation Report from the shortcut menu to generate the ROC curve of the LR model trained with different parameters.



## 8.2.8 Online model service (must be activated separately)

### 8.2.8.1 Deploy an online model service

This topic describes how to deploy the generated experiment model through the online model service. You can adjust your business strategy anytime based on predicted results.

#### Prerequisites

Before deploying the online model service, make sure that the preceding steps are completed and the components are running properly. A green check means that the component is running correctly.

#### Procedure

1. [Log on to the Apsara Stack Machine Learning for AI console](#) and click **Experiments** in the left-side navigation pane.
2. Click the **My Experiments** tab and select an experiment to navigate to the canvas.



#### Notice:

Make sure that the selected experiment is running properly. A green check means that the component is running correctly.

3. In the upper-left corner of the canvas, choose **Deploy** > **Online Model Service**.
4. Select the model to deploy and click **Next**.
5. Select a deployment mode. You can select one of the following modes:
  - [New Service](#)
  - [Add Existing Service Version](#)
  - [Create Blue-green Deployment](#)

### 8.2.8.2 Create a service

This topic describes how to use the **New Service** mode to deploy online prediction services.

#### Procedure

1. Complete [Preparations for online model prediction](#).
2. Set Processes and Quota.



#### Note:

**Processes determines the maximum number of concurrently running programs.  
Quota determines the running speed and the parameters such as RT and QPS.**

3. Click Deploy.

It takes several minutes to create the model.

4. After the model is created, click the model name to view information about the model invocation.

5. Click the icons under Monitor to view statistics about QPS, response, RT, traffic, CPU utilization, memory usage, and daily invocation.

6.



**Note:**

**Perform this step when resources are insufficient and need to be expanded.**

Click Update to expand resources.

7. Click Online Debugging in the upper-right corner of the page and select the current model.

### 8.2.8.3 Add an existing service version

This topic describes how to use the Add Existing Service Version mode to deploy online prediction services.

#### Prerequisites

Before you use the Add Existing Service Version mode to deploy online prediction services, ensure that you have deployed one version of online prediction services through the *New Service* mode.

#### Procedure

1. Complete *Preparations for online model prediction*.
2. Select Add Existing Service Version.
3. Select a deployed model. It takes several minutes to add a version.
4. After the model is deployed, select the added version from the Current Version drop-down list.

### 8.2.8.4 Create a blue-green deployment

This topic describes how to use the Create Blue-green Deployment mode to deploy online prediction services.

#### Prerequisites

Before you use the Create Blue-green Deployment mode to deploy online prediction services, ensure that you have deployed two versions of online prediction services through the *New Service* and *Add Existing Service Version* modes.

## Context

In the blue-green deployment mode, you can deploy and test the target version without stopping the source version. After confirming that the target version is running normally, switch all traffic to the target version. Blue-green deployment is safe and does not interrupt services.

## Procedure

1. Complete *Preparations for online model prediction*.
2. Select the deployed model service and click Deploy.  
The deployment may take several minutes.
3. Click Switch Traffic and adjust the ratio of traffic forwarded to the two models.  
The initial ratio is 100% for both models.
4. Perform online debugging.
  - a) Click Online Debugging in the upper-right corner of the page and select the deployed model.
  - b) Enter data (feature input) in Body. For example, the body information of the logistic regression model for heart disease prediction is as follows:

```
[{"sex":0,"cp":0,"fbs":0,"restecg":0,"exang":0,"slop":0,"thal":0,"age":0,"trestbps":0,"chol":0,"thalach":0,"oldpeak":0,"ca":0}]
```

- c) Click Run and check the result.

## 8.2.9 DataWorks task scheduling

After you have run all nodes in an experiment, you can deploy the experiment to DataWorks and schedule DataWorks to periodically run the experiment. This topic uses air quality prediction as an example scenario.

## Prerequisites

Before scheduling an experiment, you must make sure that you have successfully run all nodes in the experiment and that the experiment is deployed to DataWorks.

## Procedure

1. *Log on to the Apsara Stack Machine Learning Platform for AI console*. Click Experiments in the left-side navigation pane.

2. Click the **My Experiments** tab and select an experiment to navigate to the canvas.



**Notice:**

Make sure all components have been run in the experiment. A green check means that the component is running correctly.

3. In the upper-left corner of the canvas, choose **Deploy > Schedule DataWorks Tasks** to go to DataStudio.
4. In the DataStudio console, choose **Create > Algorithm > Machine Learning Platform for AI**, and then create a **Machine Learning experiment node**.
5. In the **Create Node** dialog box, enter the node name, select the target folder, and click **Submit**.



**Notice:**

You must select a target folder for the algorithm type.

After the experiment node is created.

6. Select the experiment from the drop-down list.
7. Configure task scheduling parameters, including the recurrence, input, and output parameters.
8. Click **Submit**. The task will be executed the next day.
9. Click **Administration** in the upper-right corner to go to the administration page. You can view the status of the machine learning task and the system log. You can also perform other operations such as adding retroactive data and testing the experiment.

## 8.3 Components

### 8.3.1 Overview

This topic describes how to use and configure machine learning components.

When building a machine learning experiment, you can select components based on the features of existing data to generate a model and make accurate predictions for your business.

Each component has one or more input or output ports. You can move the pointer over the ports to view their descriptions and connect the components.

## 8.3.2 Data source and target

**This topic describes components in the Data Source/Target category, such as the Read MaxCompute Table and Write MaxCompute Table components.**

### Read MaxCompute tables

**You can use the Read MaxCompute Table component to read MaxCompute tables. By default, this component reads data of the current project. If you want to read data from tables in another project for which you have access, you can prefix the table name with the project name in the `project name. table name` format. For example, `tianchi_project.weibo_data`. After you specify the input table, the system reads the structural data of the table. You can click the Column Information tab to view the data. This component does not support views.**

**If the selected input table is a partitioned table, the back end automatically selects the Partition checkbox. You can select or configure partition parameters. Only one partition can be selected. If you do not select the Partition checkbox or do not specify the partition parameters, the whole table is selected. If the input table is non-partitioned, the Partition checkbox cannot be selected.**

### Write MaxCompute tables

**You can use the Write MaxCompute Table component to write data to tables in the current project or tables in other projects. This component can write data to partitions. Partitions must be created for the table in the MaxCompute console before this component can write data to the partitions. You can set the table lifecycle measured in days.**

## 8.3.3 Data preprocessing

### 8.3.3.1 Sampling and filtering

#### 8.3.3.1.1 Random sampling

**Data is sampled randomly and independently. You can specify a ratio or quantity of samples to be taken and choose whether to enable sampling with replacement.**

### Parameter settings

Parameters Setting

Tuning

Sample Size Specify either the sample size or...

Sampling Fraction Range: (0,2). Specify either...

☐ Sampling with Replacement

Random Seed Positive integer.

#### PAI command

```
Pai -name sample
 -project algo_public
 -DinputTableName=wbpc
 -DoutputTableName=wbpc_sample
 -Dratio=0.3;
```

#### Algorithm parameters

Table 8-1: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	-	-

Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions in the input table are selected by default.
<b>ratio</b>	Required. The sampling fraction.	(0, 1)	-
<b>outputTableName</b>	Required. The name of the output table.	-	-
<b>outputTablePartition</b>	Optional. The partition of the output table.	-	The output table is a non-partitioned table by default.
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer in the range of [1, 3650]	No lifecycle is set by default.

### 8.3.3.1.2 Weighted sampling

Sample data is collected based on weights. The weight column must be of double or int type. Data is sampled based on the value of its corresponding weight. For example, data with a col value of 1.2 has a higher probability to be sampled than data with a col value of 1.0.

Parameter settings


Parameters Setting

Sample Size Specify either the sample size or...

Sampling Fraction Range: (0,1). Specify either...

☐ Sampling with Replacement

Weight Columns Double or bigint type.



Random Seed Positive integer.

Empty by default

Table 8-2: Parameter settings

Parameter	Description
Sample Size	You can specify the number of samples to be taken, which is 10,000 by default. For sampling without replacement, the number of samples cannot be greater than the number of data entries.
Sampling Fraction	You can use either the Sample Size or Sampling Fraction parameter. You can choose sampling with or without replacement, the latter of which is used by default. Select the checkbox to enable sampling with replacement.
Weight Columns	You can select a weight column from the drop-down list. The weight column can be of the double or bigint type.
Random Seed	The random seed, which is a positive integer. This parameter is empty by default.

- You can choose sampling with or without replacement, the latter of which is used by default. Select the checkbox to enable sampling with replacement.
- You can specify the number of samples to be taken, which is 10,000 by default.



**Note:**



**For sampling without replacement, the number of samples cannot be greater than the number of data entries.**

- You can select a weight column from the drop-down list. The weight column can be of the double or bigint type.

PAI command

```
PAI -name WeightedSample
-project algo_public
-DprobCol="previous"
-DsampleSize="500"
-DoutputTableName="test2"
-DinputPartitions="pt=20150501"
-DinputTableName="bank_data_partition";
```

Table 8-3: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is algo_public. If you change the name, the system reports an error.
<b>replace</b>	Indicates whether sampled data is replaced. If this parameter is set to true, data is replaced after it is sampled. If this parameter is set to false, data is not replaced after it is sampled.
<b>probCol</b>	The columns to be weighted. Each value indicates the weight of an entry. Normalization is not required.
<b>sampleSize</b>	The number of samples to be taken. For sampling without replacement, the number of samples cannot be greater than the number of data entries.
<b>outputTableNames</b>	The name of the output table. Separate multiple table names with commas (,).
<b>inputPartitions</b>	Optional. The partitions selected from the input table for training. If no partitions are specified, the entire table is selected.
<b>inputTableName</b>	The name of the input table.
<b>replace</b>	Optional. This parameter indicates whether sampled data is replaced. If this parameter is set to true, data is replaced after it is sampled. If this parameter is set to false, data is not replaced after it is sampled.

### 8.3.3.1.3 Filtering and mapping

You can filter data based on filtering expressions and rename columns.

Parameter settings

1. Use the WHERE condition to filter data similar to how it would function in an SQL statement.

**Filtering conditions:** Operators available include the equal (=), not equal (!=), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=) signs, as well as like and rlike.

2. Rename columns.

PAI command

```
PAI -name Filter
-project algo_public
-DoutTableName="test_9"
-DinputPartitions="pt=20150501"
-DinputTableName="bank_data_partition"
-Dfilter="age>=40";
```

Table 8-4: Parameters

Parameter	Description
name	The name of the component.
project	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is algo_public. If you change the name, the system reports an error.
outTableName	The name of the output table.
inputPartitions	Optional. The partitions selected from the input table for training. If no partitions are specified, the entire table is selected.
inputTableName	The name of the input table.
filter	The WHERE condition to filter data. Operators available include the equal (=), not equal (!=), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=) signs, as well as like and rlike.

### 8.3.3.1.4 Stratified sampling

**Stratified sampling is a statistical computing method. It works by dividing a population into several strata based on specified features, performing random sampling at each stratum, and creating a sample collection.**

Table 8-5: Parameter settings

Parameter	Description
Column Settings	<b>Stratification Column: Required.</b> Samples are stratified based on this column.
Parameter Settings	<b>Sampling Fraction/Sample Size: Required.</b> A value less than 1 represents the sampling fraction per stratum. A value greater than 1 represents the number of samples at each stratum.
	<b>Other Sampling Configurations: Optional.</b> This parameter allows you to collect different numbers of samples at different strata.
	<b>Random Seed: Optional.</b> Valid values: 1, 2, 3, 4, 5, 6, and 7.

PAI command

```
Pai -name sample
 -project algo_public
 -DinputTableName=wbpc
 -DoutputTableName=wbpc_sample
 -DstrataColName="label"
 -DsampleSize="A:200,B:300,C:500"
 -DrandomSeed=1007
 -Dlifecycle=30
```

Algorithm parameters

Table 8-6: Parameters

Parameter	Description	Valid values	Default value
inputTableName	<b>Required.</b> The name of the input table.	-	-

Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	<b>Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code>. To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code>. Separate multiple partitions with commas (,).</b>	-	<b>All partitions in the input table are selected by default.</b>
<b>strataColName</b>	<b>Required. The stratification column.</b>	-	-
<b>outputTableName</b>	<b>Required. The name of the output table.</b>	-	-
<b>sampleSize</b>	<b>Optional. An integer value that specifies the number of samples taken from each stratum. A string value must be in the <code>strata0:n0,strata1:n1....</code> format. Each item in the string represents the number of samples to be taken from the corresponding stratum.</b>	-	

Parameter	Description	Valid values	Default value
<b>sampleRatio</b>	<b>Optional.</b> A decimal value from 0 to 1 that represents the ratio of data for each stratum to be sampled. A string value must be in the <code>strata0:r0,strata1:r1...</code> format. Each item in the string represents the sampling fraction for the corresponding stratum.	-	-
<b>randomSeed</b>	<b>Optional.</b> The number of random seeds.	-	0
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	A positive integer in the range of [1, 3650]	No lifecycle is set by default.
<b>coreNum</b>	<b>Optional.</b> The number of cores.	-	Automatically calculated.
<b>memSizePerCore</b>	<b>Optional.</b> The memory size of each core.	-	Automatically calculated.

### 8.3.3.2 Data merge

#### 8.3.3.2.1 Join

This component merges two tables by associating the information in the tables and outputting the specified columns. This component is similar to the JOIN statement of SQL.

Parameter settings

- Join types: left join, internal join, right join, and full join.
- Only the equation condition is supported.

- You can manually add or delete join conditions.

PAI command

No PAI command is available.

### 8.3.3.2.2 Merge columns

You can merge data of two tables by column. The two tables must have the same number of rows.

Parameter settings

Procedure

1. Select input columns from the left table.
2. Select input columns from the right table.

When merging columns:

- The two tables must have the same number of rows.
- The names of output columns selected from the left and right tables cannot be the same.
- When selecting an output column, you can change its name.
- If no output columns are selected from the left or right table, the whole table is selected. In this case, if Automatically Rename Output Columns is selected, the duplicate columns are renamed and then output.

PAI command

```
PAI -name AppendColumns
-project algo_public
-DoutputTableColNames="petal_length,petal_width,petal_length2,
petal_width2"
-DautoRenameCol="false"
-DoutputTableName="pai_temp_770_6840_1"
-DinputTableNames="iris_twopartition,iris_twopartition"
-DinputPartitionsInfoList="dt=20150125/dp=20150124;dt=20150124/dp=
20150123"
-DselectedColNamesList="petal_length,petal_width;sepal_length,
sepal_width";
```

Table 8-7: Parameters

Parameter	Description
name	The name of the component.

Parameter	Description
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <b>algo_public</b> . If you change the name, the system reports an error.
<b>outputTableColNames</b>	The names of the columns in the new table. The column names must be separated with commas (,). If <b>autoRenameCol</b> is set to true, this parameter is ignored.
<b>autoRenameCol</b>	Optional. This parameter specifies whether to automatically rename the columns in the output table. If the value is true, the columns are renamed. If the value is false, the columns are not renamed. Default value: false.
<b>outputTableName</b>	The name of the output table.
<b>inputTableNames</b>	The name of the input table. Separate multiple table names with commas (,).
<b>inputPartitionsInfo</b>	Optional. A list of partitions selected from the corresponding input tables. Partitions of the same table must be separated with commas (,) and partitions of different tables must be separated with semicolons (;).
<b>selectedColNamesList</b>	The names of selected columns. The names of columns in the same table must be separated with commas (,) and the names of columns in different tables must be separated with semicolons (;).

### 8.3.3.2.3 Merge rows (UNION)

To merge the data of two tables by row, the quantity and data type of the output columns selected from the left and right tables must be the same. The function is integrated with the UNION and UNION ALL functions.

Parameter settings

- During the merge process, the numbers of columns selected from the left and right tables must be the same, and the data types of the corresponding columns must be the same.
- You can enter conditions in the text box by which to filter and select columns. The whole table is selected by default. Operators available include the equal (=), not equal (!=), greater than (>), less than (<), greater than or equal to (>=), and less than or equal to (<=) signs, as well as like and rlike.

- **Remove Duplicates** is selected by default. When this option is selected, duplicate rows in the output table are removed.

The following figure shows the union columns selected from the left table.

The screenshot shows a 'Select Column' dialog box with a search bar at the top. Below the search bar, there are two main sections: 'Select All' and 'Selected'. The 'Select All' section has two expandable categories: 'STRING' and 'BIGINT'. Under 'STRING', the columns 'age', 'trestbps', 'chol', 'thalach', 'oldpeak', and 'ca' are listed, with 'age' and 'ca' checked. Under 'BIGINT', the columns 'sex', 'cp', 'fbs', 'restecg', and 'exang' are listed, with 'cp' and 'fbs' checked. The 'Selected' section shows a table with the following columns and types:

Column	Type
age	STRING
ca	STRING
cp	BIGINT
fbs	BIGINT
slop	BIGINT
thal	BIGINT

At the bottom right of the dialog box, there are 'OK' and 'Cancel' buttons.

The following figure shows the union columns selected from the right table.



Select Column

Search by keyword

Select All

STRING

☒ age
 ☐ trestbps
 ☐ chol
 ☐ thalach
 ☐ oldpeak
 ☒ ca

BIGINT

☐ sex
 ☒ cp
 ☒ fbs
 ☐ restecg
 ☐ exang

Selected

List

Edit

Column	Type
age	STRING
ca	STRING
cp	BIGINT
fbs	BIGINT
slop	BIGINT
thal	BIGINT

OK

Cancel

PAI command

**No PAI command is available.**

### 8.3.3.3 Others

#### 8.3.3.3.1 Add ID column

**You can append an ID column to a table as the first column and save the table as a new table.**

Parameter settings

Parameters Setting

Tuning

All Selected by Default

Select Column

ID Column

append\_id

PAI command

```
PAI -name AppendId
```

```
-project algo_public
-DIDColName="append_id"
-DoutputTableName="test_11"
-DinputTableName="bank_data"
-DselectedColNames="age,campaign,cons_conf_idx,cons_price_idx,
emp_var_rate,euribor3m,nr_employed,pdays,poutcome,previous,y";
```

Table 8-8: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <b>algo_public</b> . If you change the name, the system reports an error.
<b>IDColName</b>	The name of the appended ID column. ID numbers start from 0 and increment by one. Example: 0, 1, 2, 3, ...
<b>outputTableName</b>	The name of the output table.
<b>inputTableName</b>	The name of the input table.
<b>selectedColNames</b>	The names of the columns to be retained. Separate multiple columns with commas (,).

### 8.3.3.3.2 Split

This component is used to split an input table or a partition based on a specified ratio, and output two tables from two output ports.

Algorithm component

#### Parameter settings

- The Split component has two output ports.
- In Parameter settings, if the splitting fraction is set to 0.7, the left output port outputs 70% of the data and the right output port outputs 30% of the data.

PAI command

```
pai -name split -project algo_public
-DinputTableName=wbpc
-Doutput1TableName=wbpc_split1
-Doutput2TableName=wbpc_split2
```

-Dfraction=0.25;

Table 8-9: Parameter settings

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions in the input table are selected by default.
<b>output1TableName</b>	<b>Required.</b> The name of output table 1.	-	-
<b>output1TablePartitions</b>	<b>Optional.</b> The partitions in output table 1.	-	Output table 1 is a non-partitioned table by default.
<b>output2TableName</b>	<b>Required.</b> The name of output table 2.	-	-
<b>output2TablePartitions</b>	<b>Optional.</b> The partitions in output table 2.	-	Output table 2 is a non-partitioned table by default.
<b>fraction</b>	<b>Required.</b> The portion of data diverted to output table 1.	(0, 1)	-

Parameter	Description	Valid values	Default value
lifecycle	Optional. The lifecycle of the output table.	A positive integer in the range of [1, 3650]	No lifecycle is set by default.

### 8.3.3.3.3 Missing value imputation

This component replaces a null value or a specified value with the maximum, minimum, average, or custom value. A list of values is defined to impute the missing values in an input table with the specified values.

- This component can replace a numeric null value with the maximum, minimum, average, or custom value.
- This component can also replace a null string, empty string, null and empty string, or specified value with a custom value.
- The missing values to be imputed can be null strings, empty strings, or custom values. If you choose empty strings, the data type of the target column must be string.

The parameters for the two input ports are as follows:

- **inputTableName:** the name of the input table for which to replace missing data.
- **inputParaTableName:** the name of the input configuration table that contains parameters generated by the missing value imputation node. Based on this parameter, configuration parameters in one table can be applied to a new table.

Parameters for the two output ports are as follows:

- **outputTableName:** the name of the imputed output table.
- **outputParaTableName:** the name of the output configuration table, which can be applied to other datasets.
- **Columns to Impute:** the names of the columns for which to replace missing values.
- **Original Value:** the values to be replaced.
- **Replaced With:** the replacement values.

PAI command

```
PAI -name FillMissingValues
 -project algo_public
 -Dconfigs="poutcome,null-empty,testing" \
 -DoutputTableName="test_3"
 -DinputPartitions="pt=20150501"
```

```
-DinputTableName="bank_data_partition";
```

#### Algorithm parameters

Parameter	Description	Valid value	Default value
<b>inputTableName</b>	Required. The name of the input table.	Table name	N/A
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training.	Partition name	The whole table is selected by default.
<b>outputTableName</b>	Required. The name of the output table.	Table name	N/A

Parameter	Description	Valid value	Default value
<b>configs</b>	<p><b>Required. The configurations for missing value imputation.</b></p> <p><b>Example:</b> <code>col1, null, 3.14; col2, empty, hello; col3, empty-null, world, where null indicates a null value and empty indicates an empty string. If you choose to use empty strings to fill the target columns, the data type of the target column must be string. The variables used to specify the replacement value as maximum, minimum, and average are max, min, and mean respectively. If you want to impute a custom value to the target column, use a user-defined variable in the <code>col4, user-defined, str, str123</code> format.</code></p>	N/A	N/A
<b>outputParaTableName</b>	<p><b>Required. The name of the output configuration table.</b></p> <p>.</p>	Table name	N/A

Parameter	Description	Valid value	Default value
<b>inputParaTableName</b>	<b>Optional. The name of the input configuration table.</b>	<b>Table name</b>	<b>No input configuration table is set by default.</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>

## Examples

### Test data

- **SQL statement to generate data:**

```
drop table if exists fill_missing_values_test_input;
create table fill_missing_values_test_input(
 col_string string,
 col_bigint bigint,
 col_double double,
 col_boolean boolean,
 col_datetime datetime);
insert overwrite table fill_missing_values_test_input
select
 *
from
(
 select
 '01' as col_string,
 10 as col_bigint,
 10.1 as col_double,
 True as col_boolean,
 cast('2016-07-01 10:00:00' as datetime) as col_datetime
 from dual
 union all
 select
 cast(null as string) as col_string,
 11 as col_bigint,
 10.2 as col_double,
 False as col_boolean,
 cast('2016-07-02 10:00:00' as datetime) as col_datetime
 from dual
 union all
 select
 '02' as col_string,
 cast(null as bigint) as col_bigint,
 10.3 as col_double,
 True as col_boolean,
```

```

 cast('2016-07-03 10:00:00' as datetime) as col_datetime
 from dual
union all
select
 '03' as col_string,
 12 as col_bigint,
 cast(null as double) as col_double,
 False as col_boolean,
 cast('2016-07-04 10:00:00' as datetime) as col_datetime
 from dual
union all
select
 '04' as col_string,
 13 as col_bigint,
 10.4 as col_double,
 cast(null as boolean) as col_boolean,
 cast('2016-07-05 10:00:00' as datetime) as col_datetime
 from dual
union all
select
 '05' as col_string,
 14 as col_bigint,
 10.5 as col_double,
 True as col_boolean,
 cast(null as datetime) as col_datetime
 from dual
) tmp;

```

#### • Input description

col_string	col_bigint	col_double	col_boolean	col_datetime
04	13	10.4	NULL	2016-07-05 10:00:00
02	NULL	10.3	true	2016-07-03 10:00:00
03	12	NULL	false	2016-07-04 10:00:00
NULL	11	10.2	false	2016-07-02 10:00:00
01	10	10.1	true	2016-07-01 10:00:00
05	14	10.5	true	NULL

#### PAI command

```

drop table if exists fill_missing_values_test_input_output;
drop table if exists fill_missing_values_test_input_model_output;
PAI -name FillMissingValues
-project algo_public
-Dconfigs="col_double,null,mean;col_string,null-empty,str_type_empty;
col_bigint,null,max;col_boolean,null,true;col_datetime,null,2016-07-06
10:00:00"
-DoutputParaTableName="fill_missing_values_test_input_model_output"
-Dlifecycle="28"

```



```
-DoutputTableName="fill_missing_values_test_input_output"
-DinputTableName="fill_missing_values_test_input";
drop table if exists fill_missing_values_test_input_output_using_model
;
drop table if exists fill_missing_values_test_input_output_us
ing_model_model_output;
PAI -name FillMissingValues
-project algo_public
-DoutputParaTableName="fill_missing_values_test_input_output_us
ing_model_model_output"
-DinputParaTableName="fill_missing_values_test_input_model_output"
-Dlifecycle="28"
-DoutputTableName="fill_missing_values_test_input_output_using_model"
-DinputTableName="fill_missing_values_test_input";
```

## Output

### • fill\_missing\_values\_test\_input\_output

```
+-----+-----+-----+-----+-----+
+
| col_string | col_bigint | col_double | col_boolean | col_datetime
|
+-----+-----+-----+-----+-----+
+
| 04 | 13 | 10.4 | true | 2016-07-05 10
:00:00 |
| 02 | 14 | 10.3 | true | 2016-07-03 10
:00:00 |
| 03 | 12 | 10.3 | false | 2016-07-04 10
:00:00 |
| str_type_empty | 11 | 10.2 | false | 2016-07-
02 10:00:00 |
| 01 | 10 | 10.1 | true | 2016-07-01 10
:00:00 |
| 05 | 14 | 10.5 | true | 2016-07-06 10
:00:00 |
+-----+-----+-----+-----+-----+
+
```

### • fill\_missing\_values\_test\_input\_model\_output

```
+-----+-----+
| feature | json |
+-----+-----+
| col_string | {"name": "fillMissingValues", "type": "string", "
paras":{"missing_value_type": "null-empty", "replaced_value": "
str_type_empty"}} |
| col_bigint | {"name": "fillMissingValues", "type": "bigint", "
paras":{"missing_value_type": "null", "replaced_value": 14}} |
| col_double | {"name": "fillMissingValues", "type": "double", "
paras":{"missing_value_type": "null", "replaced_value": 10.3}} |
| col_boolean | {"name": "fillMissingValues", "type": "boolean", "
paras":{"missing_value_type": "null", "replaced_value": 1}} |
| col_datetime | {"name": "fillMissingValues", "type": "datetime", "
paras":{"missing_value_type": "null", "replaced_value": 1467770400
000}} |
```

- **fill\_missing\_values\_test\_input\_output\_using\_model**

```
+-----+-----+-----+-----+-----+
+
| col_string | col_bigint | col_double | col_boolean | col_datetime
|
+-----+-----+-----+-----+-----+
+
| 04 | 13 | 10.4 | true | 2016-07-05 10
:00:00 |
| 02 | 14 | 10.3 | true | 2016-07-03 10
:00:00 |
| 03 | 12 | 10.3 | false | 2016-07-04 10
:00:00 |
| str_type_empty | 11 | 10.2 | false | 2016-07-
02 10:00:00 |
| 01 | 10 | 10.1 | true | 2016-07-01 10
:00:00 |
| 05 | 14 | 10.5 | true | 2016-07-06 10
:00:00 |
+-----+-----+-----+-----+-----+
+
```

- **fill\_missing\_values\_test\_input\_output\_using\_model\_model\_output**

```
+-----+-----+
| feature | json |
+-----+-----+
| col_string | {"name": "fillMissingValues", "type": "string", "
paras":{"missing_value_type": "null-empty", "replaced_value": "
str_type_empty"}} |
| col_bigint | {"name": "fillMissingValues", "type": "bigint", "
paras":{"missing_value_type": "null", "replaced_value": 14}} |
| col_double | {"name": "fillMissingValues", "type": "double", "
paras":{"missing_value_type": "null", "replaced_value": 10.3}} |
| col_boolean | {"name": "fillMissingValues", "type": "boolean", "
paras":{"missing_value_type": "null", "replaced_value": 1}} |
| col_datetime | {"name": "fillMissingValues", "type": "datetime", "
paras":{"missing_value_type": "null", "replaced_value": 1467770400
000}} |
+-----+-----+
```

### 8.3.3.3.4 Normalization

You can normalize one or more columns in a table and save the generated data to a new table.

Linear function transformation is supported. The transformation expression is  $y = (x - \text{MinValue}) / (\text{MaxValue} - \text{MinValue})$ .

*MaxValue* and *MinValue* indicate the maximum and minimum values of the sample respectively.

- Click Columns to select the columns to be normalized. Double and bigint types are supported.

- You can choose whether to retain the original columns. If you select the corresponding checkbox, the original columns will be retained. Processed columns will be renamed.

PAI command

```
PAI -name normalize_wf
 -project algo_public
 -DkeepOriginal="true"
 -DoutputTableName="test_4"
 -DinputPartitions="pt=20150501"
 -DinputTableName="bank_data_partition"
 -DselectedColNames="emp_var_rate,euribor3m";
```

Algorithm parameters

Table 8-10: Parameters

Parameter	Description	Default value
<b>inputTableName</b>	Required. The name of the input table.	N/A
<b>selectedColNames</b>	Optional. The names of columns selected from the input table.	All columns are selected by default.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training.	The whole table is selected by default.
<b>outputTableName</b>	Required. The name of the output table.	N/A
<b>outputParaTableName</b>	Required. The name of the output configuration table.	N/A
<b>inputParaTableName</b>	Optional. The name of the input configuration table.	No input configuration table is set by default.
<b>outputPMMLTableName</b>	Required. The name of the output PMML table.	N/A

Parameter	Description	Default value
<b>keepOriginal</b>	<b>Optional. This parameter specifies whether to retain the original columns. If keepOriginal is set to true, processed columns are renamed with the normalized_ prefix and the original columns are retained and their data overwritten. If keepOriginal is set to false, all columns are retained but not renamed.</b>	<b>false</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>Automatically calculated.</b>

### 8.3.3.3.5 Standardization

You can standardize one or more columns in a table and save the generated data to a new table.

- The formula used for standardization is  $(X - \text{Mean}) / (\text{Standard deviation})$ .
  - **Mean:** The mean of samples.
  - **Standard deviation:** The standard deviation of samples. This variable is used when samples are used to calculate the total deviation. To make the calculated

value closer to the mean, you must moderately increase the calculated standard deviation by using the formula  $\frac{1}{N-1}$ .

- The formula for calculating the standard deviation of samples:

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2}$$

$\bar{X}$  represents the mean of samples  $X_1, X_2, \dots, X_n$ .

- You can choose whether to retain the original columns. If you select the corresponding checkbox, the original columns will be retained. Processed columns will be renamed.
- Click Columns and select columns to be standardized. Double and bigint types are supported.

PAI command

```
PAI -name Standardize
 -project algo_public
 -DkeepOriginal="false"
 -DoutputTableName="test_5"
 -DinputTablePartitions="pt=20150501"
 -DinputTableName="bank_data_partition"
 -DselectedColNames="euribor3m,pdays"
```

Standardization component

Parameters for the two input ports are as follows:

- **inputTableName:** the name of the input table to be standardized.
- **inputParaTableName:** the name of the input configuration table that contains the parameters generated by the standardization node. You can use an input configuration table to apply the configuration parameters of one table to a new table.

Parameters for the two output ports are as follows:

- **outputTableName:** the name of the standardized output table.

- **outputParaTableName:** the name of the output parameter table, which can be applied to other datasets.

Standardization parameters

**The corresponding algorithm parameter for Reserve Original Columns is keepOriginal.**

Algorithm parameters

Table 8-11: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>selectedColNames</b>	<b>Optional.</b> The names of columns selected from the input table.	-	All columns are selected by default.
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions in the input table are selected by default.
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>outputParaTableName</b>	<b>Required.</b> The name of the output configuration table.	-	-
<b>outputPartition</b>	<b>Optional.</b> The partitions in the output table.	-	-
<b>inputParaTableName</b>	<b>Optional.</b> The name of the input configuration table.	-	No input configuration table is set by default.

Parameter	Description	Valid values	Default value
<b>keepOriginal</b>	<b>Optional. This parameter specifies whether to retain the original columns. If this parameter is set to true, the original columns are retained and the column name is suffixed with <code>_orig</code>.</b>	<b>true and false</b>	<b>false</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>-</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>-</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>-</b>	<b>Automatically calculated.</b>

## Examples

```

drop table if exists standardize_test_input;
create table standardize_test_input(
col_string string,
col_bigint bigint,
col_double double,
col_boolean boolean,
col_datetime datetime);
insert overwrite table standardize_test_input select * from (

select '01' as col_string,
10 as col_bigint,
10.1 as col_double,
True as col_boolean,
cast('2016-07-01 10:00:00' as datetime) as col_datetime from dual
union all
select cast(null as string) as col_string,
11 as col_bigint,
10.2 as col_double,
False as col_boolean,
cast('2016-07-02 10:00:00' as datetime) as col_datetime from
dual union all
select
'02' as col_string,
cast(null as bigint) as col_bigint,
10.3 as col_double,
True as col_boolean,
cast('2016-07-03 10:00:00' as datetime) as col_datetime from
dual union all
select '03' as col_string,
12 as col_bigint,
cast(null as double) as col_double,
False as col_boolean,

```

```

cast('2016-07-04 10:00:00' as datetime) as col_datetime from
dual union all
select '04' as col_string,
13 as col_bigint,
10.4 as col_double,
cast(null as boolean) as col_boolean,
cast('2016-07-05 10:00:00' as datetime) as col_datetime from
dual union all
select '05' as col_string,
14 as col_bigint,
10.5 as col_double,
True as col_boolean,
cast(null as datetime) as col_datetime from dual) tmp;

```

## PAI command

```

drop table if exists standardize_test_input_output;
drop table if exists standardize_test_input_model_output;
PAI -name Standardize
-project algo_public
-DoutputParaTableName="standardize_test_input_model_output"
-Dlifecycle="28"
-DoutputTableName="standardize_test_input_output"
-DinputTableName="standardize_test_input"
-DselectedColNames="col_double,col_bigint"
-DkeepOriginal="true";
drop table if exists standardize_test_input_output_using_model;
drop table if exists standardize_test_input_output_using_model_model_output;
PAI -name Standardize
-project algo_public
-DoutputParaTableName="standardize_test_input_output_using_model_model_output"
-DinputParaTableName="standardize_test_input_model_output"
-Dlifecycle="28"
-DoutputTableName="standardize_test_input_output_using_model"
-DinputTableName="standardize_test_input"

```

## Input description

Table 8-12: standardize\_test\_input

col_string	col_bigint	col_double	col_boolean	col_datetime
01	10	10.1	true	2016-07-01 10:00:00
NULL	11	10.2	false	2016-07-02 10:00:00
02	NULL	10.3	true	2016-07-03 10:00:00
03	12	NULL	false	2016-07-04 10:00:00
04	13	10.4	NULL	2016-07-05 10:00:00



col_string	col_bigint	col_double	col_boolean	col_datetime
05	14	10.5	true	NULL

Output description

Table 8-13: standardize\_test\_input\_output

col_string	col_bigint	col_double	col_boolean	col_datetime	stdized_col_bigint	stdized_col_double
01	10	10.1	true	2016-0	-1.264911064	-1.264911064
NULL	11	10.2	false	2016-07-02 10:00:00	-0.6324555320336759	-0.6324555320341972
02	NULL	10.3	true	2016-07-03 10:00:00	NULL	0.0
03	12	NULL	false	2016-07-04 10:00:00	0.0	NULL
04	13	10.4	NULL	2016-07-05 10:00:00	0.6324555320336759	0.6324555320341859
05	14	10.5	true	NULL	1.2649110640673518	1.2649110640683718

Table 8-14: standardize\_test\_input\_model\_output

Feature	json
col_bigint	{"name": "standardize", "type": "bigint", "paras": {"mean": 12, "std": 1.58113883008419}}
col_double	{"name": "standardize", "type": "double", "paras": {"mean": 10.3, "std": 0.1581138830082909}}

Table 8-15: standardize\_test\_input\_output\_using\_model

col_string	col_bigint	col_double	col_boolean	col_datetime
01	-1.2649110640 673515	-1.2649110640 68383	true	2016-07-01 10: 00:00
NULL	-0.6324555320 336758	-0.6324555320 341971	false	2016-07-02 10: 00:00
02	NULL	0.0	true	2016-07-03 10: 00:00
03	0.0	NULL	false	2016-07-04 10: 00:00
04	0.6324555320 336758	0.6324555320 341858	NULL	2016-07-05 10: 00:00
05	1.2649110640 673515	1.2649110640 683716	true	NULL

Table 8-16: standardize\_test\_input\_output\_using\_model\_model\_output

feature	json
col_bigint	{"name": "standardize", "type": "bigint", "paras": {"mean": 12, "std": 1.58113883008419}}
col_double	{"name": "standardize", "type": "double", "paras": {"mean": 10.3, "std": 0.1581138830082909}}

### 8.3.3.3.6 KV to Table

This component is used to convert KV pairs to a table. The key is converted to a table column, while the value is converted to a column value in the corresponding row.

**KV table format definition:**

- A key is the index of a column. Key values can be of the bigint or double types.
- A KV table can be input in the sparse format to algorithm components such as logistic and linear regression.
- Keys must be of the string type. You can input a key\_map table to the KV to Table component to map keys to columns. This component outputs a key\_map table

that contains all key-column mappings after conversion, regardless of whether you input a key\_map table.

kv
1:10;2:20;3:30

**key\_map table format definition:** a table that contains index-to-column mappings and data type information. The data types of the col\_name, col\_index, and col\_datatype columns must be string. The default data type of the col\_datatype column is double if not specified.

col_name	col_index	col_datatype
col1	1	bigint
col2	2	double

PAI command

```
PAI -name KVToTable
 -project algo_public
 -DinputTableName=test
 -DoutputTableName=test_out
 -DoutputKeyMapTableName=test_keymap_out
 -DkvColName=kv;
```

Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	The table cannot be empty.
kvColName	Required. The name of the KV column.	Only one column can be selected.	-
outputTableName	Required. The name of the output table.	Table name	-
outputKeyMapTableName	Required. The name of the output index table.	Table name	-
inputKeyMapTableName	Optional. The name of the input index table.	Table name	No input index table is set by default.

Parameter	Description	Valid values	Default value
<b>appendColName</b>	Optional. The name of the appended column.	Multiple columns can be selected.	No column is appended by default.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table.	Partition name	No partition is specified by default .
<b>kvDelimiter</b>	Optional. The delimiter used to separate keys and values.	Symbol	The default delimiter is a semicolon (;).
<b>itemDelimiter</b>	Optional. The delimiter used to separate key-value pairs.	Symbol	The default delimiter is a comma (,).
<b>top1200</b>	Optional. This parameter specifies whether to output the first 1,200 columns.	true and false	Default value: true . If the value is false, an error is returned when the number of columns reaches the upper limit.
<b>lifecycle</b>	Optional. The lifecycle of the output table.	An integer greater than or equal to -1.	Default value: -1. This value indicates that no lifecycle is set.
<b>coreNum</b>	Optional. The number of cores.	An integer greater than 0.	Default value : -1. This value indicates that the number of instances is determined by the amount of input data.

Parameter	Description	Valid values	Default value
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	(100, 65536)	<b>Default value: -1. This value indicates that the memory size is determined by the amount of input data.</b>

## Examples

### SQL statement to generate data:

```
drop table if exists test;
create table test as
select
 *
from
(
 select '1:1,2:2,3:-3.3' as kv from dual
 union all
 select '1:10,2:20,3:-33.3' as kv from dual
) tmp;
```

### PAI command

```
PAI -name KVToTable
 -project algo_public
 -DinputTableName=test
 -DoutputTableName=test_out
 -DoutputKeyMapTableName=test_keymap_out
 -DkvColName=kv;
```

## Output

The output table is shown as follows.

kv_1	kv_2	kv_3
1.0	2.0	-3.3
10.0	20.0	-33.3

The output mapping table is shown as follows.

col_name	col_index	col_type
kv_1	1	double
kv_2	2	double
kv_3	3	double

---

+-----+-----+-----+

## Input and output restrictions

Converted columns include appended columns and columns converted from KV pairs. The KV columns are output before the appended columns. MaxCompute supports a maximum of 1,200 columns. When the number of columns exceeds the maximum value, and `top1200` is set to `true`, only the first 1,200 columns are output. If `top1200` is set to `false`, an error is returned. The number of input data entries cannot exceed 100 million.

## Restrictions and guidelines

- If a `key_map` table is input, columns are converted from the keys that exist in both the `key_map` and key-value tables.
- The converted column type can only be numeric.
- If a `key_map` table is input, the data type of the converted key column is the same as that of the `key_map` table. If no `key_map` table is input, the data type of the converted key column is double.
- If a `key_map` table is not input, the name of the converted key column is in the format of 'kv column name'+key'. An error is returned if the key contains any of the following characters: `%&()*+-. /;<>=?`
- If an appended column is specified and the name of the appended column is the same as that of the converted key column, an error is returned indicating a column name conflict.
- If a row contains multiple keys, the values are added.
- A column name can contain up to 128 characters. If more than 128 characters are entered, only the first 128 characters are kept.

### 8.3.3.3.7 Table to KV

This component is used to convert data tables to KV tables. Null values in the table to be converted are not displayed in the KV table. You can specify columns to be retained in the new table. These columns will remain unchanged.

## PAI command

```
PAI -name TableToKV
 -project algo_public
 -DinputTableName=maple_tabletokv_basic_input
 -DoutputTableName=maple_tabletokv_basic_output
 -DselectedColNames=col0,col1,col2
```

```
-DappendColNames=rowid;
```

## Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	Required. The name of the input table.	Table name	-
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>selectedColNames</b>	Optional. The names of selected columns from the input table.	The column type must be bigint or double.	The whole table is selected by default.
<b>appendColNames</b>	Optional. The names of columns to remain unchanged. These columns are written in the output table without any changes.	Multiple columns can be selected.	-
<b>outputTableName</b>	Required. The name of the output KV table.	Table name	-

Parameter	Description	Valid values	Default value
<b>kvDelimiter</b>	<b>Optional. The delimiter used to separate keys and values.</b>	<b>Symbol</b>	<b>The default delimiter is a colon (:).</b>
<b>itemDelimiter</b>	<b>Optional. The delimiter used to separate key-value pairs.</b>	<b>Symbol</b>	<b>The default delimiter is a comma (,).</b>
<b>convertColToIndexId</b>	<b>Optional. This parameter specifies whether to convert columns into IDs.</b>	<b>0 and 1</b>	<b>0</b>
<b>inputKeyMapTableName</b>	<b>Optional. The name of the input index table. This parameter takes effect only in the case of <code>convertColToIndexId=1</code>. If this parameter is not specified, IDs are automatically generated.</b>	<b>Table name</b>	<b>No input index table is set by default.</b>
<b>outputKeyMapTableName</b>	<b>Optional. The name of the output index table. This parameter is required only in the case of <code>convertColToIndexId=1</code>.</b>	<b>Table name</b>	<b>The default value is determined by <code>convertColToIndexId</code>.</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>



Parameter	Description	Valid values	Default value
coreNum	Optional. The number of cores.	This parameter is used with memSizePerCore. The value must be a positive integer in the range of [1, 9999].	Automatically calculated.
memSizePerCore	Optional. The memory size of each core. Unit: MB .	A positive integer in the range of [ 1024, 65536]	Automatically calculated.

## Example 1

## Data generation

rowid	kv
0	col0:1,col1:1.1,col2:2
1	col0:0,col1:1.2,col2:3
2	col0:1,col1:2.3
3	col0:1,col1:0.0,col2:4

## PAI command

```
PAI -name TableToKV
-project algo_public
-DinputTableName=maple_tabletokv_basic_input
-DoutputTableName=maple_tabletokv_basic_output
-DselectedColNames=col0,col1,col2
-DappendColNames=rowid;
```

## Output

The output table is shown as follows.

## maple\_tabletokv\_basic\_output

rowid:bigint	kv:string
0	1:1.1,2:2
1	1:1.2,2:3
2	1:2.3
3	1:0.0,2:4

## Example 2

## PAI command

```
PAI -name TableToKV
-project projectxlib4 -DinputTableName=maple_tabletokv_basic_input
-DoutputTableName=maple_tabletokv_basic_output
-DselectedColNames=col0,col1,col2 -DappendColNames=rowid
-DconvertColToIndexId=1
-DinputKeyMapTableName=maple_test_tabletokv_basic_map_input
-DoutputKeyMapTableName=maple_test_tabletokv_basic_map_output;
```

## Output

The output table is shown as follows.

## maple\_test\_tabletokv\_basic\_map\_output

col_name:string	col_index:string	col_datatype:string
col1	1	bigint
col2	2	double

## Restrictions and guidelines

- If a **key\_map** table is input, columns are converted from the keys that exist in both the **key\_map** and key-value tables.
- If a **key\_map** table is input and its type is different from the input table, the output **key\_map** table uses the type specified by the user.
- The type of the columns that need to be converted into KV pairs in the input table must be **bigint** or **double**.

## 8.3.4 Feature engineering

## 8.3.4.1 Feature transformation

## 8.3.4.1.1 PCA

You can use principal component analysis (PCA) to reduce dimensionality.

- For more information about the PCA algorithm, see [Wikipedia](#).
- This component supports the dense data format.

## PAI command

```
PAI -name PrinCompAnalysis
-project algo_public
-DinputTableName=bank_data
-DeigOutputTableName=pai_temp_2032_17900_2
-DprincompOutputTableName=pai_temp_2032_17900_1
```

```
-DselectedColNames=pdays,previous,emp_var_rate,cons_price_idx,
cons_conf_idx,euribor3m,nr_employed
-DtransType=Simple
-DcalcuType=CORR
-DcontriRate=0.9;
```

## Algorithm parameters

Table 8-17: Parameters

Parameter	Description	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table for PCA.	-
<b>eigOutputTableName</b>	<b>Required.</b> The name of the output table that contains eigenvectors and eigenvalues.	-
<b>princompOutputTableName</b>	<b>Required.</b> The name of the output table after PCA dimensionality reduction.	-
<b>selectedColNames</b>	<b>Required.</b> The names of feature columns that are involved in the PCA procedure.	-
<b>transType</b>	<b>Optional.</b> The method used to transform the original table to the principal component table. Valid values: Simple, Sub-Mean, and Normalization.	Simple
<b>calcuType</b>	<b>Optional.</b> The eigendecomposition mode of the original table. Valid values: CORR, COVAR_SAMP, and COVAR_POP.	CORR
<b>contriRate</b>	<b>Optional.</b> The ratio of information to be retained after dimensionality reduction.	0.9
<b>remainColumns</b>	<b>Optional.</b> The columns retained from the original table after dimensionality reduction.	-

## Sample PCA output

**Table after dimensionality reduction,**shows a sample table after dimensionality reduction.

**Eigenvalues and eigenvectors,**shows the eigenvalues and eigenvectors.

## 8.3.4.2 Feature importance evaluation

### 8.3.4.2.1 Linear model feature importance

You can evaluate the quality of a linear algorithm model based on the predicted and actual output results such as the indicators and residual histogram. Indicators include SST, SSE, SSR, R2, R, MSE, RMSE, MAE, MAD, MAPE, count, yMean, and predictMean.

PAI command

```
pai -name regression_evaluation
-project algo_public
-DinputTableName=input_table
-DyColName=y_col
-DpredictionColName=prediction_col
-DindexOutputTableName=index_output_table
-DresidualOutputTableName=residual_output_table
```

Table 8-18: Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	-	-
inputTablePartitions	Optional. The partitions selected from the input table for training.	-	All partitions in the input table are selected by default.
yColName	Required. The name of the expected dependent variable column in the input table. It must be a numerical value.	-	-
predictionColName	Required. The name of the predicted dependent variable column. It must be a numerical value.	-	-

Parameter	Description	Valid values	Default value
<b>indexOutputTableName</b>	<b>Required.</b> The name of the regression indicator output table.	-	-
<b>residualOutputTableName</b>	<b>Required.</b> The name of the residual histogram output table.	-	-
<b>intervalNum</b>	<b>Optional.</b> The number of intervals to divide the histogram over.	-	100
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	<b>Optional.</b> The number of cores.	-	Automatically calculated.
<b>memSizePerCore</b>	<b>Optional.</b> The memory size of each core.	-	Automatically calculated.

## Output

The output table is in JSON format. [Table 8-19: Field description](#) describes the JSON fields.

Table 8-19: Field description

Field	Description
<b>SST</b>	Total sum of squares.
<b>SSE</b>	Sum of squared errors.
<b>SSR</b>	Sum of squares due to regression.
<b>R2</b>	Coefficient of determination.
<b>R</b>	Coefficient of multiple correlation.
<b>MSE</b>	Mean squared error.
<b>RMSE</b>	Root-mean-square error.

Field	Description
MAE	Mean absolute error.
MAD	Mean absolute difference.
MAPE	Mean absolute percentage error.
count	Number of rows.
yMean	Mean of expected dependent variables.
predictionMean	Mean of prediction results.

### 8.3.4.2.2 Random forest feature importance

You can calculate the importance of features in a random forest model.

Column settings


Fields Setting

Parameters Setting

Feature Columns Optional. ?

Select Column

Target Column Required.



PAI command

```
pai -name feature_importance
-project algo_public
-DinputTableName=input
-DoutputTableName=output
-Dlabel=label
-DmodelName=model
```

Algorithm parameters

Table 8-20: Parameters

Parameter	Description	Default value
inputTableName	Required. The name of the input table.	-
outputTableName	Required. The name of the output table.	-

Parameter	Description	Default value
labelColName	Required. The name of the label column.	-
modelName	Required. The name of the input model.	-
featureColNames	Optional. The names of feature columns selected from the input table.	All columns except the label column are selected by default.
inputTablePartitions	Optional. The partitions selected from the input table.	The whole table is selected by default.
lifecycle	Optional. The lifecycle of the output table.	No lifecycle is set by default.
coreNum	Optional. The number of cores.	Automatically calculated.
memSizePerCore	Optional. The memory size of each core.	Automatically calculated.

## 8.3.5 Statistical analysis

### 8.3.5.1 Data pivoting

This component allows you to view the distributions of feature values, feature columns, and label columns. Data can be analyzed more efficiently when you know its features. This component supports dense and sparse formats.

PAI command

```
PAI -name fe_meta_runner -project algo_public
-DinputTable="pai_dense_10_10"
-DoutputTable="pai_temp_2263_20384_1"
-DmapTable="pai_temp_2263_20384_2"
-DselectedCols="pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,age,campaign,poutcome"
-DlabelCol="y"
-DcategoryCols="previous"
-Dlifecycle="28"-DmaxBins="5" ;
```

Algorithm parameters

Parameter	Description	Required	Default value
inputTable	The name of the input table.	Yes	N/A

Parameter	Description	Required	Default value
<b>inputTablePartitions</b>	The partitions selected from the input table.	No	N/A
<b>outputTable</b>	The name of the output table.	Yes	N/A
<b>mapTable</b>	The output mapping table. The Data Pivoting component maps String and Int type data for machine learning to use for training.	Yes	N/A
<b>selectedCols</b>	The columns selected from the input table.	Yes	N/A
<b>categoryCols</b>	The columns specified to process Int or Double type columns as enumeration features.	No	null
<b>maxBins</b>	The maximum number of intervals for equal-distance division of continuous features.	No	100
<b>isSparse</b>	Indicates whether the features are in sparse format.	No	false
<b>itemSplitter</b>	The delimiter used to separate sparse feature items.	No	","
<b>kvSplitter</b>	The delimiter used to separate keys and values.	No	":"



Parameter	Description	Required	Default value
lifecycle	The lifecycle of the output table. Unit: days.	No	28

### 8.3.5.2 Whole table statistics

**This component analyzes a table or selected columns of a table.**

#### Parameter settings

**In the Input Columns box, select the columns of the table to be analyzed. By default, all columns are selected. You can enter filtering conditions for the selected columns in the condition text box. Operators available include the equal (=), not equal (!=), greater than (>), less than (<), greater than or equal to (>=), and less than or equal to (<=) signs, as well as like and rlike.**

#### PAI command

```
PAI -name SimpleSummary
-project algo_public
-DsummaryColNames="euribor3m,pdays"
-DoutputTableNames="pai_temp_667_6017_1"
-DinputTableName="bank_data"
```

```
-Dfilter="age>40";
```

Table 8-21: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <b>algo_public</b> . If you change the name, the system reports an error.
<b>summaryColNames</b>	The columns that require analysis. Separate the columns with commas(,).
<b>outputTableNames</b>	The names of the output tables generated after the system performs the whole table statistics operation.
<b>inputTableName</b>	The name of the input table.
<b>filter</b>	The filtering conditions. Operators available include the equal (=), not equal (!=), greater than (>), less than (<), greater than or equal to (>=), and less than or equal to (<=) signs, as well as like and rlike.

### 8.3.5.3 Correlation coefficient matrix

The correlation coefficient is a measure of the correlation between columns in a matrix. The valid range of values for this parameter is [-1, 1]. The count equals the number of non-zero elements in two successive columns.

Column settings

Fields Setting	Tuning
All Selected by Default	
<input type="text" value="Select Column"/>	

PAI command

```
PAI -name corrccoef
-project algo_public
-DinputTableName=maple_test_corrccoef_basic12x10_input
-DoutputTableName=maple_test_corrccoef_basic12x10_output
-DcoreNum=1
```

`-DmemSizePerCore=110`

Table 8-22: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	Table name	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions are selected by default.
<b>outputTableName</b>	<b>Required.</b> A list of output table names.	Table name	-
<b>selectedColNames</b>	<b>Optional.</b> The names of columns selected from the input table.	Column name	All columns are selected by default.
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	<b>Optional.</b> The number of cores.	This parameter is used with <b>memSizePerCore</b> . The value must be a positive integer in the range of [1, 9999].	Automatically calculated.
<b>memSizePerCore</b>	<b>Optional.</b> The memory size of each node. Unit : MB.	A positive integer in the range of [1024, 65536]	Automatically calculated.

## Examples

*Table 8-23: Data generation* describes the data generation result.

Table 8-23: Data generation

col0: double	col1: bigint	col2: double	col3: bigint	col4: double	col5: bigint	col6: double	col7: bigint	col8: double	col9: double
19	95	33	52	115	43	32	98	76	40
114	26	101	69	56	59	116	23	109	105
103	89	7	9	65	118	73	50	55	81
79	20	63	71	5	24	77	31	21	75
87	16	66	47	25	14	42	99	108	57
11	104	38	37	106	51	3	91	80	97
84	30	70	46	8	6	94	22	45	48
35	17	107	64	10	78	53	34	90	96
13	61	39	1	29	117	112	2	82	28
62	4	102	88	100	36	67	54	12	85
49	27	44	93	68	110	60	72	86	58
92	119	0	113	41	15	74	83	18	111

PAI command

```
PAI -name corrcoeff
-project algo_public
-DinputTableName=maple_test_corrcoeff_basic12x10_input
-DoutputTableName=maple_test_corrcoeff_basic12x10_output
-DcoreNum=1
-DmemSizePerCore=110
```

Output description

Table 8-24: Output table

columnsnames	col0	col1	col2
col0	1	-0.2115657251 820724	0.0598306259 706561
col1	-0.2115657251 820724	1	-0.8444477377 898585
col2	0.0598306259 706561	-0.8444477377 898585	1
col3	0.2599903570 684693	-0.1750763622 1594533	0.1851834664 7293102

columnsnames	col0	col1	col2
col4	-0.3483249188 225586	0.4094338415 0571377	-0.2093483922 8057014
col5	-0.2871625439 6809926	0.0913597602 6101403	-0.1896417512 389659
col6	0.4788016212 7435116	-0.3018506374 626574	0.1799377498 863213
col7	-0.1364651948 4213326	0.4073372691 2808044	-0.3858885676 469948
col8	-0.1950015876 4680092	-0.1182773912 4590071	0.2025456920 3773892
col9	0.3897390240 949085	0.1243385138 9455183	0.1347616075 3756655

### 8.3.5.4 Covariance

In probability theory and statistics, covariance is a measure of the joint variability of two random variables. Variance is a special case of covariance where the two measured variables are the same. If the expected values are  $E(X) = \mu$  and  $E(Y) = \nu$ , the covariance between real-number random variables  $X$  and  $Y$  is  $\text{cov}(X, Y) = E((X - \mu)(Y - \nu))$ .

PAI command

```
PAI -name cov
 -project algo_public
 -DinputTableName=maple_test_cov_basic12x10_input
 -DoutputTableName=maple_test_cov_basic12x10_output
 -DcoreNum=6
 -DmemSizePerCore=110;
```

Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-

Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>outputTableName</b>	<b>Required.</b> A list of output table names .	Table name	-
<b>selectedColNames</b>	<b>Optional.</b> The names of columns selected from the input table.	Column name	All columns are selected by default.
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	<b>Optional.</b> The number of cores.	This parameter is used with <code>memSizePerCore</code> . The value must be a positive integer in the range of [1, 9999].	Automatically calculated.
<b>memSizePerCore</b>	<b>Optional.</b> The memory size of each node. Unit: MB.	A positive integer in the range of [ 1024, 65536]	Automatically calculated.

### 8.3.5.5 Empirical probability density chart

An empirical distribution is an estimated non-parametric distribution of probability in scenarios where accurate parametric distributions cannot be made.

The algorithm uses kernel distribution to estimate the probability density of sample data. Similar to a histogram, the algorithm generates functions to describe the distribution of sample data. However, kernel distribution is different in that it overlays the contributions of all parts to generate a smooth and continuous distribution curve, while a histogram only generates discrete descriptions. When kernel distribution is used, the probability density of non-sample data points is not 0, but an overlay of weighted probability density of all sampling points in a certain kernel distribution. In this document, the kernel distribution used is Gaussian distribution.

- For more information about kernel distribution, see [Wikipedia](#).
- For more information about empirical distribution, see [Wikipedia](#).

PAI command

```
PAI -name empirical_pdf
-project algo_public
-DinputTableName="test_data"
-DoutputTableName="test_epdf_out"
-DfeatureColNames="col0,col1,col2"
-DinputTablePartitions="ds='20160101'"
-Dlifecycle=1
-DintervalNum=100
```

Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-
outputTableName	Required. The name of the output table.	Table name	-
featureColNames	Required. The names of input columns.	Multiple columns of the double or bigint type can be selected.	-



Parameter	Description	Valid values	Default value
labelColName	Optional. The name of the input label column. The feature column is stratified based on the label values in the label column.	Only one column of the bigint or string type can be selected. The number of label values cannot exceed 100.	No input label column is set by default.
inputTablePartitions	Optional. The partitions selected from the input table.	Partition name	All partitions are selected by default.
intervalNum	The number of calculation intervals. The larger the number, the higher the accuracy.	[1, 1E14)	Default value: -1. This value indicates that the number of intervals is determined based on the range of data values for each column.
lifecycle	The lifecycle of the output table.	A positive integer	Default value: -1. This value indicates that no lifecycle is set.
coreNum	Optional. The number of cores.	A positive integer	Default value: -1. This value indicates that the number of instances is determined based on the volume of input data.
memSizePerCore	Optional. The memory size of each core.	A positive integer in the range of [1024, 65536]	Default value: -1. This value indicates that the memory size is determined based on the volume of input data.

## Examples

### SQL statement to generate data:

```
drop table if exists epdf_test;
create table epdf_test as
select
 *
from
(
 select 1.0 as col1 from dual
 union all
 select 2.0 as col1 from dual
 union all
 select 3.0 as col1 from dual
 union all
 select 4.0 as col1 from dual
 union all
 select 5.0 as col1 from dual
) tmp;
```

### PAI command

```
PAI -name empirical_pdf
 -project algo_public
 -DinputTableName=epdf_test
 -DoutputTableName=epdf_test_out
 -DfeatureColNames=col1;
```

### Input description

You can select multiple columns to be calculated. You can select a label column and stratify the columns by label. For example, if the label column contains labels 0 and 1, the columns that need to be calculated are stratified into two groups. One group only contains columns with label 0 and the other group only contains columns with label 1. The probability density for each group is then calculated. If no label column is selected, all feature columns are calculated.

### Output description

This component outputs a diagram and a result table. The columns in the result table are as follows. If no label column is selected, NULL is output in the label column.

Column name	Data type
colName	string
label	string
x	double
pdf	double

**Output table:**

	colname	label	x	pdf
	col1	NULL	1.0	0.12775155176809325
829622	col1	NULL	1.0404050505050506	0.1304256933
7429525	col1	NULL	1.0808101010101012	0.1330632589
616418	col1	NULL	1.1212151515151518	0.1356613897
574596	col1	NULL	1.1616202020202024	0.1382173796
	col1	NULL	1.202025252525253	0.1407286844875733
4274642	col1	NULL	1.2424303030303037	0.1431929301
0033242	col1	NULL	1.2828353535353543	0.1456079196
6379316	col1	NULL	1.3232404040404049	0.1479716387
772349	col1	NULL	1.3636454545454555	0.1502822610
	col1	NULL	1.404050505050506	0.1525381508819247
919243	col1	NULL	1.4444555555555567	0.1547378654
764068	col1	NULL	1.4848606060606073	0.1568801559
4681753	col1	NULL	1.525265656565658	0.1589639666
5768245	col1	NULL	1.5656707070707085	0.1609884332
404685	col1	NULL	1.6060757575757592	0.1629528799
0034038	col1	NULL	1.6464808080808098	0.1648568149
1584543	col1	NULL	1.6868858585858604	0.1666999249
9138338	col1	NULL	1.727290909090911	0.1684820686
2168932	col1	NULL	1.7676959595959616	0.1702032691
3638117	col1	NULL	1.8081010101010122	0.1718637045
0080946	col1	NULL	1.8485060606060628	0.1734636990
5692428	col1	NULL	1.8889111111111134	0.1750037117
9456017	col1	NULL	1.929316161616164	0.1764843258
4938396	col1	NULL	1.9697212121212146	0.1779062363
286898	col1	NULL	2.0101262626262653	0.1792702373
7022053	col1	NULL	2.050531313131316	0.1805772092
4221673	col1	NULL	2.0909363636363665	0.1818281054
9491406	col1	NULL	2.131341414141417	0.1830239382

7472337	col1	NULL	2.1717464646464677	0.1841657656
123305	col1	NULL	2.2121515151515183	0.1852546770
9496213	col1	NULL	2.252556565656569	0.1862917795
3109434	col1	NULL	2.2929616161616195	0.1872781850
	col1	NULL	2.333366666666667	0.18821499601297229
7850022	col1	NULL	2.3737717171717208	0.1891032934
6940221	col1	NULL	2.4141767676767714	0.1899441242
7711185	col1	NULL	2.454581818181822	0.1907384893
6168018	col1	NULL	2.4949868686868726	0.1914873328
	col1	NULL	2.535391919191923	0.1921915315221827
8972659	col1	NULL	2.575796969696974	0.1928518853
0630113	col1	NULL	2.6162020202020244	0.1934691091
4446043	col1	NULL	2.656607070707075	0.1940438242
142701	col1	NULL	2.6970121212121256	0.1945765526
9517916	col1	NULL	2.7374171717171762	0.1950677105
2158667	col1	NULL	2.777822222222227	0.1955176045
4194602	col1	NULL	2.8182272727272775	0.1959264271
	col1	NULL	2.858632323232328	0.1962942551623821
770638	col1	NULL	2.8990373737373787	0.1966210478
790639	col1	NULL	2.9394424242424293	0.1969066468
	col1	NULL	2.9798474747474748	0.19715077683721793
1663747	col1	NULL	3.0202525252525305	0.1973530473
1309964	col1	NULL	3.0606575757575781	0.1975129556
6457925	col1	NULL	3.1010626262626317	0.1976298905
9675995	col1	NULL	3.1414676767676823	0.1977031372
5349683	col1	NULL	3.181872727272733	0.1977318828
5793107	col1	NULL	3.2222777777777836	0.1977152226
4530828	col1	NULL	3.262682828282834	0.1976521677
0453194	col1	NULL	3.303087878787885	0.1975416527
6210697	col1	NULL	3.3434929292929354	0.1973825442
3938664	col1	NULL	3.383897979797986	0.1971736504
1193162	col1	NULL	3.4243030303030366	0.1969137302

982942	col1	NULL	3.4647080808080872	0.1966015035
0464843	col1	NULL	3.505113131313138	0.1962356621
5135703	col1	NULL	3.5455181818181885	0.1958148794
0778076	col1	NULL	3.585923232323239	0.1953378225
623475	col1	NULL	3.6263282828282897	0.1948031623
560816	col1	NULL	3.6667333333333403	0.1942095854
0939734	col1	NULL	3.707138383838391	0.1935558047
7394655	col1	NULL	3.7475434343434415	0.1928405705
4364004	col1	NULL	3.787948484848492	0.1920626819
6158253	col1	NULL	3.8283535353535427	0.1912209968
6253852	col1	NULL	3.8687585858585933	0.1903144429
	col1	NULL	3.909163636363644	0.1893420275936375
5928747	col1	NULL	3.9495686868686946	0.1883028475
	col1	NULL	3.989973737373745	0.1871960984396676
3567092	col1	NULL	4.030378787878796	0.1860210834
9674377	col1	NULL	4.070783838383846	0.1847772216
	col1	NULL	4.111188888888897	0.1834640560916829
	col1	NULL	4.151593939393948	0.1820812603860928
9383914	col1	NULL	4.191998989898998	0.1806286457
	col1	NULL	4.232404040404049	0.179106166873458
4406796	col1	NULL	4.272809090909099	0.1775139267
	col1	NULL	4.31321414141415	0.17585218159888508
9794325	col1	NULL	4.353619191919201	0.1741213444
	col1	NULL	4.394024242424251	0.1723219884250765
9762067	col1	NULL	4.434429292929302	0.1704548485
2064342	col1	NULL	4.4748343434343525	0.1685208240
	col1	NULL	4.515239393939403	0.1665209782808102
7824907	col1	NULL	4.555644444444454	0.1644565395
9798905	col1	NULL	4.596049494949504	0.1623288999
2571825	col1	NULL	4.636454545454555	0.1601396140
157465	col1	NULL	4.6768595959596055	0.1578903963
2216193	col1	NULL	4.717264646464656	0.1555831187

	col1	NULL	4.757669696969707	0.1532198066072439
	col1	NULL	4.798074747474757	0.1508026344442397
	col1	NULL	4.838479797979808	0.1483339207
3462115	col1	NULL	4.878884848484859	0.1458161222
6291346	col1	NULL	4.919289898989909	0.1432518277151203
	col1	NULL	4.95969494949496	0.1406437506896507
	col1	NULL	5.000100000000001	0.13799472213247665
+-----+-----+-----+-----+				

Input and output restrictions

**The maximum number of label columns that can be specified is 100.**

### 8.3.5.6 Chi-square goodness of fit test

**This component is used to determine the differences between the observed frequencies and the expected frequencies for each classification of a single multiclass classification nominal variable. The null hypothesis assumes that the observed frequencies and the expected frequencies are consistent.**

PAI command

```
PAI -name chisq_test
 -project algo_public
 -DinputTableName=pai_chisq_test_input
 -DcolName=f0
 -DprobConfig=0:0.3,1:0.7
 -DoutputTableName=pai_chisq_test_output0
 -DoutputDetailTableName=pai_chisq_test_output0_detail
```

Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	<b>Table name</b>	-
<b>colName</b>	<b>Required. The name of the column that requires a chi-square test.</b>	<b>Column name</b>	-
<b>outputTableName</b>	<b>Required. The name of the output table.</b>	<b>Table name that has not been used</b>	-

Parameter	Description	Valid values	Default value
<b>outputDetailTableName</b>	<b>Required.</b> The name of the output detail table.	Table name that has not been used	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	Partition list	All partitions are selected by default.
<b>probConfig</b>	<b>Optional.</b> The class probability configuration.	The configuration is stored in a key-value pair format: <code>class:probability</code> . The sum of all probabilities is 1.	All classes have the same probability by default.

## Examples

### Testing data

```
create table pai_chisq_test_input as
select * from
(
 select '1' as f0,'2' as f1 from dual
 union all
 select '1' as f0,'3' as f1 from dual
 union all
 select '1' as f0,'4' as f1 from dual
 union all
 select '0' as f0,'3' as f1 from dual
 union all
 select '0' as f0,'4' as f1 from dual
)tmp;
```

### PAI command

```
PAI -name chisq_test
 -project algo_public
 -DinputTableName=pai_chisq_test_input
 -DcolName=f0
 -DprobConfig=0:0.3,1:0.7
 -DoutputTableName=pai_chisq_test_output0
 -DoutputDetailTableName=pai_chisq_test_output0_detail
```


### Output description

Output table **outputTableName** is a JSON array containing only one row and one column.

```
{
 "Chi-Square": {
 "comment": "Pearsons chi-square test",
```

```
"df": 1,
"p-value": 0.75,
"value": 0.2380952380952381
}
```

Output table `outputDetailTableName` includes the following columns: data source class (f0 or f1), observed frequency (observed), expected frequency (expected), and standard residuals ( $\text{residuals} = (\text{observed} - \text{expected}) / \sqrt{\text{expected}}$ ).

f0	f1		observed	expected	residuals
0	2		0.0	0.4	-0.6324555320336759
0	3		1.0	0.8	0.22360679774997894
0	4		1.0	0.8	0.22360679774997894
1	2		1.0	0.6000000000000001	0.5163977794943221
1	3		1.0	1.2000000000000002	-0.1825741858350555
1	4		1.0	1.2000000000000002	-0.1825741858350555

### 8.3.5.7 Chi-square test of independence

This component verifies whether two factors (each having two or more classes) are mutually independent. The null hypothesis is that two factors are independent of each other.

PAI command

```
PAI -name chisq_test
-project algo_public
-DinputTableName=pai_chisq_test_input
-DxColName=f0
-DyColName=f1
-DoutputTableName=pai_chisq_test_output2
-DoutputDetailTableName=pai_chisq_test_output2_detail
```

Parameters

Parameter	Description	Valid values	Default value
<code>inputTableName</code>	Required. The name of the input table.	Table name	-



Parameter	Description	Valid values	Default value
<b>xColName</b>	<b>Required.</b> The name of the column that requires a chi-square test.	<b>Column name</b>	-
<b>yColName</b>	<b>Required.</b> The name of the column that require a chi-square test.	<b>Column name</b>	-
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	<b>Table name that has not been used</b>	-
<b>outputDetailTableName</b>	<b>Required.</b> The name of the output detail table.	<b>Table name that has not been used</b>	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	<b>Partition list</b>	<b>All partitions are selected by default.</b>
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>

## Examples

### • Testing data

```
create table pai_chisq_test_input as
select * from
(
select '1' as f0,'2' as f1 from dual
union all
select '1' as f0,'3' as f1 from dual
union all
select '1' as f0,'4' as f1 from dual
union all
select '0' as f0,'3' as f1 from dual
union all
select '0' as f0,'4' as f1 from dual
)tmp;
```

### • PAI command

```
PAI -name chisq_test
 -project algo_public
```

```
-DinputTableName=pai_chisq_test_input
-DxColName=f0
-DyColName=f1
-DoutputTableName=pai_chisq_test_output2
-DoutputDetailTableName=pai_chisq_test_output2_detail
```

• **Output description**


**Output table outputTableName is a JSON array containing only one row and one column.**

```
{
 "Chi-Square": {
 "comment": "Pearsons chi-square test",
 "df": 2,
 "p-value": 0.75,
 "value": 0.8333333333333334
 }
}
```

**Output table outputDetailTableName has the following columns:**

Column name	Description
xColName	Class
yColName	Class
observed	Observed frequency
expected	Expected frequency
residuals	Residuals = (observed - expected)/sqrt (expected)

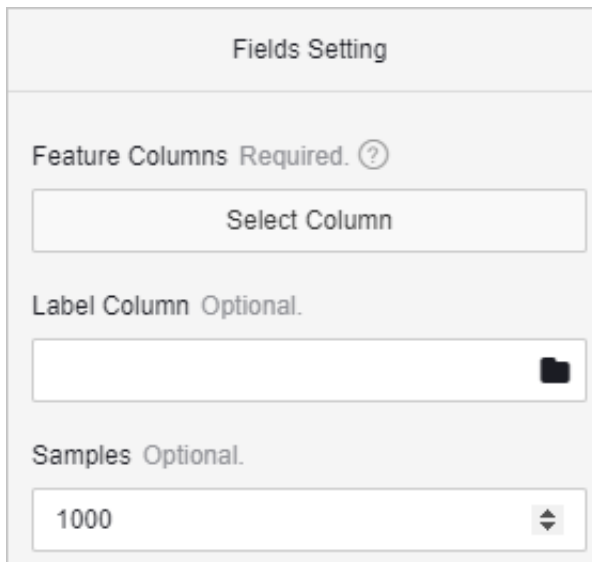
**Data:**

f0	f1		observed	expected	residuals
0	2		0.0	0.4	-0.6324555320336759
0	3		1.0	0.8	0.22360679774997894
0	4		1.0	0.8	0.22360679774997894
1	2		1.0	0.6000000000000001	0.5163977794943221
1	3		1.0	1.2000000000000002	-0.1825741858350555
1	4		1.0	1.2000000000000002	-0.1825741858350555

### 8.3.5.8 Scatter plot

In regression analysis, this component outputs a scatter plot that shows the distribution of data points in a Cartesian coordinate system.

Column settings



The 'Fields Setting' dialog box contains three sections: 'Feature Columns Required. (?)' with a 'Select Column' button; 'Label Column Optional.' with a text input field and a folder icon; and 'Samples Optional.' with a text input field containing '1000' and a spinner icon.

PAI command

```
PAI -name scatter_diagram
-project algo_public
-DselectedCols=emp_var_rate,cons_price_rate,cons_conf_idx,euribor3m
-DsampleSize=1000
-DlabelCol=y
-DmapTable=pai_temp_2447_22859_2
-DinputTable=scatter_diagram
-DoutputTable=pai_temp_2447_22859_1
```

Table 8-25: Parameters

Parameter	Description	Default value
<b>inputTable</b>	<b>Required.</b> The name of the input table.	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	-
<b>outputTable</b>	<b>Required.</b> The name of the output table that stores the samples.	-
<b>mapTable</b>	<b>Required.</b> The name of the output table that stores the maximum value , minimum value, and enumeration values of each feature.	-

Parameter	Description	Default value
<b>selectedCols</b>	<b>Required. The columns selected from the input table from which to draw a scatter plot. A maximum of five features can be selected.</b>	-
<b>labelCol</b>	<b>Optional. An Int or String column to serve as the enumeration label column.</b>	<b>No enumeration label column is set by default.</b>
<b>sampleSize</b>	<b>Optional. The number of samples to collect from the input data.</b>	1000
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table measured in days.</b>	28

## Examples

**Input data**

```
create table scatter_diagram as select emp_var_rate,cons_price_rate,
cons_conf_idx,euribor3m,y from pai_bank_data limit 10
```

Table 8-26: Parameters

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
1.4	93.918	-42.7	4.962	0
-0.1	93.2	-42.0	4.021	0
-1.7	94.055	-39.8	0.729	1
-1.8	93.075	-47.1	1.405	0
-2.9	92.201	-31.4	0.869	1
1.4	93.918	-42.7	4.961	0
-1.8	92.893	-46.2	1.327	0
-1.8	92.893	-46.2	1.313	0
-2.9	92.963	-40.8	1.266	1
-1.8	93.075	-47.1	1.41	0
1.1	93.994	-36.4	4.864	0
1.4	93.444	-36.1	4.964	0
1.4	93.444	-36.1	4.965	1

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
-1.8	92.893	-46.2	1.291	0
1.4	94.465	-41.8	4.96	0
1.4	93.918	-42.7	4.962	0
-1.8	93.075	-47.1	1.365	1
-0.1	93.798	-40.4	4.86	1
1.1	93.994	-36.4	4.86	0
1.4	93.918	-42.7	4.96	0
-1.8	93.075	-47.1	1.405	0
1.4	94.465	-41.8	4.967	0
1.4	93.918	-42.7	4.963	0
1.4	93.918	-42.7	4.968	0
1.4	93.918	-42.7	4.962	0
-1.8	92.893	-46.2	1.344	0
-3.4	92.431	-26.9	0.754	0
-1.8	93.075	-47.1	1.365	0
-1.8	92.893	-46.2	1.313	0
1.4	93.918	-42.7	4.961	0
1.4	94.465	-41.8	4.961	0
-1.8	92.893	-46.2	1.327	0
-1.8	92.893	-46.2	1.299	0
-2.9	92.963	-40.8	1.268	1
1.4	93.918	-42.7	4.963	0
-1.8	92.893	-46.2	1.334	0
1.4	93.918	-42.7	4.96	0
-1.8	93.075	-47.1	1.405	0
1.4	94.465	-41.8	4.96	0
1.4	93.444	-36.1	4.962	0
1.1	93.994	-36.4	4.86	0
1.1	93.994	-36.4	4.857	0

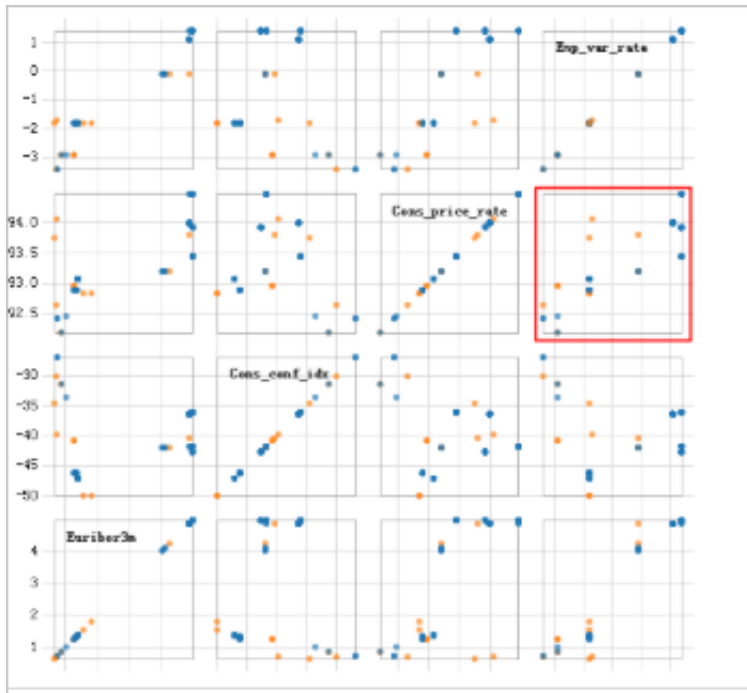
emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
1.4	93.918	-42.7	4.961	0
-3.4	92.649	-30.1	0.715	1
1.4	93.444	-36.1	4.966	0
-0.1	93.2	-42.0	4.076	0
1.4	93.444	-36.1	4.965	0
-1.8	92.893	-46.2	1.354	0
1.4	93.444	-36.1	4.967	0
1.4	94.465	-41.8	4.959	0
-1.8	92.893	-46.2	1.354	0
1.4	94.465	-41.8	4.958	0
-1.8	92.893	-46.2	1.354	0
1.4	94.465	-41.8	4.864	0
1.1	93.994	-36.4	4.859	0
1.1	93.994	-36.4	4.857	0
-1.8	92.893	-46.2	1.27	0
1.1	93.994	-36.4	4.857	0
1.1	93.994	-36.4	4.859	0
1.4	94.465	-41.8	4.959	0
1.1	93.994	-36.4	4.856	0
-1.8	93.075	-47.1	1.405	0
-1.8	92.843	-50.0	1.811	1
-0.1	93.2	-42.0	4.021	0
-2.9	92.469	-33.6	1.029	0
1.4	93.918	-42.7	4.962	0
-1.8	93.075	-47.1	1.365	0
1.1	93.994	-36.4	4.857	0
-1.8	92.893	-46.2	1.259	0
1.1	93.994	-36.4	4.857	0
1.4	94.465	-41.8	4.866	0

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
-2.9	92.201	-31.4	0.883	0
-0.1	93.2	-42.0	4.076	0
1.1	93.994	-36.4	4.857	0
1.4	93.918	-42.7	4.96	0
1.4	93.444	-36.1	4.962	0
1.1	93.994	-36.4	4.858	0
1.1	93.994	-36.4	4.857	0
1.1	93.994	-36.4	4.856	0
1.4	93.918	-42.7	4.968	0
1.4	93.444	-36.1	4.966	0
1.4	94.465	-41.8	4.962	0
1.4	93.444	-36.1	4.963	0
-1.8	92.843	-50.0	1.56	1
1.4	93.918	-42.7	4.96	0
1.4	93.444	-36.1	4.963	0
-3.4	92.431	-26.9	0.74	0
1.1	93.994	-36.4	4.856	0
1.4	93.918	-42.7	4.962	0
1.1	93.994	-36.4	4.856	0
-0.1	93.2	-42.0	4.245	1
1.1	93.994	-36.4	4.857	0
-1.8	93.075	-47.1	1.405	0
-1.8	92.893	-46.2	1.327	0
-0.1	93.2	-42.0	4.12	0
1.4	94.465	-41.8	4.958	0
-1.8	93.749	-34.6	0.659	1
1.1	93.994	-36.4	4.858	0
1.1	93.994	-36.4	4.858	0
1.4	93.444	-36.1	4.963	0

## Parameter settings

**Scatter plot configuration:** select `emp_var_rate`, `cons_price_rate`, `cons_conf_idx`, and `euribor3m` as the feature columns, and select `y` as the label column.

Figure 8-2: Output



You can view the distribution of classification tags between every two features in the scatter plot.

### 8.3.5.9 Two-sample T-test

A two-sample T-test is composed of an independent sample T-test and a paired sample T-test. Two samples independent of each other are called independent samples. An independent sample T-test checks whether two samples are significantly different from each other. The T-test is based on the premise that two samples are independent of each other and come from two normally distributed populations. A paired sample T-test checks whether the mean values from two paired populations are significantly different from each other.

## PAI command

```
PAI -name t_test
 -project algo_public
 -DxTableName=pai_t_test_all_type
 -DxColName=col1_double
 -DxTablePartitions=ds=2010/dt=1
```



```
-DyTableName=pai_t_test_all_type
-DyColName=col1_double
-DyTablePartitions=ds=2010/dt=1
-DoutputTableName=pai_t_test_out
-Dalternative=less
-Dmu=47
-DconfidenceLevel=0.95
-Dpaired=False
-DvarEqual=True
```

## Parameters

Parameter	Description	Valid values	Default value
<b>xTableName</b>	<b>Required.</b> The name of input table x.	Table name	-
<b>xTablePartitions</b>	<b>Optional.</b> The partitions selected from input table x for testing, in the format of <code>Partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>xColName</b>	<b>Required.</b> The column selected from table x for testing.	Column name. The type must be double or bigint.	-
<b>yTableName</b>	<b>Required.</b> The name of input table y.	Table name	-
<b>yTablePartitions</b>	<b>Optional.</b> The partitions selected from input table y for testing, in the format of <code>Partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>yColName</b>	<b>Required.</b> The name of the column selected from table y for testing.	Column name. The type must be double or bigint.	-
<b>paired</b>	<b>Optional.</b> A value of True indicates that it is a paired sample T-test. A value of False indicates that it is an independent sample T-test.	True and False	False

Parameter	Description	Valid values	Default value
<b>alternative</b>	<b>Optional. The alternative hypothesis.</b>	<b>two.sided, less, and greater</b>	<b>two.sided</b>
<b>mu</b>	<b>Optional. The hypothesized mean.</b>	<b>double</b>	<b>0</b>
<b>varEqual</b>	<b>Optional. This parameter indicates whether two population variances are equal.</b>	<b>True and False</b>	<b>False</b>
<b>confidenceLevel</b>	<b>Optional. The confidence level.</b>	<b>0.8, 0.9, 0.95, 0.99, 0.995, and 0.999</b>	<b>0.95</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>This parameter is used with memSizePerCore. The value must be a positive integer in the range of [1, 9999].</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each node. Unit: MB.</b>	<b>A positive integer in the range of [1024, 65536].</b>	<b>Automatically calculated.</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>

#### Custom resources



#### Note:

- For a regular table, we recommend that you do not set **coreNum** and **memSizePerCore**, and instead allow the default values to be used automatically.

- **If you do not have sufficient compute resources, use the following code to calculate the amount of compute resources needed:**

```
def CalcCoreNumAndMem(row, centerCount, kOneCoreDataSize=1024):
 """Calculates the number of nodes and memory size needed for each
 node.
 Args:
 row: the number of rows in the input table.
 col: the number of columns in the input table.
 kOneCoreDataSize: the amount of data needed to be
 calculated per node. Unit: MB. The value must be a positive integer.
 Default value: 1024.
 Return:
 coreNum, memSizePerCore
 Example:
 coreNum, memSizePerCore = CalcCoreNumAndMem(1000,99, 100,
 kOneCoreDataSize=2048)

 """
 kMBytes = 1024.0 * 1024.0
 #Number of compute nodes
 coreNum = max(1, int(row * 2 * 8 / kMBytes / kOneCoreDataSize))
 #Memory size per node = Data volume
 memSizePerCore = max(1024, int(kOneCoreDataSize*2))
 return coreNum, memSizePerCore
```

#### Examples

- **SQL statement to generate data:**

```
create table pai_test_input as
select * from
(
 select 1 as f0,2 as f1 from dual
 union all
 select 1 as f0,3 as f1 from dual
 union all
 select 1 as f0,4 as f1 from dual
 union all
 select 0 as f0,3 as f1 from dual
 union all
 select 0 as f0,4 as f1 from dual
)tmp;
```

- **PAI command**

```
PAI -name t_test
 -project algo_public
 -DxTableName=pai_test_input
 -DxColName=f0
 -DyTableName=pai_test_input
 -DyColName=f1
 -DyTablePartitions=ds=2010/dt=1
 -DoutputTableName=pai_t_test_out
 -Dalternative=less
 -Dmu=47
 -DconfidenceLevel=0.95
 -Dpaired=False
```

```
-DvarEqual=True
```

- **Output description**

**The output table is a JSON array containing only one row and one column.**

```
{
 "AlternativeHypthesis": "difference in means not equals to 0",
 "ConfidenceInterval": "(-2.5465, -0.4535)",
 "ConfidenceLevel": 0.95,
 "alpha": 0.050000000000000004,
 "df": 19,
 "mean of the differences": -1.5,
 "p": 0.0080000000000000007,
 "t": -3
}
```

Input and output restrictions

**The input and output are not limited.**

### 8.3.5.10 One-sample T-test

**A one-sample T-test verifies whether the mean of a normally distributed population differs significantly from a target value. A T-test is performed based on the condition that the sample population is normally distributed.**

PAI command

```
PAI -name t_test -project algo_public
-DxTableName=pai_t_test_all_type
-DxColName=col1_double
-DoutputTableName=pai_t_test_out
-DxTablePartitions=ds=2010/dt=1
-Dalternative=less
-Dmu=47
-DconfidenceLevel=0.95
```

Algorithm parameters

Parameter	Description	Valid values	Default value
xTableName	Required. The name of input table x.	Table name	-
xColName	Required. The column selected from table x for testing.	Column name. The type must be double or bigint.	-
outputTableName	Required. The name of the output table.	Table name that has not been used	-

Parameter	Description	Valid values	Default value
<b>xTablePartitions</b>	<b>Optional. The partitions selected from input table x.</b>	<b>Partition list</b>	<b>All partitions are selected by default.</b>
<b>alternative</b>	<b>Optional. The alternative hypothesis.</b>	<b>two.sided, less, and greater</b>	<b>two.sided</b>
<b>mu</b>	<b>Optional. The hypothesized mean .</b>	<b>double</b>	<b>0</b>
<b>confidenceLevel</b>	<b>Optional. The confidence level.</b>	<b>0.8, 0.9, 0.95, 0.99, 0.995, and 0.999</b>	<b>0.95</b>

Output description

**The output table is a JSON array containing only one row and one column.**

```
{
 "AlternativeHypthesis": "mean not equals to 0",
 "ConfidenceInterval": "(44.72234194006504, 46.27765805993496)",
 "ConfidenceLevel": 0.95,
 "alpha": 0.05,
 "df": 99,
 "mean": 45.5,
 "p": 0,
 "stdDeviation": 3.919647479510927,
 "t": 116.081867662439
}
```

### 8.3.5.11 Lorenz curve

The Lorenz curve is a graph to illustrate the distribution of wealth across a population. The X axis represents the total population arranged from least wealthy to most wealthy, while the Y axis represents the total wealth. If this graph is a straight line, it indicates perfectly equal distribution of wealth. The Gini coefficient is calculated by taking the area between the equal distribution curve and the actual Lorenz curve for a population as a fraction of the total area beneath the equal distribution curve. As the distribution of wealth becomes less equal, the Gini coefficient will increase, whereas a population with equal distribution of wealth will have a Gini coefficient of 0.

To study the distribution of income among a population, American statistician Max Otto Lorenz proposed the famous Lorenz curve in 1905. In 1921, Italian economist

**Corrado Gini defined the Gini coefficient as a measure of inequality in a population based on the Lorenz curve.**

PAI command

```
PAI -name LorenzCurve
 -project algo_public
 -DinputTableName=maple_test_lorenz_basic10_input
 -DcolName=col0
 -DoutputTableName=maple_test_lorenz_basic10_output -DcoreNum=20
 -DmemSizePerCore=110;
```

Parameters

Parameter	Description	Valid value	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	Table name	N/A
<b>outputTableName</b>	<b>Required. The name of the output table.</b>	Table name that has not been used	N/A
<b>colName</b>	<b>Optional. The column name. Separate multiple columns with commas (,).</b>	Column name	The whole table is selected by default.
<b>N</b>	<b>The number of quantiles.</b>	N/A	100
<b>inputPartitions</b>	<b>Optional. The partitions selected from the input table for training, in the partition_name=value format. To specify multiple partitions, use the following format: name1=value1/name2=value2. Separate multiple partitions with commas (,).</b>	Partition name	All partitions are selected by default.

Parameter	Description	Valid value	Default value
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>This parameter is used with memSizePerCore. The value must be a positive integer in the range of [1, 9999].</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB</b>	<b>A positive integer in the range of [ 1024, 65536]</b>	<b>Automatically calculated.</b>

## Examples

### Data generation

<b>col0:double</b>
4
7
2
8
6
3
9
5
0
1
10

### PAI command

```
PAI -name LorenzCurve
 -project algo_public
 -DinputTableName=maple_test_lorenz_basic10_input
 -DcolName=col0
 -DoutputTableName=maple_test_lorenz_basic10_output
```

```
-DcoreNum=20
-DmemSizePerCore=110;
```

## Output

Quantile	col0
0	0
1	0.01818181818181818
2	0.01818181818181818
3	0.01818181818181818
4	0.01818181818181818
5	0.01818181818181818
6	0.01818181818181818
7	0.01818181818181818
8	0.01818181818181818
9	0.01818181818181818
10	0.01818181818181818
11	0.05454545454545454
12	0.05454545454545454
13	0.05454545454545454
14	0.05454545454545454
...	...
85	0.8181818181818182
86	0.8181818181818182
87	0.8181818181818182
88	0.8181818181818182
89	0.8181818181818182
90	1
91	1
92	1
93	1
94	1
95	1



Quantile	col0
96	1
97	1
98	1
99	1
100	1

### 8.3.5.12 Normality test

This component is used to determine whether observed values are normally distributed.

This component consists of three test methods: Anderson-Darling test (see [Wikipedia](#)), Kolmogorov-Smirnov test (see [Wikipedia](#)), and Q-Q plot (see [Wikipedia](#)). You can use one or more methods as needed.

Algorithm description:

- Original hypothesis H0: The observed values are normally distributed. H1: The observed values are not normally distributed.
- The KS p-value calculation method progressively calculates CDF of KS distribution regardless of the sample size. For more information, see [Wikipedia](#).
- If the sample size is greater than 1000, the Q-Q plot method collects samples to calculate and output plots. This means that the data points in plots do not necessarily cover all samples.

PAI command

```
PAI -name normality_test
 -project algo_public
 -DinputTableName=test
 -DoutputTableName=test_out
 -DselectedColNames=col1,col2
 -Dlifecycle=1;
```

Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-

Parameter	Description	Valid values	Default value
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	<b>Table name that has not been used</b>	-
<b>selectedColNames</b>	<b>Optional.</b> The names of selected columns.	<b>Multiple double or bigint type columns can be selected.</b>	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	<b>Partition name</b>	<b>All partitions are selected by default.</b>
<b>enableQQplot</b>	<b>Optional.</b> This parameter specifies whether to use the Q-Q plot.	<b>true and false</b>	<b>true</b>
<b>enableADtest</b>	<b>Optional.</b> This parameter specifies whether to perform the Anderson-Darling test.	<b>true and false</b>	<b>true</b>
<b>enableKStest</b>	<b>Optional.</b> This parameter specifies whether to perform the Kolmogorov-Smirnov test.	<b>true and false</b>	<b>true</b>
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	<b>An integer greater than or equal to -1</b>	<b>Default value: -1. This value indicates that no lifecycle is set.</b>
<b>coreNum</b>	<b>Optional.</b> The number of cores.	<b>An integer greater than 0</b>	<b>Default value : -1. This value indicates that the number of instances is determined by the amount of input data.</b>

Parameter	Description	Valid values	Default value
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>(100, 65536)</b>	<b>Default value: -1. This value indicates that the memory size is determined by the amount of input data.</b>

## Examples

- **SQL statement to generate data:**

```
drop table if exists normality_test_input;
create table normality_test_input as
select
 *
from
(
 select 1 as x from dual
 union all
 select 2 as x from dual
 union all
 select 3 as x from dual
 union all
 select 4 as x from dual
 union all
 select 5 as x from dual
 union all
 select 6 as x from dual
 union all
 select 7 as x from dual
 union all
 select 8 as x from dual
 union all
 select 9 as x from dual
 union all
 select 10 as x from dual
) tmp;
```

- **PAI command**

```
PAI -name normality_test
 -project projectxlib4
 -DinputTableName=normality_test_input
 -DoutputTableName=normality_test_output
 -DselectedColNames=x
 -Dlifecycle=1;
```

- **Input description**

**Input format:** select the columns that need to be calculated. The columns must be of the double or bigint type.

## • Output description

A diagram and a result table are output. The columns in the result table are as follows. The result table has two partitions:

- `p='test'` shows the result of the AD or KS test. Data is output when `enableADtest` or `enableKStest` is set to true.
- `p='plot'` shows the Q-Q plot data. When `enableQQplot` is set to true, data is output and the columns that meet the `p='test'` condition are reused. In the case of `p='plot'`, the `testvalue` column records the original observed data (x axis of the Q-Q plot), and the `pvalue` column records the expected data that is normally distributed (y axis of the Q-Q plot).

Output table:

colname	testname	testvalue	pvalue	p
x	NULL	1.0	0.8173291742279805	plot
x	NULL	2.0	2.470864450785345	plot
x	NULL	3.0	3.5156067948020056	plot
x	NULL	4.0	4.3632330349313095	plot
x	NULL	5.0	5.128868067945126	plot
x	NULL	6.0	5.871131932054874	plot
x	NULL	7.0	6.6367669650686905	plot
x	NULL	8.0	7.4843932051979944	plot
x	NULL	9.0	8.529135549214654	plot
x	NULL	10.0	10.182670825772018	plot
x	Anderson_Darling_Test	0.1411092332197832	0.	
9566579606430077	test			
x	Kolmogorov_Smirnov_Test	0.09551932503797644	0.	
9999888659426232	test			

Column name	Data type	Definition
colName	string	Column name
testname	string	Test name
testvalue	double	Test value on the x axis of the Q-Q plot

Column name	Data type	Definition
pvalue	double	Test p value on the y axis of the Q-Q plot
p	double	Partition name

### 8.3.5.13 Percentile

This component calculates the percentile of the values in a column.

Parameter settings

Select the column to be analyzed. Only the double and bigint types are supported.

PAI command

```
PAI -name Percentile
-project algo_public
-DoutputTableName="pai_temp_666_6014_1"
-DcolName="euribor3m"
-DinputTableName="bank_data";
```

Table 8-27: Parameters

Parameter	Description
name	The name of the component.
project	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is algo_public. If you change the name, the system reports an error.
outputTableName	The name of the output table automatically generated after the system performs the percentile calculation.
colName	The column selected for percentile calculation. Only the numeric type is supported.
inputTableName	The name of the input table.

### 8.3.5.14 Pearson coefficient

This component calculates the Pearson correlation coefficient of two numeric columns in an input table or a partition, and saves the result to the output table.

Component description

- **The component has only two parameters:** input column 1 and input column 2. Enter the names of the two columns for which the Pearson correlation coefficient is calculated.

- After you run the component, right-click the component and choose View Analytics Report from the shortcut menu.
- The Pearson correlation coefficient is listed in the row.

PAI command

```
pai -name pearson
-project algo_test
-DinputTableName=wpbc
-Dcol1Name=f1
-Dcol2Name=f2
-DoutputTableName=wpbc_pear;
```

Algorithm parameters

Table 8-28: Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-
inputTablePartitions	The partitions selected from the input table for calculation.	The parameter value must be in the <code>partition_name=value</code> format. To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	All partitions in the input table are selected by default.
col1Name	Required. The name of input column 1.	Column name	-
col2Name	Required. The name of input column 2.	Column name	-
outputTableName	Required. The name of the output table.	Table name	-

### 8.3.5.15 Histogram

**This component analyzes data in a column and outputs a histogram.**

Parameter settings

- **Select the columns to be analyzed. Only the double and bigint types are supported.**
- **View the analysis report.**
- **You can adjust the step and move the slider to view the entire histogram.**

## 8.3.6 Machine learning

### 8.3.6.1 Binary classification

#### 8.3.6.1.1 GBDT binary classification

**This component is used for binary classification based on GBDT regression and sorting. Values greater than the threshold value are considered positive samples, while values that are less than or equal to the threshold value are considered negative samples.**

Procedure

1. **Drag and drop the GBDT Binary Classification component onto the canvas for training and set the parameters, as shown in the following figure.**

Table 8-29: Parameters

Parameter	Description
Feature Columns	The double and bigint types are supported. A maximum of 800 columns can be specified.
Label Column	You can select all columns except the input column. The values must be of the binary type.
Stratification Column	Optional. The whole table is selected by default. The double and bigint types are supported.

2. **You can change the data type of the input columns.**

**The input columns of GBDT binary classification only support the continuous type and are processed in the same way as the discrete type.**

### 3. Set the parameters.

Table 8-30: Parameters

Parameter	Description
Metric Type	The normalized discounted cumulative gain (NDCG) and discounted cumulative gain (DCG).
Trees	Valid values: [1,10000]. Default value: 500.
Learning Rate	Valid values: (0, 1). Default value: 0.05.
Training Sample Fraction	Valid values: (0, 1). Default value: 0.6.
Training Feature Fraction	Valid values: (0, 1). Default value: 0.6.
Maximum Leaves	The value must be an integer in the range of [2, 1000]. Default value: 32.
Testing Data Fraction	Valid values: [0, 1]. Default value: 0.0.
Maximum Tree Depth	The value must be an integer in the range of [1, 11]. Default value: 11.
Minimum Samples per Leaf Node	The value must be an integer in the range of [100, 1000]. Default value: 500.
Random Seed	The value must be an integer in the range of [0, 10]. Default value: 0.
Maximum Splits per Feature	Valid values: [1, 1000]. Default value: 500.

### 4. View the output. For more information, see the description of the [Random Forest](#) component.

**Note:**

- GBDT and GBDT\_LR have different default types of loss functions. The default loss function of GBDT is regression loss:mean squared error loss. The default loss function of GBDT\_LR is logistic regression loss. The system automatically writes the default loss function for GBDT\_LR.
- For GBDT binary classification, the label column must be of the binary type. String type data is not supported.



- When connecting the ROC curve component, set the prediction component parameters and select a base value.

PAI command (F/L setup settings are not used)

```
PAI -name GBDT_LR
-project algo_public
-DfeatureSplitValueMaxSize="500"
-DrandSeed="0"
-Dshrinkage="0.5"
-DmaxLeafCount="32"
-DlabelColName="y"
-DinputTableName="bank_data_partition"
-DminLeafSampleCount="500"
-DgroupIDColName="nr_employed"
-DsampleRatio="0.6"
-DmaxDepth="11"
-DmodelName="xlab_m_GBDT_LR_21208"
-DmetricType="2"
-DfeatureRatio="0.6"
-DinputTablePartitions="pt=20150501"
-DtestRatio="0.0"
-DfeatureColNames="age,previous,cons_conf_idx,euribor3m"
-DtreeCount="500";
```

Table 8-31: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <code>algo_public</code> . If you change the name, the system reports an error.
<b>featureSplitValueMaxSize</b>	Optional. The maximum number of splits per feature. Valid values: [1, 1000]. Default value: 500.
<b>randSeed</b>	Optional. The number of random seeds. The value must be an integer in the range of [0, 10]. Default value: 0.
<b>shrinkage</b>	Optional. The learning rate. Valid values: (0, 1). Default value : 0.05.
<b>maxLeafCount</b>	Optional. The maximum number of leaves. The value must be an integer in the range of [2, 1000]. Default value: 32.
<b>labelColName</b>	The name of the label column selected from the input table.
<b>inputTableName</b>	The name of the input table for training.
<b>minLeafSampleCount</b>	Optional. The minimum number of samples per leaf node . The value must be an integer in the range of [100, 1000]. Default value: 500.

Parameter	Description
groupIDColName	Optional. The name of the stratification column. The whole table is considered as a stratum by default.
sampleRatio	Optional. The fraction of samples collected for training. Valid values: (0, 1). Default value: 0.6.
maxDepth	Optional. The maximum depth of a tree. The value must be an integer in the range of [1, 11]. Default value: 11.
modelName	The name of the output model.
metricType	Optional. The type of a metric. Valid values: 0 and 1. 0 represents normalized discounted cumulative gain (NDCG) and 1 represents discounted cumulative gain (DCG).
featureRatio	Optional. The fraction of features collected for training. Valid values: (0, 1). Default value: 0.6.
inputTablePartitions	Optional. The partitions selected from the input prediction table. If no partitions are specified, the whole table is selected.
testRatio	Optional. The fraction of testing samples. Valid values: [0, 1]. Default value: 0.0.
featureColNames	The names of feature columns selected from the input table for training.
treeCount	Optional. The number of trees. Valid values: [1, 10000]. Default value: 500.

### 8.3.6.1.2 Linear SVM

Support-vector machines (SVMs) are developed based on the VC dimension theory and the structural risk minimization principle.

This linear SVM version is not implemented using the kernel function. For more information, see Trust Region Method for L2-SVM at <http://www.csie.ntu.edu.tw/~cjlin/papers/logistic.pdf>. This algorithm only supports binary classification.


#### Procedure

##### 1. Configure column settings.

- **Feature Columns:** You can select a feature column of the bigint or double type.
- **Label Column:** The data type of the label column can be bigint, double, or string. This component only supports binary classification.

## 2. Set parameters.

Table 8-32: Parameters

Parameter	Description
Positive Sample Label	Optional. The value of the positive sample. If this parameter is not specified, the system randomly selects a value. We recommend that you specify this parameter when the positive and negative samples are significantly different.
Positive Penalty Factor	Optional. The weight of the positive sample. Valid values: (0, $+\infty$ ). Default value: 1.0.
Negative Penalty Factor	Optional. The weight of the negative sample. Valid values: (0, $+\infty$ ). Default value: 1.0.
Convergence Coefficient	Optional. The convergence deviation. Valid values: (0, 1). Default value: 0.001.  <b>Note:</b> If no base value is specified, Positive Penalty Factor and Negative Penalty Factor must be set to the same value.

## 3. View the output. For more information, see the description of the [Random Forest](#) component.

PAI command (F/L setup settings are not used)

```
PAI -name LinearSVM
-project algo_public
-DnegativeCost="1.0"
-DmodelName="xlab_m_LinearSVM_6143"
-DpositiveCost="1.0"
-Depsilon="0.001"
-DlabelColName="y"
-DfeatureColNames="pdays,emp_var_rate,cons_conf_idx"
-DinputTableName="bank_data"
-DpositiveLabel="0";
```

Table 8-33: Parameters

Parameter	Description
name	The name of the component.
project	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is algo_public. If you change the name, the system reports an error.

Parameter	Description
<b>negativeCost</b>	<b>Optional. The weight of the negative sample. It is the penalty factor of the negative sample. Valid values: (0, <math>+\infty</math>). Default value: 1.0.</b>
<b>modelName</b>	<b>The name of the output model.</b>
<b>positiveCost</b>	<b>Optional. The weight of the positive sample. It is the penalty factor of the positive sample. Valid values: (0, <math>+\infty</math>). Default value: 1.0.</b>
<b>epsilon</b>	<b>Optional. The convergence coefficient. Valid values: (0, 1). Default value: 0.001.</b>
<b>labelColName</b>	<b>The name of the label column.</b>
<b>featureColNames</b>	<b>The names of feature columns selected from the input table for training.</b>
<b>inputTableName</b>	<b>The name of the input table for training.</b>
<b>positiveLabel</b>	<b>Optional. The value of the positive sample. If this parameter is not specified, the system randomly selects a value.</b>

### 8.3.6.1.3 Logistic regression for binary classification

Binary classification is a classic logistic regression method. Logistic regression on the algorithm platform supports multiclass classification. The logistic regression component supports two data types: sparse and dense.

Parameter settings

Table 8-34: Parameters

Parameter	Description
<b>Regularization Type</b>	<b>Optional. The type of regularization. Valid values: L1, L2, and None. Default value: L1.</b>
<b>Maximum Iterations</b>	<b>Optional. The maximum number of L-BFGS iterations. Default value: 100.</b>
<b>Regularization Coefficient</b>	<b>Optional. The regularization coefficient. Default value: 1.0. If regularizedType is set to None, this parameter is ignored.</b>
<b>Minimum Convergence Deviance</b>	<b>Optional. The condition to terminate L-BFGS. This is the log-likelihood deviation between two iterations. Default value: 1.0e-06.</b>

The logistic regression component outputs a model, which is available in the model list.

**Model name format:** Experiment Name + "-" + Component Name + "model".

PAI command (F/L setup settings are not used)

```
PAI -name LogisticRegression
-project algo_public
-DmodelName="xlab_m_logistic_regression_6096"
-DregularizedLevel="1"
-DmaxIter="100"
-DregularizedType="l1"
-Depsilon="0.000001"
-DlabelColName="y"
-DfeatureColNames="pdays,emp_var_rate"
-DgoodValue="1"
-DinputTableName="bank_data";
```

Table 8-35: Parameters

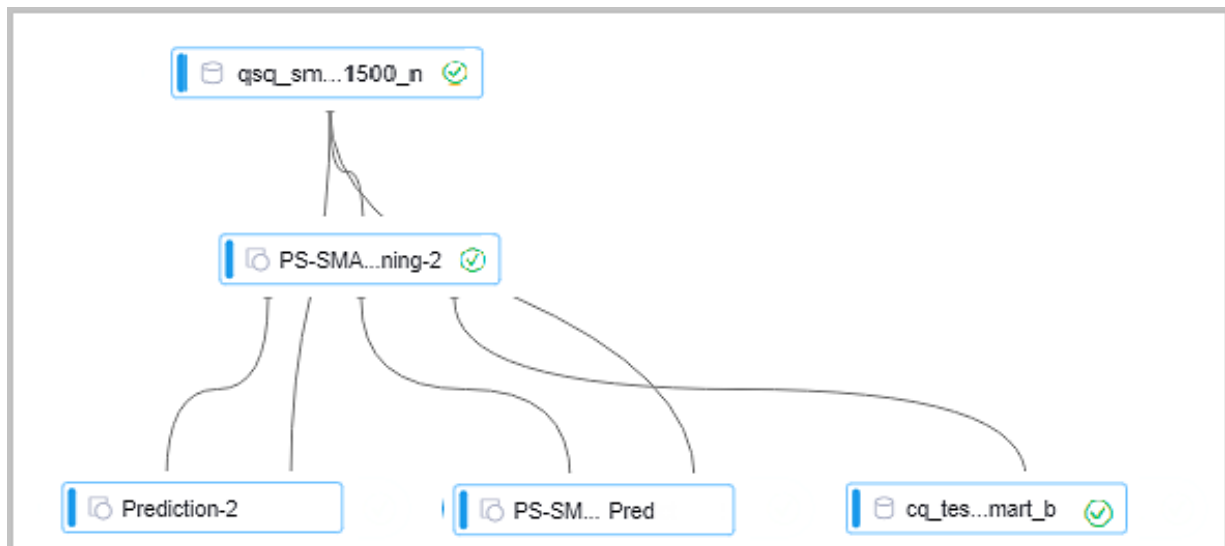
Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <code>algo_public</code> . If you change the name, the system reports an error.
<b>modelName</b>	The name of the output model.
<b>regularizedLevel</b>	Optional. The regularization coefficient. Default value: 1.0. If <code>regularizedType</code> is set to <code>None</code> , this parameter is ignored.
<b>maxIter</b>	Optional. The maximum number of L-BFGS iterations. Default value: 100.
<b>regularizedType</b>	Optional. The type of regularization. Valid values: <code>L1</code> , <code>L2</code> , and <code>None</code> . Default value: <code>L1</code> .
<b>epsilon</b>	Optional. The convergence deviation. It is the condition to terminate L-BFGS. This is the log-likelihood deviation between two iterations. Default value: <code>1.0e-06</code> .
<b>labelColName</b>	The name of the label column selected from the input table.
<b>featureColNames</b>	The names of feature columns selected from the input table for training.
<b>goodValue</b>	Optional. The base value. For binary classification, specify the label value of the training coefficient. If this parameter is not specified, the system randomly selects a value.

Parameter	Description
inputTableName	The name of the input table for training.

#### 8.3.6.1.4 PS-SMART binary classification

A [parameter server](#) (PS) is used to train a large number of models online and offline. Scalable Multiple Additive Regression Tree (SMART) is an implementation of Gradient Boosting Decision Tree (GBDT) on PS. PS-SMART can run training tasks containing up to tens of billions of samples and hundreds of thousands of features on thousands of nodes. It also supports failover for high stability. PS-SMART supports various data formats, training targets, evaluation targets, output feature importance, and histogram approximation for training acceleration.

Quick start



As shown in the figure, a PS-SMART binary classification model is learned based on training data. The model has three output ports:

- **Output model:** offline model, which is connected to the unified prediction component. This model does not support the output of leaf node numbers.
- **Output model table:** a binary table that is not readable and is used to ensure compatibility with the PS-SMART prediction component. The table supports the output of leaf node numbers, which ensures higher efficiency, less resource consumption, and higher stability.
- **Output feature importance table:** lists the importance of each feature. Three importance types are supported. For more information, see [Parameters](#).

## PAI command

### • Training

```
PAI -name ps_smart
 -project algo_public
 -DinputTableName="smart_binary_input"
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545859_2"
 -DoutputImportanceTableName="pai_temp_24515_545859_3"
 -DlabelColName="label"
 -DfeatureColNames="f0,f1,f2,f3,f4,f5"
 -DenableSparse="false"
 -Dobjective="binary:logistic"
 -Dmetric="error"
 -DfeatureImportanceType="gain"
 -DtreeCount="5";
 -DmaxDepth="5"
 -Dshrinkage="0.3"
 -Dl2="1.0"
 -Dl1="0"
 -Dlifecycle="3"
 -DsketchEps="0.03"
 -DsampleRatio="1.0"
 -DfeatureRatio="1.0"
 -DbaseScore="0.5"
 -DminSplitLoss="0"
```

### • Prediction

```
PAI -name prediction
 -project algo_public
 -DinputTableName="smart_binary_input";
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545860_1"
 -DfeatureColNames="f0,f1,f2,f3,f4,f5"
 -DappendColNames="label,qid,f0,f1,f2,f3,f4,f5"
 -DenableSparse="false"
```

-Dlifecycle="28"

## Parameters

### • Data parameters

Command option	Parameter	Description	Valid values	Remarks
<b>featureColNames</b>	<b>Feature Columns</b>	The names of feature columns selected from the input table for training.	If the column name is in dense format , it must be of the bigint or double type. If the column name is in sparse KV format, it must be a string, and its keys and values must be numeric.	Required
<b>labelColName</b>	<b>Label Column</b>	The name of the label column selected from the input table .	The column name can be of either string or numeric type , but only numeric data can be stored in the columns . For example , in binary classification , the column value can be 0 or 1.	Required
<b>weightCol</b>	<b>Weight Column</b>	This column specifies the weight of each sample.	The column name can be of the numeric type.	Optional. Default value: null.



Command option	Parameter	Description	Valid values	Remarks
<b>enableSparse</b>	<b>Use Sparse Format</b>	This parameter specifies whether the data in the input table is in sparse format, in which key-value pairs are separated by spaces whereas keys and values are separated by colons (:), for example, 1:0.3 3:0.9.	[true, false]	Optional. Default value: false.
<b>inputTableName</b>	<b>Input Table Name</b>	N/A	N/A	Required
<b>modelName</b>	<b>Output Model Name</b>	N/A	N/A	Required
<b>outputImportanceTableName</b>	<b>Output Feature Importance Table Name</b>	N/A	N/A	Optional. Default value: null.
<b>inputTablePartitions</b>	<b>Input Table Partitions</b>	N/A	N/A	Optional. The parameter value must be in ds=1/pt=1 format.

Command option	Parameter	Description	Valid values	Remarks
<b>outputTable</b>	<b>Output Model Table Name</b>	The output table is a MaxCompute table that uses the binary format and is not readable. The prediction component that comes with SMART can be used to generate leaf node numbers .	String	Optional
<b>lifecycle</b>	<b>Output Table Lifecycle</b>	N/A	Positive integer	Optional. Default value: 3.

• Algorithm parameters

Command option	Parameter	Description	Valid values	Remarks
<b>objective</b>	<b>Objective Function Type</b>	The objective function type affects learning and must be selected properly. Select binary:logistic for binary classification.	N/A	Required

Command option	Parameter	Description	Valid values	Remarks
<b>metric</b>	<b>Evaluation Indicator Type</b>	Evaluation indicators in the training set, which are exported to stdout of the coordinator in a logview.	logloss, error and auc	Optional. Default value: null.
<b>treeCount</b>	<b>Trees</b>	The number of trees. The training time is proportional to this number.	Positive integer	Optional. Default value: 1.
<b>maxDepth</b>	<b>Maximum Tree Depth</b>	The maximum depth of a tree. We recommend that you set this value to 5, which means the tree can contain up to 32 leaf nodes.	A positive integer in the range of [1, 20]	Optional. Default value: 5.
<b>sampleRatio</b>	<b>Data Sampling Fraction</b>	The data sampling rate when trees are built. The sample data is used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means data sampling is disabled.

Command option	Parameter	Description	Valid values	Remarks
featureRatio	Feature Sampling Fraction	The feature sampling rate when trees are built. The sample features are used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means feature sampling is disabled.
l1	L1 Penalty Coefficient	This parameter determines the number of leaf nodes. The greater the value, the less the leaf nodes. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 0.
l2	L2 Penalty Coefficient	This parameter determines the size of a leaf node. The greater the value, the more evenly the leaf nodes are distributed. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 1.0.

Command option	Parameter	Description	Valid values	Remarks
<b>shrinkage</b>	<b>Learning Rate</b>	N/A	(0, 1]	Optional. Default value: 0.3.
<b>sketchEps</b>	<b>Sketch-based Approximate Precision</b>	The threshold for selecting quantiles when you build a sketch . The number of buckets is $O(1.0/sketchEps)$ . The smaller the parameter value, the more buckets are generated. Typically, you do not need to modify this value.	(0, 1)	Optional. Default value: 0.03.
<b>minSplitLoss</b>	<b>Minimum Split Loss Change</b>	The minimum split loss changes required for splitting a node. The greater the value, the more conservatively the node splits.	Non-negative real number	Optional. Default value: 0.

Command option	Parameter	Description	Valid values	Remarks
<b>featureNum</b>	<b>Features</b>	The number of features or the maximum feature ID. Specify this parameter for resource usage estimation.	Positive integer	Optional
<b>baseScore</b>	<b>Global Offset</b>	Original predicted values of all samples.	Real number	Optional. Default value: 0.5.
<b>featureImportanceType</b>	<b>Feature Importance Type</b>	The type of feature importance. weight indicates the number of times that a feature splits. gain indicates information gain brought by the feature. cover indicates the number of samples that the feature covers on the splitting nodes.	weight, gain, and cover	Optional. Default value: gain.

• **Note**

- Specify different values for the objective parameter in different learning models. On the binary classification Web GUI, the objective function is

automatically specified and invisible to users. On the command line, set the objective parameter to `binary:logistic`.

- Mappings between metrics and objective functions are: logloss for negative loglikelihood for logistic regression, error for binary classification error, and auc for Area under curve for classification.

#### Execution optimization

Command option	Parameter	Description	Valid values	Remarks
<b>coreNum</b>	<b>Cores</b>	The number of cores. The greater the value, the faster the computing algorithm runs.	Positive integer	Optional. Automatically calculated.
<b>memSizePer Core</b>	<b>Memory Size per Core (MB)</b>	The memory size of each core, where 1024 represents 1 GB of memory.	Positive integer	Optional. Automatically calculated.

#### Example

- **Data generation**

The following example uses data in dense format.

```
drop table if exists lm_test_input;
create table smart_binary_input lifecycle 3 as
select
*
from
(
select 0.72 as f0, 0.42 as f1, 0.55 as f2, -0.09 as f3, 1.79 as f4,
-1.2 as f5, 0 as label from dual
union all
select 1.23 as f0, -0.33 as f1, -1.55 as f2, 0.92 as f3, -0.04 as f4,
-0.1 as f5, 1 as label from dual
union all
select -0.2 as f0, -0.55 as f1, -1.28 as f2, 0.48 as f3, -1.7 as f4,
1.13 as f5, 1 as label from dual
union all
select 1.24 as f0, -0.68 as f1, 1.82 as f2, 1.57 as f3, 1.18 as f4,
0.2 as f5, 0 as label from dual
union all
```

```
select -0.85 as f0, 0.19 as f1, -0.06 as f2, -0.55 as f3, 0.31 as f4
, 0.08 as f5, 1 as label from dual
union all
select 0.58 as f0, -1.39 as f1, 0.05 as f2, 2.18 as f3, -0.02 as f4
, 1.71 as f5, 0 as label from dual
union all
select -0.48 as f0, 0.79 as f1, 2.52 as f2, -1.19 as f3, 0.9 as f4,
-1.04 as f5, 1 as label from dual
union all
select 1.02 as f0, -0.88 as f1, 0.82 as f2, 1.82 as f3, 1.55 as f4,
0.53 as f5, 0 as label from dual
union all
select 1.19 as f0, -1.18 as f1, -1.1 as f2, 2.26 as f3, 1.22 as f4,
0.92 as f5, 0 as label from dual
union all
select -2.78 as f0, 2.33 as f1, 1.18 as f2, -4.5 as f3, -1.31 as f4,
-1.8 as f5, 1 as label from dual
) tmp;
```

- **Training**

Configure the training data and training components, as shown in [Quick start](#).

Select the label column as the target column and columns f0, f1, f2, f3, f4, f5 as feature columns.

- You do not need to set the number of features because this number is calculated automatically by the algorithm. If you have a large number of features and want the algorithm to accurately estimate the amount of required resources, specify the actual number of features.
- To accelerate the training, set the number of cores on the execution optimization page. The greater the number, the faster the algorithm runs. Typically, you do not need to enter the memory size per core because the algorithm can accurately calculate the memory size. The PS algorithm starts to run only when all hosts have obtained the required resources. Therefore, you may need to wait for a longer period of time when the cluster is busy and resources are requested in large volumes.
- You can view the output values of the metrics in the stdout of the coordinator in a logview (HTTP link starting with <http://logview.odps.aliyun-inc.com:8080/logview>). A single PS-SMART training job can contain multiple tasks, and therefore multiple logviews are created. Select the logview whose name starts with PS to view the output of the PS job.



- Prediction

- Use the unified prediction component

The model generated after training is saved in binary format and can be used for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#).

If the dense format is used, you only need to select feature columns. (All columns are selected by default, and extra columns do not affect the prediction.) If the KV format is used, set the data format to sparse format and select the correct delimiter. In the SMART model, key-value pairs are separated by space characters. Therefore, the delimiter must be set to space or \u0020 (escape expression of spaces).

In the "prediction\_detail" column, value 1 indicates a positive sample, and value 0 indicates a negative sample. The values following 0 and 1 indicate the probabilities of the corresponding classes.

- Use the PS-SMART prediction component

The output model table obtained after training is saved in binary format and can be used by the PS-SMART prediction component for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#). Set the required parameters, including the data format, feature columns, target column, and number of classes. The ID column can only be a string type column other than a feature column or a target column. The loss function must be set to binary:logistic.

The prediction\_score column lists probabilities of predicted positive samples. A sample is predicted as a positive sample if its score is greater than 0.5. Otherwise, it is predicted as a negative sample. The leaf\_index column lists the predicted leaf node numbers. Each sample has N numbers, where N is the number of decision trees. Each tree is mapped to a number, which indicates the leaf node number of the sample on this tree.



Note:

- The output model table is a binary table that is not readable and is used to support the PS-SMART prediction component. The table provides outputs such as leaf node numbers and evaluation indicators. However, the output

table has strict requirements on data formats, which negatively affects user experience. This component is being continually improved, and may be replaced by another component in the future.

- A string type column must be selected as the label column. You can enter strings in the column but cannot be blank or NULL. A feature column can be converted to the string type by using the data type conversion component.
- The loss function must be explicitly set to binary:logistic. By default, the function does not work.

- View feature importance

To view feature importance, you can export the third output port to an output table, or right-click PS-SMART training component and choose View Data > Output Feature Importance Table from the shortcut menu.

order ▲	id ▲	value ▲
1	0	0.5690338015556335
2	1	0.21714292466640472
3	4	0.21382322907447815

In the table, the ID column lists the numbers of input features. In this example, the data is in dense format. The input features are f0,f1,f2,f3,f4,f5. Therefore, ID 0 represents f0 and ID 4 represents f4. If the KV format is used, the IDs represent keys in key-value pairs. Each value indicates a feature importance type. The default value is gain, indicating the sum of information gains brought by a feature in the model. The preceding figure shows only three features because only these three features are used during the tree split process. In this case, the importance of unused features is 0.

## FAQ

- Q: Does PS\_SMART support non-numerical features and tags?
- A: No.
- Q: What is the scale of features supported by PS-SMART? Can we use large-scale 0-1 features?
- A: Although PS-SMART supports tasks that contain hundreds of thousands of features, such tasks consume large amounts of resources and run slowly. Therefore, we recommend that you do not use such a large number of features

- The GBDT algorithm is suitable for training with continuous features. The categorical features require one-hot coding to filter out infrequent features before they can be used for training. The continuous numerical features can be used for training with the GBDT algorithm directly. Discretization is not recommended for numerical features.
- Q: Why is the result different every time although the SMART algorithm has the same data and the same parameter settings?
- A: The PS-SMART algorithm applies randomness in many scenarios. For example, the `data_sample_ratio` and `fea_sample_ratio` items introduce data and feature sampling respectively. In addition, the PS-SMART algorithm uses histograms to show similarity. When multiple workers run in a cluster in distributed mode, local sketches are merged to global sketches in a random order. Although different merging orders result in different tree structures, this does not introduce too much variation to the output model. Therefore, it is normal situation to obtain different results after the algorithm runs multiple times with the same data and same parameter settings.



**Note:**

- The target column in a PS-SMART binary classification model supports only numerical values (0 for negative samples and 1 for positive samples). Even if values in the MaxCompute table are strings, they are saved as numerical values. If the classification target is a type string similar to Good or Bad, convert it to 1 or 0.
- In the key-value format, feature IDs must be positive integers, and feature values must be real numbers. If feature IDs are strings, use the serialization component to serialize them. If the feature values are classification type strings, perform feature engineering, such as discretization.

## 8.3.6.2 Multiclass classification

### 8.3.6.2.1 KNN

The KNN algorithm is used to resolve classification issues. For a row in the prediction table, this component selects K entries nearest to the row from the

**training table. It then assigns the row to the class that is most common among the K entries.**

PAI command

```
PAI -name knn
 -DtrainTableName=pai_knn_test_input
 -DtrainFeatureColNames=f0,f1
 -DtrainLabelColName=class
 -DpredictTableName=pai_knn_test_input
 -DpredictFeatureColNames=f0,f1
 -DoutputTableName=pai_knn_test_output
 -Dk=2;
```

Parameters

Parameter	Description	Valid value	Default value
<b>trainTableName</b>	<b>Required. The name of the training table.</b>	<b>Table name</b>	<b>N/A</b>
<b>trainFeatureColNames</b>	<b>Required. The names of feature columns selected from the training table.</b>	<b>Column name</b>	<b>N/A</b>
<b>trainLabelColName</b>	<b>Required. The name of the label column selected from the training table.</b>	<b>Column name</b>	<b>N/A</b>
<b>trainTablePartitions</b>	<b>Optional. The partitions selected from the training table.</b>	<b>Partition name</b>	<b>All partitions are selected by default.</b>
<b>predictTableName</b>	<b>Required. The name of the prediction table.</b>	<b>Table name</b>	<b>N/A</b>
<b>outputTableName</b>	<b>Required. The name of the output table.</b>	<b>Table name</b>	<b>N/A</b>

Parameter	Description	Valid value	Default value
<b>predictFeatureColNames</b>	Optional. The names of feature columns selected from the prediction table.	Column name	N/A
<b>predictTablePartition</b>	Optional. The partitions selected from the prediction table.	Partition name	All partitions are selected by default.
<b>appendColNames</b>	Optional. The names of columns appended to the output table from the prediction table.	Column name	N/A
<b>outputTablePartition</b>	Optional. The partitions in the output table.	Partition name	The output table is non-partitioned by default.
<b>k</b>	Optional. The number of the nearest neighbors.	A positive integer in the range of [1, 1000]	100
<b>enableSparse</b>	Optional. This parameter specifies whether the data in the input table is in sparse format.	true and false	false
<b>itemDelimiter</b>	Optional. The delimiter used to separate key-value pairs when the data in the input table is in sparse format.	Symbol	The default delimiter is a space .

Parameter	Description	Valid value	Default value
<b>kvDelimiter</b>	<b>Optional. The delimiter used to separate keys and values when the data in the input table is in sparse format.</b>	<b>Symbol</b>	<b>The default delimiter is a colon (:).</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>This parameter is used with memSizePerCore. The value must be a positive integer in the range of [1, 20000].</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB .</b>	<b>A positive integer in the range of [ 1024, 65536]</b>	<b>Automatically calculated.</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>

## Examples

### • Test data

```
create table pai_knn_test_input as
select * from
(
 select 1 as f0,2 as f1, 'good' as class from dual
 union all
 select 1 as f0,3 as f1, 'good' as class from dual
 union all
 select 1 as f0,4 as f1, 'bad' as class from dual
 union all
 select 0 as f0,3 as f1, 'good' as class from dual
 union all
 select 0 as f0,4 as f1, 'bad' as class from dual
)tmp;
```

### • PAI command

```
PAI -name knn
 -DtrainTableName=pai_knn_test_input
 -DtrainFeatureColNames=f0,f1
 -DtrainLabelColName=class
 -DpredictTableName=pai_knn_test_input
 -DpredictFeatureColNames=f0,f1
```

```
-DoutputTableName=pai_knn_test_output
-Dk=2;
```

- **Output description**

f0	f1	prediction_result	prediction_score	prediction_detail
1	4	bad	1.0	{"bad": 1}
0	4	bad	1.0	{"bad": 1}
0	3	bad	0.5	{"bad": 0.5, "good": 0.5}
1	3	good	1.0	{"good": 1}
1	2	good	1.0	{"good": 1}

- **f0 and f1:** the appended columns in the output table.
- **prediction\_result:** the classification result.
- **prediction\_score:** the probabilities for the classification result.
- **prediction\_detail:** the latest K conclusions and their probabilities.

### 8.3.6.2.2 Logistic regression for multiclass classification

Logistic regression of Apsara Stack Machine Learning Platform for AI supports multiclass classification. The logistic regression component supports two data formats: sparse and dense.

Parameter settings

Table 8-36: Parameters

Parameter	Description
<b>Regularization Type</b>	<b>Optional.</b> The type of regularization. Valid values: L1, L2, and None. <b>Default value:</b> L1.
<b>Max Iterations</b>	<b>Optional.</b> The maximum number of L-BFGS iterations. <b>Default value:</b> 100.
<b>Regularization Coefficient</b>	<b>Optional.</b> The regularization coefficient. <b>Default value:</b> 1.0. If regularizedType is set to None, this parameter is ignored.
<b>Minimum Convergence Deviance</b>	<b>Optional.</b> The condition to terminate L-BFGS. This is the log-likelihood deviance between two iterations. <b>Default value:</b> 1.0e-06.

The logistic regression component outputs a model, which is available in the model list.

**Model naming format:** Experiment Name + "-" + Component Name + "model".

PAI command (F/L setup settings are not used)

```
PAI -name LogisticRegression
-project algo_public
-DmodelName="xlab_m_logistic_regression_6096"
-DregularizedLevel="1"
-DmaxIter="100"
-DregularizedType="l1"
-Depsilon="0.000001"
-DlabelColName="y"
-DfeatureColNames="pdays,emp_var_rate"
-DgoodValue="1"
-DinputTableName="bank_data";
```

Table 8-37: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <code>algo_public</code> . If you change the name, the system reports an error.
<b>modelName</b>	The name of the output model.
<b>regularizedLevel</b>	Optional. The regularization coefficient. Default value: 1.0. If <code>regularizedType</code> is set to <code>None</code> , this parameter is ignored.
<b>maxIter</b>	Optional. The maximum number of L-BFGS iterations. Default value: 100.
<b>regularizedType</b>	Optional. The type of regularization. Valid values: L1, L2, and None. Default value: L1.
<b>epsilon</b>	Optional. The convergence deviation. It is the condition to terminate L-BFGS. This is the loglikelihood deviation between two iterations. Default value: 1.0e-06.
<b>labelColName</b>	The name of the label column selected from the input table.
<b>featureColNames</b>	The names of feature columns selected from the input table for training.
<b>goodValue</b>	Optional. The base value. For multiclass classification, specify the label value of the training coefficient. If this parameter is not specified, the system randomly selects a value.
<b>inputTableName</b>	The name of the input table for training.



### 8.3.6.2.3 Random forest

A random forest is a classifier that contains multiple decision trees. Its output class is decided by the mode of individual tree output classes.

#### Procedure

1. Drag and drop the Random Forest component onto the canvas and select columns.

Table 8-38: Parameters

Parameter	Description
Feature Columns	Optional. All columns except the label and weight columns are selected by default.
Excluded Columns	Optional. This parameter is used to exclude specified columns from training. This parameter is mutually exclusive with <code>featureColNames</code> .
Columns Forced to Convert	Optional. The default feature parsing rules are as follows: <ul style="list-style-type: none"><li>• Parse columns of string, boolean, and datetime types to discrete columns.</li><li>• Parse columns of double and bigint types to contiguous columns.</li><li>• Set the <code>forceCategorical</code> parameter to parse bigint type columns to categorical columns.</li></ul>
Weight Columns	Optional. You can select all columns except the input and label columns. The double and bigint types are supported.
Label Column	You can select all columns except the input column. The bigint, double, and string types are supported.

2. Set the parameters of the Random Forest component.

Table 8-39: Parameters

Parameter	Description
Trees	The number of trees in the forest. Valid values: (0, 1000).

Parameter	Description
Single-tree Algorithm Type	<p>Optional. The algorithm type of each tree in the forest. Valid values: id3, c4.5, and cart. If the forest has <math>n</math> trees and the condition is <code>algorithmTypes = a,b</code>, then <code>[0,a)</code> indicates id3, <code>[a,b)</code> indicates cart, and <code>[b,n)</code> indicates c4.5.</p> <p>If this parameter is set to <code>[2, 4]</code> for a forest with five trees, <code>[0, 1)</code> indicates the ID3 algorithm, <code>[2, 3)</code> indicates the CART algorithm, and 4 indicates the C4.5 algorithm. If the value is None, the algorithms are evenly allocated across the forest.</p>
Random Features per Tree	The number of features selected randomly. Valid values: 1 to N. N indicates the number of features.
Minimum Samples per Leaf Node	Optional. The minimum number of samples per leaf node. The value must be a positive integer no less than 2.
Minimum Fraction of Samples on Leaf Node to Samples on Parent Node	The minimum fraction of samples on a leaf node to samples on a parent node. A value of -1 indicates that no limit is set. Default value: -1. Valid values: <code>[0, 1]</code> .
Maximum Tree Depth	The maximum depth of a tree. -1 indicates a completely grown tree. Valid values: <code>[1, ∞)</code> .
Random Samples Input per Tree	The number of random samples input per tree. Valid values: <code>(1000, 1000000]</code> .



**Note:**

- With improvement of the bagging method, the Random Forest component builds a forest without correlated trees in the big cube. Random forests are similar to the boosting method in many aspects, particularly their training processes.
- For the growth of a single tree, this method provides the id3, cart, and c4.5 options. The `treeNum` parameter is used to specify the number of trees in the forest, in the range of `[1, 1000]`. The structure of a single tree can be controlled based on the edited template. You can use other parameters to specify the minimum number of samples per leaf node, the minimum

fraction of samples on a leaf node to samples on a parent node, and the maximum depth of a tree.

- Each row in the weight column corresponds to a sample and indicates the proportion of this sample in training. If the age column is selected as the weight column, the sample in the row with a higher weight value in the age column has a higher proportion during the training.
- The "input table is empty!" error may occur in the following situations: The sampling fraction is too small, which means that the value of maxRecordSize is too small, or the input table is empty.

PAI command (F/L setup settings are not used)

```
PAI -name RandomForest
-project algo_public
-DmodelName="xlab_m_random_forests_6036"
-DrandomColNum="1.0"
-DlabelColName="campaign"
-DmaxTreeDeep="10"
-DmaxRecordSize="100000"
-DfeatureColNames="age,pdays,previous,emp_var_rate,cons_price_idx,
cons_conf_idx,euribor3m,nr_employed"
-DisFeatureContinuous="1,1,1,1,1,1,1,1"
-DminNumPer="-1"
-DminNumObj="2"
-DinputTableName="bank_data"
-DweightColName="y"
-DtreeNum="10";
```

Table 8-40: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <b>algo_public</b> . If you change the name, the system reports an error.
<b>modelName</b>	The name of the output model.
<b>randomColNum</b>	Optional. The random attribute type. This parameter specifies the number of features randomly selected each time a single tree is generated. -1 indicates $\log_2 N$ . Valid values: 1 to N. N indicates the number of features.
<b>labelColName</b>	The name of the label column selected from the input table.
<b>maxTreeDeep</b>	Optional. The maximum depth of a tree. -1 indicates a completely grown tree. Valid values: [1, $\infty$ ].

Parameter	Description
<b>maxRecordSize</b>	<b>Optional. The maximum number of samples per tree. Valid values: (1000, 1000000). -1 indicates 100000.</b>
<b>featureColNames</b>	<b>The names of feature columns selected from the input table for training.</b>
<b>isFeatureContinuous</b>	<b>Specifies whether the feature for subsequent columns is continuous or discrete. 1 indicates that the feature column data is continuous, while 0 indicates that the feature column data is discrete. 1, 0, 0 indicates that values are continuous in the first feature column and discrete in the second and third feature columns. The number of values corresponds to the feature length.</b>
<b>minNumPer</b>	<b>Optional. The minimum fraction of samples on a leaf node to samples on a parent node. A value of -1 indicates that no limit is set. Valid values: [0.0, 1.0].</b>
<b>minNumObj</b>	<b>Optional. The minimum number of samples per leaf node.</b>
<b>inputTableName</b>	<b>The name of the input table for training.</b>
<b>weightColName</b>	<b>Optional. The name of the weight column selected from the input table. If there is no weight column, set this parameter to None. If there is any weight column, the value of weightColName is greater than 0.</b>
<b>treeNum</b>	<b>The number of trees. Valid values: (0, 1000).</b>

#### 8.3.6.2.4 Naive Bayes

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions between the features. A probabilistic model that can more accurately describe this potential is called an independent feature model.

Component description

**Component parameter settings (For more information, see the description of the Random Forest component.)**

PAI command

```
PAI -name NaiveBayes
 -project algo_public
 -DmodelName="xlab_m_NaiveBayes_23772"
 -DinputTablePartitions="pt=20150501"
 -DlabelColName="poutcome"
 -DfeatureColNames="age,previous,cons_conf_idx,euribor3m"
```

```
-DisFeatureContinuous="1,1,1,1"
-DinputTableName="bank_data_partition";
```

Table 8-41: Parameters

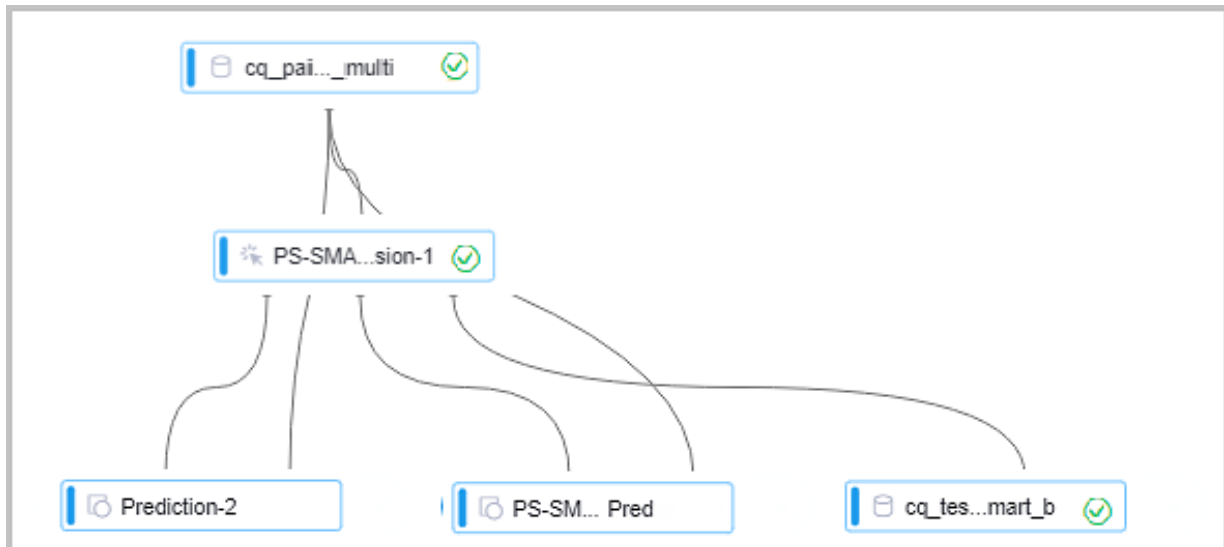
Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <code>algo_public</code> . If you change the name, the system reports an error.
<b>modelName</b>	The name of the model generated by training.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input prediction table. If no partitions are specified, the entire table is selected.
<b>labelColName</b>	The name of the label column selected from the input table.
<b>featureColNames</b>	The names of feature columns selected from the input table for training.
<b>isFeatureContinuous</b>	Specifies whether the feature for subsequent columns is continuous or discrete. 1 indicates that the feature column data is continuous, while 0 indicates that the feature column data is discrete. 1,0,0 indicates that values are continuous in the first feature column and discrete in the second and third feature columns. The number of values corresponds to the feature length.
<b>inputTableName</b>	The name of the input table.

#### 8.3.6.2.5 PS-SMART multiclass classification

A *parameter server* (PS) is used to train a large number of models online and offline.

Scalable Multiple Additive Regression Tree (SMART) is an implementation of Gradient Boosting Decision Tree (GBDT) on PS. PS-SMART can run training tasks containing up to tens of billions of samples and hundreds of thousands of features on thousands of nodes. It also supports failover for high stability. PS-SMART supports various data formats, training targets, evaluation targets, output feature importance, and histogram approximation for training acceleration.

## Quick start



As shown in the figure, a PS-SMART multiclass classification model is learned based on training data. The model has three output ports:

- **Output model: offline model**, which is connected to the unified prediction component. This model does not support the output of leaf node numbers.
- **Output model table: a binary table** that is not readable and is used to support the PS-SMART prediction component. The table provides outputs such as leaf node numbers and assessment metrics. However, the output table has strict requirements on data formats, which negatively affects user experience. This component is being continually improved, and may be replaced by another component in the future.
- **Output feature importance table: lists the importance of each feature.** Three importance types are supported. For more information, see [Parameters](#).

## PAI command

## • Training

```
PAI -name ps_smart
 -project algo_public
 -DinputTableName="smart_multiclass_input"
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545859_2"
 -DoutputImportanceTableName="pai_temp_24515_545859_3"
 -DlabelColName="label"
 -DfeatureColNames="features"
 -DenableSparse="true"
 -Dobjective="multi:softprob"
 -Dmetric="mlogloss"
 -DfeatureImportanceType="gain"
 -DtreeCount="5";
 -DmaxDepth="5"
```

```
-Dshrinkage="0.3"
-Dl2="1.0"
-Dl1="0"
-Dlifecycle="3"
-DsketchEps="0.03"
-DsampleRatio="1.0"
-DfeatureRatio="1.0"
-DbaseScore="0.5"
-DminSplitLoss="0"
```

#### • Prediction

```
PAI -name prediction
 -project algo_public
 -DinputTableName="smart_multiclass_input";
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545860_1"
 -DfeatureColNames="features"
 -DappendColNames="label, features"
 -DenableSparse="true"
 -DkvDelimiter=":"
 -Dlifecycle="28"
```

#### Parameters

#### • Data parameters

Command option	Parameter	Description	Valid values	Remarks
featureCol Names	Feature Column	The names of feature columns selected from the input table for training.	If the column name is in dense format, it must be of the bigint or double type. If the column name is in sparse KV format, it must be a string, and its keys and values must be numeric.	Required

Command option	Parameter	Description	Valid values	Remarks
labelColName	Label Column	The name of the label column selected from the input table .	The column name can be of either string or numeric type , but only numeric data can be stored in the columns. For multiclass classification, column values can be 0, 1 , 2, ..., n-1, where n is the number of classes.	Required
weightCol	Weight Column	This column specifies the weight of each sample.	The column name can be of the numeric type.	Optional. Default value: null.
enableSparse	Use Sparse Format	This parameter specifies whether the data in the input table is in sparse format, in which key-value pairs are separated by spaces whereas keys and values are separated by colons (:), for example, 1:0.3 3:0.9.	true, false	Optional. Default value: false.



Command option	Parameter	Description	Valid values	Remarks
<b>inputTableName</b>	<b>Input Table Name</b>	N/A	N/A	<b>Required</b>
<b>modelName</b>	<b>Output Model Name</b>	N/A	N/A	<b>Required</b>
<b>outputImportanceTableName</b>	<b>Output Feature Importance Table Name</b>	N/A	N/A	<b>Optional.</b> <b>Default value:</b> <b>null.</b>
<b>inputTablePartitions</b>	<b>Input Table Partitions</b>	N/A	N/A	<b>Optional.</b> The parameter value must be in ds=1/pt=1 format.
<b>outputTableName</b>	<b>Output Model Table</b>	The output table is a MaxCompute table that uses the binary format and is not readable. The prediction component that comes with SMART can be used to generate leaf node numbers .	String	<b>Optional</b>
<b>lifecycle</b>	<b>Output Table Lifecycle</b>	N/A	<b>Positive integer</b>	<b>Optional.</b> <b>Default value:</b> <b>3.</b>

• Algorithm parameters

Command option	Parameter	Description	Valid values	Remarks
<b>classNum</b>	<b>Classes</b>	The number of classes in multiclass classification. If the number of classes is <b>n</b> , the label column name can be 0, 1, 2, ..., or <b>n-1</b> .	A non-negative integer, greater than or equal to 3.	Required
<b>objective</b>	<b>Objective Function Type</b>	The objective function type affects learning and must be selected properly. Set it to <b>multi:softprob</b> for multiclass classification.	N/A	Required
<b>metric</b>	<b>Evaluation Indicator Type</b>	Evaluation indicators in the training set, which are exported to stdout of the coordinator in a logview.	<b>mloglossmerror</b>	Optional. Default value: null.
<b>treeCount</b>	<b>Trees</b>	The number of trees. The training time is proportional to this number.	Positive integer	Optional. Default value: 1.

Command option	Parameter	Description	Valid values	Remarks
<b>maxDepth</b>	<b>Maximum Decision Tree Depth</b>	The maximum depth of a tree. We recommend that you set this value to 5 , which means the tree can contain up to 32 leaf nodes.	A positive integer in the range of [1, 20 ].	Optional. Default value: 5.
<b>sampleRatio</b>	<b>Data Sampling Fraction</b>	The data sampling rate when trees are built. The sample data is used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means data sampling is disabled.
<b>featureRatio</b>	<b>Feature Sampling Fraction</b>	The feature sampling rate when trees are built. The sample features are used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means feature sampling is disabled.

Command option	Parameter	Description	Valid values	Remarks
l1	L1 Penalty Coefficient	This parameter determines the number of leaf nodes . The greater the value, the fewer the leaf nodes. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 0.
l2	L2 Penalty Coefficient	This parameter determines the size of a leaf node. The greater the value, the more evenly the leaf nodes are distributed. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 1.0.
shrinkage	Learning Rate	N/A	(0, 1]	Optional. Default value: 0.3.

Command option	Parameter	Description	Valid values	Remarks
<b>sketchEps</b>	<b>Sketch-based Approximate Precision</b>	The threshold for selecting quantiles when you build a sketch . The number of buckets is $O(1.0/\text{sketchEps})$ . The smaller the parameter value, the more buckets are generated. Typically, you do not need to modify this value.	(0, 1)	Optional. Default value: 0.03.
<b>minSplitLoss</b>	<b>Minimum Split Loss</b>	The minimum split loss changes required for splitting a node. The greater the value, the more conservatively the node splits.	Non-negative real number	Optional. Default value: 0.
<b>featureNum</b>	<b>Features</b>	The number of features or the maximum feature ID. Specify this parameter for resource usage estimation.	Positive integer	Optional

Command option	Parameter	Description	Valid values	Remarks
<b>baseScore</b>	<b>Global Offset</b>	Original predicted values of all samples.	Real number	Optional. Default value: 0.5.
<b>featureImportanceType</b>	<b>Feature Importance Type</b>	The type of feature importance. weight indicates the number of times that a feature splits. gain indicates information gain brought by the feature. cover indicates the number of samples that the feature covers on the splitting nodes.	weight, gain, and cover	Optional. Default value: gain.

• **Note**

- Specify different values for the objective parameter in different learning models. On the multiclass classification Web GUI, the objective function is automatically specified and invisible to users. On the command line, set the objective parameter to `multi:softprob`.
- Mappings between metrics and objective functions are: `mlogloss` for multiclass negative log likelihood, and `merror` for multiclass classification error.

• **Execution optimization**

Command option	Parameter	Description	Valid values	Remarks
coreNum	Cores	The number of cores. The greater the value, the faster the computing algorithm runs.	Positive integer	Optional. Automatically calculated.
memSizePer Core	Memory Size per Core (MB)	The memory size of each core, where 1024 represents 1 GB of memory.	Positive integer	Optional. Automatically calculated.

Example

• **Data generation**

The following example uses data in sparse KV format.

```
drop table if exists smart_multiclass_input;
create table smart_multiclass_input lifecycle 3 as
select
*
from
(
select 2 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as
features from dual
union all
select 1 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as
features from dual
union all
select 1 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as
features from dual
union all
select 2 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as
features from dual
union all
select 1 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as
features from dual
union all
select 1 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as
features from dual
union all
select 0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as features
from dual
union all
```

```
select 1 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as
features from dual
union all
select 0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as
features from dual
union all
select 1 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as
features from dual
) tmp;
```

The data has five dimensions of features.

- **Training**

Configure the training data and training components, as shown in [Quick start](#).

Select the label column as the target column and the features column as the feature column.

- You do not need to set the number of features because this number is calculated automatically by the algorithm. If you have a large number of features and want the algorithm to accurately estimate required resources, specify the actual number of features.
- To accelerate the training, set the number of cores on the execution optimization page. The greater the number, the faster the algorithm runs. Typically, you do not need to enter the memory size per core because the algorithm can accurately calculate the memory size. The PS algorithm starts to run only when all hosts have obtained the required resources. Therefore, you may need to wait for a longer time when the cluster is busy and resources are requested in large volumes.
- You can view the output values of the metrics in the stdout of the coordinator in a logview (HTTP link starting with <http://logview.odps.aliyun-inc.com:8080/logview>). A single PS-SMART training job can contain multiple tasks, which creates multiple logviews. Select the logview whose name starts with PS to view the output of the PS job.

Then, perform operations in the logview.



- Prediction

- Use the unified prediction component

The model generated after training is saved in binary format and can be used for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#).

If the dense format is used, you only need to select feature columns. (All columns are selected by default, and extra columns do not affect the prediction.) If the KV format is used, set the data format to sparse format and select the correct delimiter. In the SMART model, key-value pairs are separated with spaces. Therefore, the delimiter must be set to space or \u0020 (escape expression of spaces).

In the prediction\_detailcolumn, values 0, 1, and 2 indicate classes, and the values following them indicate probabilities of the corresponding classes. The predict\_result column lists the selected classes with the highest probability, and the predict\_score column lists the probability of each selected class.

- Use the PS-SMART prediction component

The output model table obtained after training is saved in binary format and can be used by the PS-SMART prediction component for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#). Set the required parameters, including the data format, feature columns, target column, and number of classes. The ID column can only be a string type column other than a feature column or a target column. The loss function must be explicitly set to multi:softprob.

The score\_class\_k columns list probabilities of class k. The class with the highest probability is the predicted class. The leaf\_index column lists the predicted leaf node numbers. Each sample has  $N \times M$  numbers, where N is the number of decision trees, and M is the number of classes. In this example, each sample has 15 numbers ( $5 \times 3 = 15$ ). Each tree is mapped to a number, which indicates the leaf node number of the sample on this tree.



**Note:**

- The output model table is a binary table that is not readable and is used to support the PS-SMART prediction component. The table provides outputs

such as leaf node numbers and evaluation indicators. However, the output table has strict requirements on data formats, which negatively affects user experience. This component is being continually improved, and may be replaced by another component in the future.

- A string type column must be selected as the label column. You can enter strings in the column but cannot be blank or NULL. A feature column can be converted to the string type by using the data type conversion component.
- The loss function must be explicitly set to multi:softprob. By default, the loss function does not work.

- View feature importance

To view feature importance, you can export the third output port to an output table, or right-click PS-SMART training component and choose View Data > Output Feature Importance Table from the shortcut menu. The following figure shows the output feature importance table.

order ▲	id ▲	value ▲
1	1	0.276059627532959
2	3	0.20854459702968597
3	4	0.31002077460289
4	5	0.20537501573562622

In the table, the ID column lists the numbers of input features. In this example, the data is in KV format, and the IDs represent keys in key-value pairs. If the dense format is used and input features are f0,f1,f2,f3,f4,f5, ID 0 represents f0 and ID 4 represents f4. Each value indicates a feature importance type. The default value is gain, indicating the sum of information gains brought by a feature in the model. The preceding figure shows only four features because only these four features are used during the tree split process. In this case, the importance of unused features is 0.

## FAQ

- Q: Does PS\_SMART support non-numerical features and tags?
- A: No.
- Q: What is the scale of features supported by PS-SMART? Can we use large-scale 0 -1 features?

- **A:** Although PS-SMART supports tasks that contain hundreds of thousands of features, such tasks consume large amounts of resources and run slowly. Therefore, we recommend that you do not use a large number of features. The GBDT algorithm is suitable for training with continuous features. The categorical features require one-hot coding (to filter out infrequent features) before they can be used for training. The continuous numerical features can be used for training with the GBDT algorithm directly. Discretization is not recommended for numerical features.
- **Q:** Why is the result different every time although the SMART algorithm has the same data and the same parameter settings?
- **A:** The PS-SMART algorithm applies randomness in many scenarios. For example, the `data_sample_ratio` and `fea_sample_ratio` items introduce data and feature sampling respectively. In addition, the PS-SMART algorithm uses histograms to show similarity. When multiple workers run in a cluster in distributed mode, local sketches are merged to global sketches in a random order. Although different merging orders result in different tree structures, this does not introduce too much variation to the output model. Therefore, it is normal situation to obtain different results after the algorithm runs multiple times with the same data and same parameter settings.

**Note:**

- The target column in a PS-SMART multiclass classification model supports only positive integer IDs (class numbers are 0, 1, 2, ..., n-1, where n is the number of classes). Even if the values in the MaxCompute table are strings, they are saved as numerical values. If the classification target is a type string similar to Good, Medium, or Bad, convert it into a numeric value (0, 1, 2, ..., n-1).
- In the key-value format, feature IDs must be positive integers, and feature values must be real numbers. If feature IDs are strings, use the serialization component to serialize them. If the feature values are classification type strings, perform feature engineering, such as discretization.

### 8.3.6.3 K-means clustering

K-means clustering is a widely used algorithm that is used to divide n objects into k clusters while maintaining high similarity within each cluster. Similarity is calculated based on the average value of objects in a cluster. This algorithm is

similar to the expectation maximization algorithm for calculating mixed normality distribution, as both algorithms try to find the natural clustering center in data. K-means clustering randomly selects k objects. Each object represents the average value or center of a cluster. Based on its distance from each cluster center, each remaining object is then assigned to the nearest cluster and the average value of each cluster is re-calculated. This process is repeated until the criterion function converges. This algorithm assumes that object properties are from the spatial vector. Its objective is to minimize the sum of the mean square deviance inside each group.

#### Parameter settings

Table 8-42: Parameters

Parameter	Description
Clusters	The number of clusters. Default value: 10.
Distance Measurement Method	Valid values: euclidean, cityblock (the sum of absolute deviations), and cosine. Default value: euclidean.
Initial Centroid Location	Valid values: sample (randomly selected), topk (first K rows), uniform (evenly distributed and randomly generated), matrix (an initial centroid table must be specified), and kmpp (k-means++ initialization). Default value: sample.
Maximum Iterations	The maximum number of iterations. Default value: 100.
Minimum Iteration Precision	The minimum iteration precision. Default value: 0.0.

#### Procedure

1. After running the K-means Clustering component, you can view the cluster center table.

**Cluster center table:** The number of columns in this table is equal to the total number of columns selected from the input table. The number of rows is equal to the number of clusters, with each row representing a cluster center location.

## 2. Right-click the target table and choose View Data to view the cluster index table (idxTablename).

- **Cluster index table:** The number of rows is equal to the total number of rows in the input table. The value in each row represents the cluster index of the point in the corresponding row of the input table.
- **The names of all columns are displayed.** A classification marking column is appended to the table.
- **0, 1, 2, 3 are classification IDs.**
- **You can also use the table name generated by PAI command to view the cluster center table, cluster index table, and cluster count table in IDE.**



### Note:

If `matrix` is selected as the initial centroid location, you must define the initial centroid table, with the same columns as the original table. The number of rows is the same as the number of clusters. When you prepare the table, configure `k` centers and use SQL or MapReduce for sampling, or select another method based on your requirements.

### PAI command

```
PAI -name KMeans
-project algo_public
-DcenterCount="10"
-DidxTableName="bank_data_index"
-DdistanceType="euclidean"
-DappendColsIndex="0,1,2,3,4,5,6,7,8,9,10"
-DcenterTableName="pai_temp_3300_27298_3"
-Dloop="100"
-DclusterCountTableName="pai_temp_3300_27298_2"
-DinitCentersMethod="sample"
-Daccuracy="0.0"
-DinputTableName="bank_data"
-DselectedColNames="cons_conf_idx,emp_var_rate,euribor3m,pdays,previous";
```

Table 8-43: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <code>algo_public</code> . If you change the name, the system reports an error.

Parameter	Description
<b>centerCount</b>	The number of clusters. The value must be an integer. Default value: 10.
<b>idxTableName</b>	The name of the output cluster index table. The number of rows is equal to the total number of rows in the input table. The value in each row represents the cluster index of the point in the corresponding row of the input table.
<b>distanceType</b>	Optional. The method used to measure the distance. Valid values: euclidean, cityblock, and cosine. Default value: euclidean.
<b>appendColsIndex</b>	Optional. The name of the ID column appended to the output table. No ID column is appended to the output table by default.
<b>centerTableName</b>	The name of the output cluster center table. The number of columns in this table is equal to the total number of columns selected from the input table. The number of rows is equal to the number of clusters, with each row representing a cluster center location.
<b>loop</b>	Optional. The maximum number of iterations. The value must be an integer. Default value: 100.
<b>clusterCountTableName</b>	The name of the cluster point count table. The number of rows is equal to the number of clusters, which indicates the total number of cluster points in the class in each clustering centroid row.
<b>initCentersMethod</b>	Optional. The method used to determine the initial centroid location. The options include sample (randomly selected), topk (first K rows), uniform (evenly distributed and randomly generated), matrix (an initial centroid table must be specified), and kmpp (k-means++ initialization). Default value: sample.
<b>accuracy</b>	The minimum iteration precision. Default value: 0.0.
<b>inputTableName</b>	The name of the input table.
<b>selectedColNames</b>	The names of columns selected from the input table, which are separated with commas (.). Only double type is supported.
<b>initCenterTableName</b>	Optional. The name of the table that stores the initial center values. This table is not required unless <code>initCentersMethod</code> is set to <code>matrix</code> .

## 8.3.6.4 Regression

### 8.3.6.4.1 GBDT regression

Gradient boosting decision tree (GBDT) is an iterative decision tree algorithm based on multiple decision trees. The final output is the sum of conclusions of all trees. GBDT can be applied to almost all regression models (linear or nonlinear) and has a wider scope of application than logistic regression that is only applicable to linear regression.

For more information, see [A Regression Framework for Learning Ranking Functions Using Relative Relevance Judgments](#). For more information, see [GBDT binary classification](#).

PAI command

```
PAI -name gbdt
-project algo_public
-DfeatureSplitValueMaxSize="500"
-DlossType="0"
-DrandSeed="0"
-DnewtonStep="0"
-Dshrinkage="0.05"
-DmaxLeafCount="32"
-DlabelColName="campaign"
-DinputTableName="bank_data_partition"
-DminLeafSampleCount="500"
-DsampleRatio="0.6"
-DgroupIDColName="age"
-DmaxDepth="11"
-DmodelName="xlab_m_GBDT_83602"
-DmetricType="2"
-DfeatureRatio="0.6"
-DinputTablePartitions="pt=20150501"
-Dtau="0.6"
-Dp="1"
-DtestRatio="0.0"
-DfeatureColNames="previous,cons_conf_idx,euribor3m"
-DtreeCount="500"
```

Table 8-44: Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-
featureColNames	Optional. The names of feature columns selected from the input table for training.	Column name	All columns are selected by default.

Parameter	Description	Valid values	Default value
<b>labelColName</b>	<b>Optional.</b> The name of the label column selected from the input table.	<b>Column name</b>	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions are selected by default.
<b>modelName</b>	<b>Required.</b> The name of the output model.	-	-
<b>outputImportanceTableName</b>	<b>Optional.</b> The name of the output feature importance table.	-	-
<b>groupIDColName</b>	<b>Optional.</b> The name of the stratification column.	<b>Column name</b>	The whole table is selected by default.



Parameter	Description	Valid values	Default value
<b>lossType</b>	<b>Optional. The loss function type. The function types include</b> 0: GBRANK, 1: LAMBDAMART_DCG, 2: LAMBDAMART_NDCG, 3: LEAST_SQUARE, and 4: LOG_LIKELIHOOD.	0, 1, 2, 3, and 4	0
<b>metricType</b>	<b>Optional. The type of metrics. 0 (NDCG) indicates the normalized discounted cumulative gain, 1 (DCG) indicates the discounted cumulative gain, and 2 (AUC) is applicable only to 0/1 label.</b>	0, 1, and 2	2
<b>treeCount</b>	<b>Optional. The number of trees.</b>	[1, 10000]	500
<b>shrinkage</b>	<b>Optional. The learning rate.</b>	(0, 1]	0.05
<b>maxLeafCount</b>	<b>Optional. The maximum number of leaves. This value must be an integer.</b>	[2, 1000]	32
<b>maxDepth</b>	<b>Optional. The maximum depth of a tree. This value must be an integer.</b>	[1, 11]	11

Parameter	Description	Valid values	Default value
<b>minLeafSampleCount</b>	Optional. The minimum number of samples on a leaf node. This value must be an integer.	[100, 1000]	500
<b>sampleRatio</b>	Optional. The fraction of training samples.	(0, 1]	0.6
<b>featureRatio</b>	Optional. The fraction of training features.	(0, 1]	0.6
<b>tau</b>	Optional. The Tau parameter in gbrank loss.	[0, 1]	0.6
<b>p</b>	Optional. The p parameter in gbrank loss.	[1, 10]	1
<b>randSeed</b>	Optional. The random seed.	[0, 10]	0
<b>newtonStep</b>	Optional. This parameter specifies whether to use the Newton method.	0 and 1	1
<b>featureSplitValueMaxSize</b>	Optional. The maximum number of splits per feature.	[1, 1000]	500
<b>lifecycle</b>	Optional. The lifecycle of the output table.	-	No lifecycle is set by default.

## Examples

### SQL statement to generate data:

```
drop table if exists gbdt_ls_test_input;
create table gbdt_ls_test_input as select * from (
select cast(1 as double) as f0,
cast(0 as double) as f1,
cast(0 as double) as f2,
cast(0 as double) as f3,
```

```
cast(0 as bigint) as label from dual union all
select cast(0 as double) as f0,
cast(1 as double) as f1,
cast(0 as double) as f2,
cast(0 as double) as f3,
cast(0 as bigint) as label from dual union all
select cast(0 as double) as f0,
cast(0 as double) as f1,
cast(1 as double) as f2,
cast(0 as double) as f3,
cast(1 as bigint) as label from dual union all
select cast(0 as double) as f0,
cast(0 as double) as f1,
cast(0 as double) as f2,
cast(1 as double) as f3,
cast(1 as bigint) as label from dual union all
select cast(1 as double) as f0,
cast(0 as double) as f1,
cast(0 as double) as f2,
cast(0 as double) as f3,
cast(0 as bigint) as label from dual union all
select cast(0 as double) as f0,
cast(1 as double) as f1,
cast(0 as double) as f2,
cast(0 as double) as f3,
cast(0 as bigint) as label from dual) a;
```

PAI command

- **Training:**

```
drop offlinemodel if exists gbdt_ls_test_model;
PAI -name gbdt
-project algo_public
-DfeatureSplitValueMaxSize="500"
-DlossType="3"
-DrandSeed="0"
-DnewtonStep="1"
-Dshrinkage="0.5"
-DmaxLeafCount="32"
-DlabelColName="label"
-DinputTableName="gbdt_ls_test_input"
-DminLeafSampleCount="1"
-DsampleRatio="1"
-DmaxDepth="10"
-DmetricType="0"
-DmodelName="gbdt_ls_test_model"
-DfeatureRatio="1"
-Dp="1"
-Dtau="0.6"
-DtestRatio="0"
-DfeatureColNames="f0,f1,f2,f3"
-DtreeCount="10"
```

- **Prediction:**

```
drop table if exists gbdt_ls_test_prediction_result;
PAI -name prediction
-project algo_public
-DdetailColName="prediction_detail"
-DmodelName="gbdt_ls_test_model"
-DitemDelimiter=","
```

```
-DresultColName="prediction_result"
-Dlifecycle="28"
-DoutputTableName="gbdt_ls_test_prediction_result"
-DscoreColName="prediction_score"
-DkvDelimiter=":"
-DinputTableName="gbdt_ls_test_input"
-DenableSparse="false"
-DappendColNames="label"
```

#### Input description

Table 8-45: gbdt\_ls\_test\_input

f0	f1	f2	f3	label
1.0	0.0	0.0	0.0	0
0.0	0.0	1.0	0.0	1
0.0	0.0	0.0	1.0	1
0.0	1.0	0.0	0.0	0
1.0	0.0	0.0	0.0	0
0.0	1.0	0.0	0.0	0

#### Output description

Table 8-46: gbdt\_ls\_test\_prediction\_result

label	prediction_result	prediction_score	prediction_detail
0	NULL	0.0	{"label": 0}
0	NULL	0.0	{"label": 0}
1	NULL	0.9990234375	{"label": 0.9990234375}
1	NULL	0.9990234375	{"label": 0.9990234375}
0	NULL	0.0	{"label": 0}
0	NULL	0.0	{"label": 0}

### 8.3.6.4.2 Linear regression

This component is used to resolve regression issues and analyze the linear relationship between a dependent variable and multiple independent variables. Certain columns from an input table are selected as feature columns and one

column is selected as the label column for linear regression training and linear regression model generation.

PAI command

```
PAI -name linearregression
 -project algo_public
 -DinputTableName=lm_test_input
 -DfeatureColNames=x
 -DlabelColName=y
 -DmodelName=lm_test_input_model_out;
```

Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>modelName</b>	<b>Required.</b> The name of the output model.	-	-
<b>outputTableName</b>	<b>Optional.</b> The name of the output model evaluation table.	<b>This parameter must be specified when <code>enableFitGoodness</code> is set to true.</b>	-
<b>labelColName</b>	<b>Required.</b> The name of the label column.	<b>The name must be a double or bigint type value. Only one column can be specified.</b>	-
<b>featureColNames</b>	<b>Required.</b> The name of the feature column.	<b>The name must be a double or bigint type value in dense format, or a string type value in sparse format. Multiple columns can be specified.</b>	-
<b>inputTable Partitions</b>	<b>Optional.</b> The partitions selected from the input table for training.	-	<b>No partitions are selected by default.</b>

Parameter	Description	Valid values	Default value
<b>maxIter</b>	<b>Optional. The maximum number of iterations.</b>	-	<b>100</b>
<b>epsilon</b>	<b>Optional. The minimum likelihood deviance.</b>	-	<b>0.000001</b>
<b>enableSparse</b>	<b>Optional. This parameter specifies whether the data is in sparse format.</b>	<b>true and false</b>	<b>false</b>
<b>enableFitGoodness</b>	<b>Optional. This parameter specifies whether to perform model evaluation. Model evaluation can be performed using a variety of metrics , including R-squared, adjusted R-Squared, Akaike information criterion, degrees of freedom, residual standard deviation, and deviation.</b>	<b>true and false</b>	<b>false</b>

Parameter	Description	Valid values	Default value
<b>enableCoefficientEstimate</b>	<b>Optional. This parameter specifies whether to estimate the regression coefficient. The metrics of this parameter are value t, value p, and confidence interval [2.5%, 97.5%]. This parameter takes effect only when enableFitGoodness is set to true. This parameter is ignored when enableFitGoodness is set to false.</b>	<b>true and false</b>	<b>false</b>
<b>itemDelimiter</b>	<b>Optional. The delimiter used to separate key-value pairs. This parameter takes effect only when enableSparse is set to true.</b>	<b>-</b>	<b>Use spaces on command lines and use commas (,) on webpages.</b>
<b>kvDelimiter</b>	<b>Optional. The delimiter used to separate keys and values. This parameter takes effect only when enableSparse is set to true.</b>	<b>-</b>	<b>The default delimiter is a colon (:).</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>An integer greater than or equal to -1</b>	<b>Default value: -1. This value indicates that no lifecycle is set.</b>

Parameter	Description	Valid values	Default value
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>An integer larger than 0</b>	<b>Default value : -1. This value indicates that the number of instances is determined by the amount of input data.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>(100, 65536)</b>	<b>Default value: -1. This value indicates that the memory size is determined by the amount of input data.</b>

#### Examples

- **SQL statement to generate data:**

```
drop table if exists lm_test_input;
create table lm_test_input as
select
 *
from
(
 select 10 as y, 1.84 as x1, 1 as x2, '0:1.84 1:1' as
sparsecol1 from dual
 union all
 select 20 as y, 2.13 as x1, 0 as x2, '0:2.13' as sparsecol1
from dual
 union all
 select 30 as y, 3.89 as x1, 0 as x2, '0:3.89' as sparsecol1
from dual
 union all
 select 40 as y, 4.19 as x1, 0 as x2, '0:4.19' as sparsecol1
from dual
 union all
 select 50 as y, 5.76 as x1, 0 as x2, '0:5.76' as sparsecol1
from dual
 union all
 select 60 as y, 6.68 as x1, 2 as x2, '0:6.68 1:2' as
sparsecol1 from dual
 union all
 select 70 as y, 7.58 as x1, 0 as x2, '0:7.58' as sparsecol1
from dual
 union all
 select 80 as y, 8.01 as x1, 0 as x2, '0:8.01' as sparsecol1
from dual
 union all
 select 90 as y, 9.02 as x1, 3 as x2, '0:9.02 1:3' as
sparsecol1 from dual
 union all
```



```
select 100 as y, 10.56 as x1, 0 as x2, '0:10.56' as
sparsecol1 from dual
) tmp;
```

- **PAI command**

```
PAI -name linearregression
 -project algo_public
 -DinputTableName=lm_test_input
 -DlabelColName=y
 -DfeatureColNames=x1,x2
 -DmodelName=lm_test_input_model_out
 -DoutputTableName=lm_test_input_conf_out
 -DenableCoefficientEstimate=true
 -DenableFitGoodness=true
 -Dlifecycle=1;
pai -name prediction
 -project algo_public
 -DmodelName=lm_test_input_model_out
 -DinputTableName=lm_test_input
 -DoutputTableName=lm_test_input_predict_out
 -DappendColNames=y;
```

- **Output description:**

- **When enableFitGoodness is set to true, partitions specified by p='goodness' are created in the model evaluation table. The output metrics are R-squared, adjusted R-Squared, Akaike information criterion, degrees of freedom, residual standard deviation, and deviation.**
- **When enableCoefficientEstimate is set to true, partitions specified by p='coefficient' are created in the model evaluation table. The table contains the intercepts and the name, coefficient, t-score, p-value, and confidence interval [2.5%, 97.5%] of the features.**
- **Output model evaluation table: lm\_test\_input\_conf\_out.**

colname	value	tscore	pvalue	confidenceinterval	p
Intercept	-6.42378496687763	-2.2725755951390028	0.06	{"2.5%": -11.964027, "97.5%": -0.883543}	coefficient
x1	10.260063429838898	23.270944360826963	0.0	{"2.5%": 9.395908, "97.5%": 11.124219}	coefficient
x2	0.35374498323846265	0.2949247320997519	0.81	{"2.5%": -1.997160, "97.5%": 2.704650}	coefficient

colname	value	tscore	pvalue	confidenceinterval	p
rsquared	0.9879675667384592	NULL	NULL	NULL	goodness
adjusted_r_squared	0.9845297286637332	NULL	NULL	NULL	goodness
aic	59.331109494251805	NULL	NULL	NULL	goodness
degree_of_freedom	7.0	NULL	NULL	NULL	goodness
standardEr_r_residual	3.765777749448906	NULL	NULL	NULL	goodness
deviance	99.26757440771128	NULL	NULL	NULL	goodness

- Output prediction table: lm\_test\_input\_predict\_out.

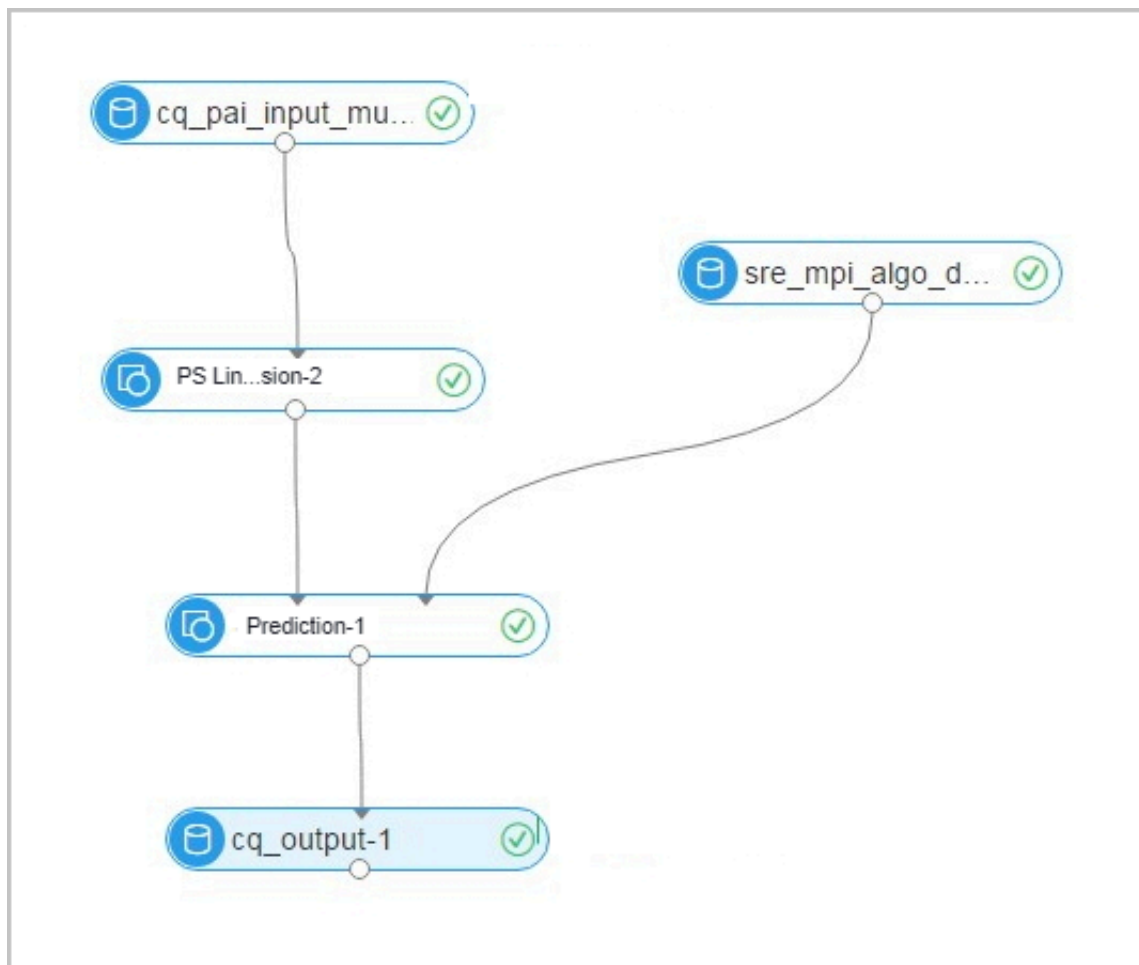
y	prediction_result	prediction_score	prediction_detail
10	NULL	12.808476727264404	{"y": 12.8084767272644}
20	NULL	15.43015013867922	{"y": 15.43015013867922}
30	NULL	33.48786177519568	{"y": 33.48786177519568}
40	NULL	36.565880804147355	{"y": 36.56588080414735}
50	NULL	52.674180388994415	{"y": 52.67418038899442}
60	NULL	62.82092871092313	{"y": 62.82092871092313}
70	NULL	71.34749583130122	{"y": 71.34749583130122}
80	NULL	75.75932310613193	{"y": 75.75932310613193}

y	prediction _result	prediction_score	prediction_detail
90	NULL	87.1832221199846	{"y": 87.18322211998461 }
100	NULL	101.92248485222113	{"y": 101.9224848522211 }

#### 8.3.6.4.3 PS linear regression

Linear regression is a classic regression algorithm used to analyze the linear relationship between a dependent variable and multiple independent variables. Parameter servers (PSs) are used to run large amounts of training tasks online and offline. Parameter servers can use hundreds of billions of samples to efficiently train billions of feature models. The PS linear regression model can run training tasks with hundreds of billions of samples and billions of features, and supports L1 and L2 regular expressions.

Quick start



## PAI command

### • Training

```
PAI -name ps_linearregression
 -project algo_public
 -DinputTableName="lm_test_input"
 -DmodelName="linear_regression_model"
 -DlabelColName="label"
 -DfeatureColNames="features"
 -Dl1Weight=1.0
 -Dl2Weight=0.0
 -DmaxIter=100
 -Depsilon=1e-6
 -DenableSparse=true
```

### • Prediction

```
drop table if exists logistic_regression_predict;
PAI -name prediction
 -DmodelName="linear_regression_model"
 -DoutputTableName="linear_regression_predict"
 -DinputTableName="lm_test_input"
 -DappendColNames="label,features"
 -DfeatureColNames="features"
 -DenableSparse=true
```

## Parameters

### • Data parameters

Command option	Parameter	Description	Valid values	Default value
featureColNames	Feature Columns	<b>Required.</b> The names of feature columns selected from the input table for training.	If a column name is in dense format , it must be of the bigint or double type. If the column name is in sparse KV format, it must be a string.	-

Command option	Parameter	Description	Valid values	Default value
<b>labelColName</b>	<b>Label Column</b>	Required. The name of the label column selected from the input table.	The column name must be of the bigint or double type.	-
<b>enableSparse</b>	<b>Use Sparse Format</b>	Optional. If you choose to use the sparse KV format, do not use feature ID 0. We recommend that the feature IDs start from 1.	true and false	false
<b>itemDelimiter</b>	<b>KV Pair Delimiter</b>	Optional. The delimiter used to separate key-value pairs when data in the input table is in sparse format.	Symbol	The default delimiter is a space.
<b>kvDelimiter</b>	<b>KV Delimiter</b>	Optional. The delimiter used to separate keys and values when data in the input table is in sparse format.	Symbol	The default delimiter is a colon (:).
<b>inputTableName</b>	<b>Input Table Name</b>	Required.	Table name	-
<b>modelName</b>	<b>Output Model Name</b>	Required.	Model name	-

Command option	Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	<b>Input Table Partitions</b>	Optional.	Partition name	The parameter value must be in the ds=1/pt=1 format.
<b>enableModelIo</b>	<b>Output to Offline Model</b>	Optional. When this parameter is set to false , the data is output to a MaxCompute table where you can view model weights .	true and false	true

• Algorithm parameters

Command option	Parameter	Description	Valid values	Default value
<b>l1Weight</b>	<b>L1 Weight</b>	Optional. The L1 regularization coefficient. The larger this value is, the fewer non-zero elements a model has. To overfit the model, set this parameter to a larger value.	A non-negative real number	1.0

Command option	Parameter	Description	Valid values	Default value
<b>l2Weight</b>	<b>L2 Weight</b>	Optional. The L2 regularization coefficient. The larger this value is, the smaller the absolute values of the model parameters are. To overfit the model, set this parameter to a larger value.	A non-negative real number	0
<b>maxIter</b>	<b>Maximum Iterations</b>	Optional. The maximum number of LBFGS/OWL-QN iterations. Value 0 indicates that no limit is set.	A non-negative integer	100

Command option	Parameter	Description	Valid values	Default value
<b>epsilon</b>	<b>Minimum Convergence Deviance</b>	Optional. The mean of the relative loss change rates in ten iterations, which is used as a condition to determine whether to terminate the optimization algorithm. The smaller this value is, the stricter the condition is, and the longer the algorithm runs.	A real number between 0 and 1	1.0e-06



Command option	Parameter	Description	Valid values	Default value
modelSize	Largest Feature ID	Optional. The largest feature ID among all feature IDs (feature dimension ). It can be larger than the actual largest feature ID. The larger this value is , the higher the memory usage is. If you leave this parameter empty, the system starts an SQL task to calculate the largest feature ID automatically.	A non-negative integer	0



**Note:**

Both the maximum iterations and minimum convergence deviance determine when the algorithm stops. If both parameters are set, the algorithm stops when one of the conditions is met.

• Execution optimization

Command option	Parameter	Description	Valid values	Default value
coreNum	Cores	Optional. The number of cores. The larger this value is, the faster the computing algorithm runs.	A positive integer	Automatically allocated.
memSizePerCore	Memory Size per Core (MB)	Optional. The memory size of each core, where 1024 represents 1 GB of memory.	A positive integer	Automatically allocated. Typically, you do not need to set this parameter because the algorithm can accurately estimate the memory size required.

Examples

• Data generation

The following example uses data in sparse KV format:

```
drop table if exists lm_test_input;
create table lm_test_input as
select
*
from
(
select 2 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as
features from dual
union all
select 1 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as
features from dual
union all
select 1 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as
features from dual
union all
select 2 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as
features from dual
union all
```

```
select 1 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as
features from dual
union all
select 1 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as
features from dual
union all
select 0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as features
from dual
union all
select 1 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as
features from dual
union all
select 0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as
features from dual
union all
select 1 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as
features from dual
) tmp;
```

The feature IDs start from 1, and the maximum feature ID is 5.

#### • Training

Configure the training data and training components based on [Quick start](#). Select the label column as the target column and features column as the feature column. Then, select the sparse data format.

- You can retain the default value 0 for the largest feature ID. The algorithm can start an SQL task to calculate the largest feature ID automatically. If you do not want to start the SQL task, enter a value greater than 5. This value indicates the number of feature columns in dense format and indicates the largest feature ID in KV format.
- To accelerate the training, you can set the number of cores on the tuning page . The larger the number is, the faster the algorithm runs. Typically, you do not need to enter the memory size per core because the algorithm can accurately calculate the memory size. The PS algorithm starts to run only when all hosts have obtained the resources. Therefore, you may need to wait a longer period of time when the cluster is busy and resources are requested in large volume.

#### • Prediction

The model generated after training is saved in binary format and can be used for prediction. Configure the input settings (model and testing data) for the prediction component and set parameters based on [Quick start](#).

Select the KV format for training and set a correct delimiter. When the KV format is used, key-value pairs are separated by spaces. Therefore, the delimiter must be set to space or `\u0020` (the escape expression of space).

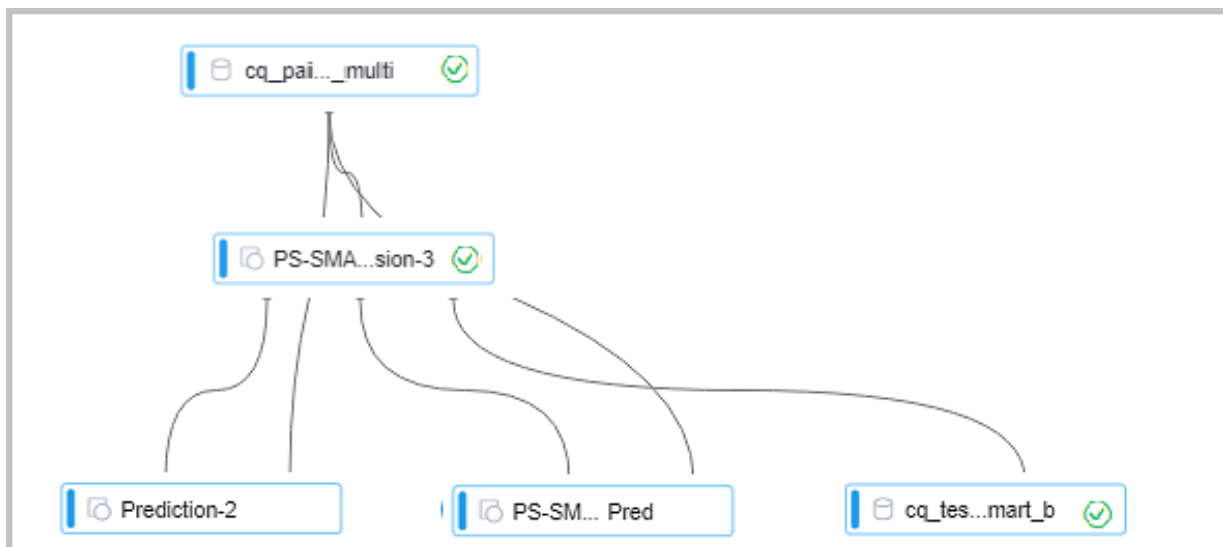
## Restrictions and guidelines

In the key-value format, feature IDs must be positive integers, and feature values must be real numbers. If feature IDs are strings, use the serialization component to serialize them. If the feature values are classification type strings, perform feature engineering, such as discretization.

### 8.3.6.4.4 PS-SMART regression

A *parameter server* (PS) is used to train a large number of models online and offline. Scalable Multiple Additive Regression Tree (SMART) is an implementation of Gradient boosting decision tree (GBDT) on PS. PS-SMART can run training tasks containing up to tens of billions of samples and hundreds of thousands of features on thousands of nodes. It also supports failover for high stability. PS-SMART supports various data formats, training targets, evaluation targets, output feature importance, and histogram approximation for training acceleration.

## Quick start



As shown in the figure, a PS-SMART regression model is learned based on training data. The model has three output ports:

- **Output model:** offline model, which is connected to the unified prediction component. This model does not support the output of leaf node numbers.
- **Output model table:** a binary table that is not readable and is used to ensure compatibility with the PS-SMART prediction component. The table provides outputs such as leaf node numbers and evaluation metrics. However, the output table has strict requirements on data formats, which negatively affects user

experience. This component is being continually improved, and may be replaced by another component in the future.

- **Output feature importance table:** lists the importance of each feature. Three importance types are supported. For more information, see [Parameters](#).

PAI command

- **Training**

```
PAI -name ps_smart
 -project algo_public
 -DinputTableName="smart_regression_input"
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545859_2"
 -DoutputImportanceTableName="pai_temp_24515_545859_3"
 -DlabelColName="label"
 -DfeatureColNames="features"
 -DenableSparse="true"
 -Dobjective="reg:linear"
 -Dmetric="rmse"
 -DfeatureImportanceType="gain"
 -DtreeCount="5";
 -DmaxDepth="5"
 -Dshrinkage="0.3"
 -DL2="1.0"
 -DL1="0"
 -Dlifecycle="3"
 -DsketchEps="0.03"
 -DsampleRatio="1.0"
 -DfeatureRatio="1.0"
 -DbaseScore="0.5"
 -DminSplitLoss="0"
```

- **Prediction**

```
PAI -name prediction
 -project algo_public
 -DinputTableName="smart_regression_input";
 -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
 -DoutputTableName="pai_temp_24515_545860_1"
 -DfeatureColNames="features"
 -DappendColNames="label, features"
 -DenableSparse="true"
```

-Dlifecycle="28"

## Parameters

### • Data parameters

Command option	Parameter	Description	Valid values	Remarks
<b>featureColNames</b>	<b>Feature Column</b>	The names of feature columns selected from the input table for training.	If the column name is in dense format , it must be of the bigint or double type. If the column name is in sparse KV format, it must be a string, and its keys and values must be numeric.	Required
<b>labelColName</b>	<b>Label Column</b>	The name of the label column selected from the input table .	The column name can be of either string or numeric type , but only numeric data can be stored in the columns . For example , the column value can be 0 or 1 for regression.	Required
<b>weightCol</b>	<b>Weight Column</b>	This column specifies the weight of each sample.	The column name can be of the numeric type.	Optional. Default value: null.

Command option	Parameter	Description	Valid values	Remarks
<b>enableSparse</b>	<b>Use Sparse Format</b>	This parameter specifies whether the data in the input table is in sparse format, in which key-value pairs are separated by spaces whereas keys and values are separated by colons (:), for example, 1:0.3 3:0.9.	true, false	Optional. Default value: false.
<b>inputTableName</b>	<b>Input Table Name</b>	N/A	N/A	Required
<b>modelName</b>	<b>Output Model Name</b>	N/A	N/A	Required
<b>outputImportanceTableName</b>	<b>Output Feature Importance Table Name</b>	N/A	N/A	Optional. Default value: null.
<b>inputTablePartitions</b>	<b>Input Table Partitions</b>	N/A	N/A	Optional. The parameter value must be in ds=1/pt=1 format.

Command option	Parameter	Description	Valid values	Remarks
<b>outputTable</b>	<b>Output Model Table Name</b>	The output table is a MaxCompute table that uses the binary format and is not readable. The prediction component that comes with SMART can be used to generate leaf node numbers .	String	Optional
<b>lifecycle</b>	<b>Output Table Lifecycle</b>	N/A	Positive integer	Optional. Default value: 3.

• **Algorithm parameters**

Command option	Parameter	Description	Valid values	Remarks
<b>objective</b>	<b>Objective Function Type</b>	The objective function type affects learning and must be selected properly. Multiple loss functions are available for regression . For more information, see the notes.		Required. The default type is Linear regression.



Command option	Parameter	Description	Valid values	Remarks
<b>metric</b>	<b>Evaluation Indicator Type</b>	Evaluation indicators in the training set, which must correspond to the objective function type and are exported to stdout of the coordinator in a logview . For more information, see the following notes and samples.		Optional. Default value: null.
<b>treeCount</b>	<b>Trees</b>	The number of trees. The training time is proportional to this number.	Positive integer	Optional. Default value: 1.
<b>maxDepth</b>	<b>Maximum Decision Tree Depth</b>	The maximum depth of a tree. We recommend that you set this value to 5 , which means the tree can contain up to 32 leaf nodes.	A positive integer in the range of [1, 20]	Optional. Default value: 5.

Command option	Parameter	Description	Valid values	Remarks
<b>sampleRatio</b>	<b>Data Sampling Fraction</b>	The data sampling rate when trees are built. The sample data is used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means data sampling is disabled.
<b>featureRatio</b>	<b>Feature Sampling Fraction</b>	The feature sampling rate when trees are built. The sample features are used to build a weak learner to accelerate training.	(0, 1]	Optional. The default value is 1.0, which means feature sampling is disabled.
<b>l1</b>	<b>L1 Penalty Coefficient</b>	This parameter determines the number of leaf nodes. The greater the value, the fewer the leaf nodes. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 0.

Command option	Parameter	Description	Valid values	Remarks
l2	L2 Penalty Coefficient	This parameter determines the size of a leaf node. The greater the value, the more evenly the leaf nodes are distributed. You can set this parameter to a greater value if overfitting occurs.	Non-negative real number	Optional. Default value: 1.0.
shrinkage	Learning Rate	N/A	(0, 1]	Optional. Default value: 0.3.
sketchEps	Sketch-based Approximate Precision	The threshold for selecting quantiles when you build a sketch. The number of buckets is $O(1.0/\text{sketchEps})$ . The smaller the parameter value, the more buckets are generated. Typically, you do not need to change this value.	(0, 1)	Optional. Default value: 0.03.

Command option	Parameter	Description	Valid values	Remarks
<b>minSplitLoss</b>	<b>Minimum Split Loss</b>	The minimum split loss changes required for splitting a node. The greater the value, the more conservatively the node splits.	Non-negative real number	Optional. Default value: 0.
<b>featureNum</b>	<b>Features</b>	The number of features or the maximum feature ID. Specify this parameter for resource usage estimation.	Positive integer	Optional
<b>baseScore</b>	<b>Global Offset</b>	Original predicted values of all samples.	Real number	Optional. Default value: 0.5.

Command option	Parameter	Description	Valid values	Remarks
<b>featureImportanceType</b>	<b>Feature Importance Type</b>	The type of feature importance. weight indicates the number of times that a feature splits. gain indicates information gain brought by the feature. cover indicates the number of samples that the feature covers on the splitting nodes.	weight, gain, and cover	Optional. Default value: gain.
<b>tweedieVarPower</b>	<b>Tweedie Distribution Index</b>	Tweedie distribution index indicates the relationship between the variance and mean. For example, $\text{Var}(y) \sim E(y)^{\text{tweedie\_variance\_power}}$ .	(1, 2)	Optional. Default value: 1.5.

• **Note**

- Specify different values for the objective parameter in different learning models. The regression Web GUI provides multiple objective functions.

```
reg:linear (Linear regression) // The range of label numbers is (-∞, +∞).
reg:logistic (Logistic regression) // The range of label numbers is [0, 1].
```

```
count:poisson (Poisson regression for count data, output mean of
poisson distribution) // Label numbers must be greater than 0.
reg:gamma (Gamma regression for modeling insurance claims severity
, or for any outcome that might be [gamma-distributed](https://en
.wikipedia.org/wiki/Gamma_distribution#Applications)) // Label
numbers must be greater than 0.
reg:tweedie (Tweedie regression for modeling total loss in
insurance, or for any outcome that might be [Tweedie-distributed](
https://en.wikipedia.org/wiki/Tweedie_distribution#Applications).)
// Label numbers must be greater than or equal to 0.
```

- **Metrics for these objective functions are:**

```
rmse (rooted mean square error, corresponding to objective reg:
linear)
mae (mean absolute error, corresponding to objective reg:linear)
poisson-nloglik (negative loglikelihood for poisson regression,
corresponding to objective count:poisson)
gamma-deviance (Residual deviance for gamma regression, correspond
ing to objective reg:gamma)
gamma-nloglik (Negative log-likelihood for gamma regression,
corresponding to objective reg:gamma)
tweedie-nloglik (tweedie-nloglik@1.5, negative log-likelihood
for Tweedie regression, at a specified value of the tweedie_va
riance_power parameter)
```

• **Execution optimization**

Command option	Parameter	Description	Valid values	Remarks
coreNum	Cores	The number of cores. The greater the value, the faster the computing algorithm runs.	Positive integer	Optional. Automatically allocated.
memSizePer Core	Memory Size per Core (MB)	The memory size of each core, where 1024 represents 1 GB of memory .	Positive integer	Optional. Automatically allocated.

## Example

- **Data generation**

The following example uses data in sparse KV format.

```
drop table if exists smart_regression_input;
create table smart_regression_input as
select
*
from
(
select 2.0 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as
features from dual
union all
select 1.0 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as
features from dual
union all
select 1.0 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as
features from dual
union all
select 2.0 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as
features from dual
union all
select 1.0 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as
features from dual
union all
select 1.0 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as
features from dual
union all
select 0.0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as
features from dual
union all
select 1.0 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as
features from dual
union all
select 0.0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as
features from dual
union all
select 1.0 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as
features from dual
) tmp;
```

Feature IDs are numbered starting from 1, and the maximum feature ID is 5.

- **Training**

Select the label column as the target column and the features column as the feature column.

- You do not need to set the number of features because this number is calculated automatically by the algorithm. If you have a large number of features and want the algorithm to accurately estimate the amount of required resources, specify the actual number of features.
- To accelerate the training, set the number of cores on the execution optimization page. The greater the number, the faster the algorithm runs. Typically,

you do not need to enter the memory size per core because the algorithm can accurately calculate the memory size. The PS algorithm starts to run only when all hosts have obtained the required resources. Therefore, you may need to wait for a longer time when the cluster is busy and resources are requested in large volumes.

- You can view the output values of the metrics in the stdout of the coordinator in a logview (HTTP link starting with <http://logview.odps.aliyun-inc.com:8080/logview>). A single PS-SMART training job can contain multiple tasks, and therefore multiple logviews are created. Select the logview whose name starts with PS to view the output of the PS job.

- Prediction

- Use the unified prediction component

The model generated after training is saved in binary format and can be used for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#).

If the dense format is used, you only need to select feature columns. (All columns are selected by default, and extra columns do not affect the prediction.) If the KV format is used, set the data format to sparse format and select the correct delimiter. In the SMART model, key-value pairs are separated with spaces. Therefore, the delimiter must be set to space or \u0020 (escape expression of spaces).

- Use the PS-SMART prediction component

The output model table obtained after training is saved in binary format and can be used by the PS-SMART prediction component for prediction. Configure the input model and test data for the prediction component, as shown in [Quick start](#). Set the required parameters, including the data format, feature columns, target column, and number of classes. The ID column can only be a string type column other than a feature column or a target column. The loss function must be explicitly set to the objective function used for training.

The prediction\_score column lists the predicted values. The leaf\_index column lists the predicted leaf node numbers. Each sample has N numbers,



where N is the number of decision trees. Each tree is mapped to a number, which indicates the leaf node number of the sample on this tree.



**Note:**

- The output model table is a binary table that is not readable and is used to support the PS-SMART prediction component. The table provides outputs such as leaf node numbers and evaluation indicators. However, the output table has strict requirements on data formats, which negatively affects user experience. This component is being continually improved, and may be replaced by another component in the future.
- A string type column must be selected as the label column. You can enter strings in the column but cannot be blank or NULL. A feature column can be converted to the string type by using the data type conversion component.
- The loss function must be explicitly set to the objective function used for training. By default, the loss function does not work.

- View feature importance

To view feature importance, you can export the third output port to an output table, or right-click PS-SMART training component and choose View Data > Output Feature Importance Table from the shortcut menu.

order ▲	id ▲	value ▲
1	1	0.14059734344482422
2	4	0.8594027161598206

In the table, the ID column lists the numbers of input features. In this example, the data is in KV format and the IDs represent keys in key-value pairs. If the dense format is used and input features are f0,f1,f2,f3,f4,f5, ID 0 represents f0, and ID 4 represents f4. Each value indicates a feature importance type. The default value is gain, indicating the sum of information gains brought by a feature in the model. The preceding figure shows only two features because only these two features are used during the tree split process. In this case, the importance of unused features is 0.

## FAQ

- Q: Does PS\_SMART support non-numerical features and tags?

- **A: No.**
- **Q: What is the scale of features supported by PS-SMART? Can we use large-scale 0-1 features?**
- **A: Although PS-SMART supports tasks that contain hundreds of thousands of features, such tasks consume large amounts of resources and run slowly. Therefore, we recommend that you do not use a large number of features. The GBDT algorithm is suitable for training with continuous features. The categorical features require one-hot coding (to filter out infrequent features) before they can be used for training. The continuous numerical features can be used for training with the GBDT algorithm directly. Discretization is not recommended for numerical features.**
- **Q: Why is the result different every time although the SMART algorithm has the same data and the same parameter settings?**
- **A: The PS-SMART algorithm applies randomness in many scenarios. For example, the data\_sample\_ratio and fea\_sample\_ratio items introduce data and feature sampling respectively. In addition, the PS-SMART algorithm uses histograms to show similarity. When multiple workers run in a cluster in distributed mode, local sketches are merged to global sketches in a random order. Although different merging orders result in different tree structures, this does not introduce too much variation to the output model. Therefore, it is normal situation to obtain different results after the algorithm runs multiple times with the same data and same parameter settings.**



**Note:**

- **The target column in a PS-SMART regression model supports only numerical values. Even if values in the MaxCompute table are strings, they are saved as numerical values.**
- **In the key-value format, feature IDs must be positive integers, and feature values must be real numbers. If feature IDs are strings, use the serialization component to serialize them. If feature values are classification type strings, perform feature engineering, such as discretization.**

### 8.3.6.5 Collaborative filtering (etrec)

etrec is an item-based collaborative filtering algorithm that takes two input columns and provides the top N items that have the highest similarity.

Set the user and item columns.

- You can configure three similarity types.
- **topN** indicates the first N items with the highest similarity.
- **Calculation method:** the method used to calculate items that appear multiple times.

PAI command

```
PAI -name pai_etrec
-project algo_public
-DsimilarityType="wbcosine"
-Dweight="1"
-DminUserBehavior="2"
-Dlifecycle="28"
-DtopN="2000"
-Dalpha="0.5"
-DoutputTableName="etrec_test_result"
-DmaxUserBehavior="500"
-DinputTableName="etrec_test_input"
-Doperator="add"
-DuserColName="user"
-DitemColName="item"
```

Parameters

Table 8-47: Parameters

Parameter	Description	Valid value	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	N/A	N/A
<b>userColName</b>	<b>Required. The name of the input table column selected as the user column.</b>	N/A	N/A
<b>itemColName</b>	<b>The name of the input table column selected as the item column.</b>	N/A	N/A

Parameter	Description	Valid value	Default value
<b>payloadColName</b>	Optional. The name of the input table column selected as the payload column.	N/A	No payload column is set by default.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training.	N/A	The whole table is selected by default.
<b>outputTableName</b>	Required. The name of the output table.	N/A	N/A
<b>outputTablePartition</b>	Optional. The partitions in the output table.	N/A	The output table is non-partitioned by default.
<b>similarityType</b>	Optional. The type of similarity.	wbcosine, asymcosine, and jaccard	wbcosine
<b>topN</b>	Optional. N items with the highest similarity.	[1, 10000]	2000
<b>minUserBehavior</b>	Optional. The minimum user behavior.	[2,)	2
<b>maxUserBehavior</b>	Optional. The maximum user behavior.	[2, 100000]	500
<b>itemDelimiter</b>	Optional. The delimiter used to separate items in the output table.	N/A	The default delimiter is a space .
<b>kvDelimiter</b>	Optional. The delimiter used to separate keys and values in the output table.	N/A	The default delimiter is a colon (:).

Parameter	Description	Valid value	Default value
<b>alpha</b>	<b>Optional. The value of the smoothing factor for asymcosine.</b>	N/A	<b>0.5</b>
<b>weight</b>	<b>Optional. The weight used for asymcosine.</b>	N/A	<b>1.0</b>
<b>operator</b>	<b>Optional. The action to be performed when the same items exist for one user.</b>	<b>add, mul, min, and max</b>	<b>add</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	N/A	<b>1</b>

#### Examples

- **SQL statement to generate data:**

```
drop table if exists etrec_test_input;
create table etrec_test_input as select * from
(
 select cast(0 as string) as user,
 cast(0 as string) as item from dual
union all
 select cast(0 as string) as user,
 cast(1 as string) as item from dual
union all
 select cast(1 as string) as user,
 cast(0 as string) as item from dual
union all
 select cast(1 as string) as user,
 cast(1 as string) as item from dual) a;
```

- **PAI command**

```
drop table if exists etrec_test_result;
PAI -name pai_etrec
 -project algo_public
 -DsimilarityType="wbcosine"
 -Dweight="1"
 -DminUserBehavior="2"
 -Dlifecycle="28"
 -DtopN="2000"
 -Dalpha="0.5"
 -DoutputTableName="etrec_test_result"
 -DmaxUserBehavior="500"
 -DinputTableName="etrec_test_input"
 -Doperator="add"
 -DuserColName="user"
```

```
-DitemColName="item"
```

- **Input description**

Table 8-48: etrec\_test\_input

User	Item
0	0
0	1
1	0
1	1

- **Output description**

Table 8-49: etrec\_test\_result

Item ID	Similarity
0	1:1
1	0:1

### 8.3.6.6 Evaluation

#### 8.3.6.6.1 Regression model evaluation

You can evaluate a regression model based on the predicted and actual results.

Indicators include SST, SSE, SSR, R2, R, MSE, RMSE, MAE, MAD, MAPE, count, yMean, and predictMean.

PAI command

```
Pai -name regression_evaluation
-project algo_public
-DinputTableName=input_table
-DyColName=y_col
-DpredictionColName=prediction_col
-DoutputTableName=output_table;
```

Table 8-50: Parameters

Parameter	Description	Default value
inputTableName	Required. The name of the input table.	-

Parameter	Description	Default value
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training .	<b>All partitions of the input table are selected by default.</b>
<b>yColName</b>	<b>Required.</b> The name of the original dependent variable column in the input table. It must be a numerical value.	-
<b>predictionColName</b>	<b>Required.</b> The name of the predicted dependent variable column. It must be a numerical value.	-
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	-
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	<b>No lifecycle is set by default.</b>

## Output

The following table describes the JSON columns.

Table 8-51: Column description

Column	Description
<b>SST</b>	<b>The sum of squares total.</b>
<b>SSE</b>	<b>The sum of squares error.</b>
<b>SSR</b>	<b>The sum of squares regression.</b>
<b>R2</b>	<b>The coefficient of determination.</b>
<b>R</b>	<b>The coefficient of multiple correlations.</b>
<b>MSE</b>	<b>The mean squared error.</b>
<b>RMSE</b>	<b>The root-mean-square error.</b>
<b>MAE</b>	<b>The mean absolute error.</b>
<b>MAD</b>	<b>The mean absolute difference.</b>
<b>MAPE</b>	<b>The mean absolute percentage error.</b>
<b>count</b>	<b>The number of rows.</b>

Column	Description
yMean	The mean of original dependent variables.
predictionMean	The mean of prediction results.

### 8.3.6.6.2 Clustering model evaluation

You can evaluate clustering models, including metrics and icons, based on raw data and clustering models.

PAI command

```
PAI -name cluster_evaluation
-project algo_public
-DinputTableName=pai_cluster_evaluation_test_input
-DselectedColNames=f0,f3
-DmodelName=pai_kmeans_test_model
-DoutputTableName=pai_ft_cluster_evaluation_out;
```

Parameters

Table 8-52: Parameters

Parameter	Description	Valid value	Default value
inputTableName	Required. The name of the input table.	Table name	N/A
selectedColNames	Optional. The names of columns selected from the input table for evaluation. The column names must be separated with commas (.). The column names must be the same as the feature names saved in the model.	Column name	All columns are selected by default.



Parameter	Description	Valid value	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for evaluation, in the <code>name1=value1/name2=value2</code> format. Separate multiple partitions with commas (.).	N/A	All partitions are selected by default.
<b>enableSparse</b>	Optional. This parameter specifies whether data in the input table is in sparse format.	true and false	false
<b>itemDelimiter</b>	Optional. The delimiter used to separate key-value pairs when data in the input table is in sparse format.	N/A	The default delimiter is a space .
<b>kvDelimiter</b>	Optional. The delimiter used to separate keys and values when data in the input table is in sparse format.	N/A	The default delimiter is a colon (:).
<b>modelName</b>	Required. The name of the input clustering model.	Model name	N/A
<b>outputTableName</b>	Required. The name of the output table.	Table name	N/A
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.

## Evaluation formula

The Calinski-Harabasz metric is also known as the variance ratio criterion (VRC), which is defined as follows:

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)},$$

- $SS_B$  represents the inter-clustering variance. The definition is as follows:

$$SS_B = \sum_{i=1}^k n_i \|m_i - m\|^2,$$

- $k$  represents the number of cluster centers.
- $m_i$  represents the center of cluster  $i$ .
- $m$  represents the mean of the input data.

- $SS_W$  represents the intra-clustering variance. The definition is as follows:

$$SS_W = \sum_{i=1}^k \sum_{x \in c_i} \|x - m_i\|^2,$$

- $k$  represents the number of cluster centers.
- $x$  represents a data point.
- $c_i$  represents the number  $i$  cluster.
- $m_i$  represents the center of cluster  $i$ .
- $N$  represents the total number of records.  $k$  represents the number of cluster centers.

## Examples

- Test data

```
create table if not exists pai_cluster_evaluation_test_input
as select * from (select 1 as id,
1 as f0,2 as f3 from dual union all
select 2 as id, 1 as f0,3 as f3 from dual union all
select 3 as id, 1 as f0,4 as f3 from dual union all
select 4 as id, 0 as f0,3 as f3 from dual union all
select 5 as id, 0 as f0,4 as f3 from dual)tmp;
```

- Clustering model building

```
pai -name kmeans
-project algo_public
-DinputTableName=pai_cluster_evaluation_test_input
```

```
-DselectedColNames=f0,f3
-DcenterCount=3
-Dloop=10
-Daccuracy=0.00001
-DdistanceType=euclidean
-DinitCenterMethod=random
-Dseed=1
-DmodelName=pai_kmeans_test_model
-DidxTableName=pai_kmeans_test_idx
```

• **PAI command**

```
PAI -name cluster_evaluation
-project algo_public
-DinputTableName=pai_cluster_evaluation_test_input
-DselectedColNames=f0,f3
-DmodelName=pai_kmeans_test_model
-DoutputTableName=pai_ft_cluster_evaluation_out;
```

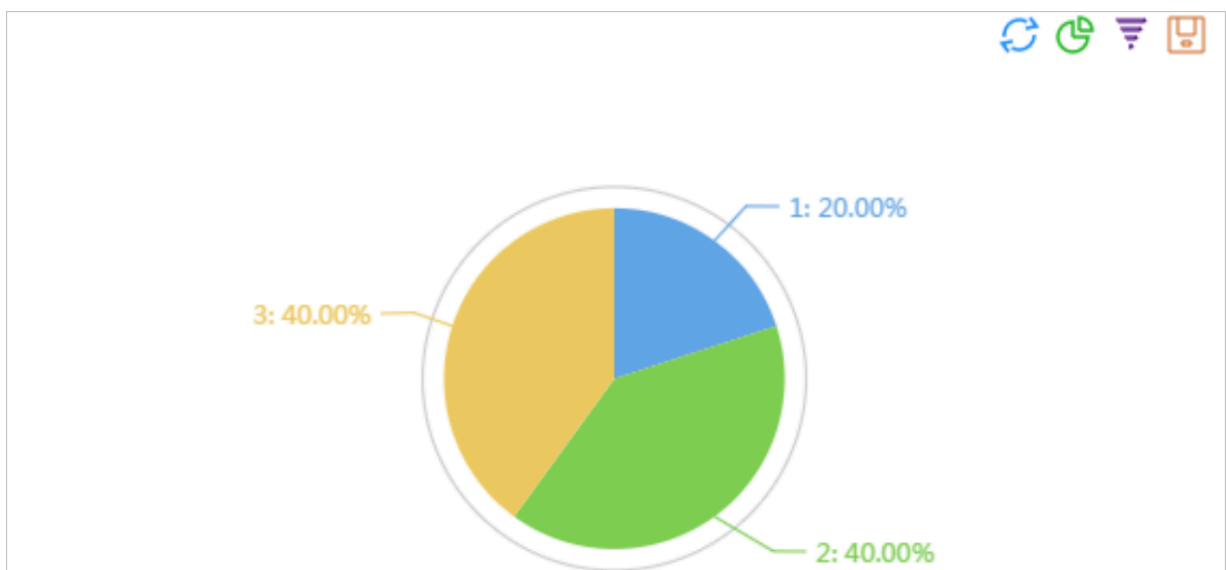
• **Output description**

Table 8-53: Output table (outputTableName)

Column	Description
<b>count</b>	<b>The total number of records.</b>
<b>centerCount</b>	<b>The number of cluster centers.</b>
<b>calinhara</b>	<b>The Calinski Harabasz metric.</b>
<b>clusterCounts</b>	<b>The number of points included in each cluster.</b>

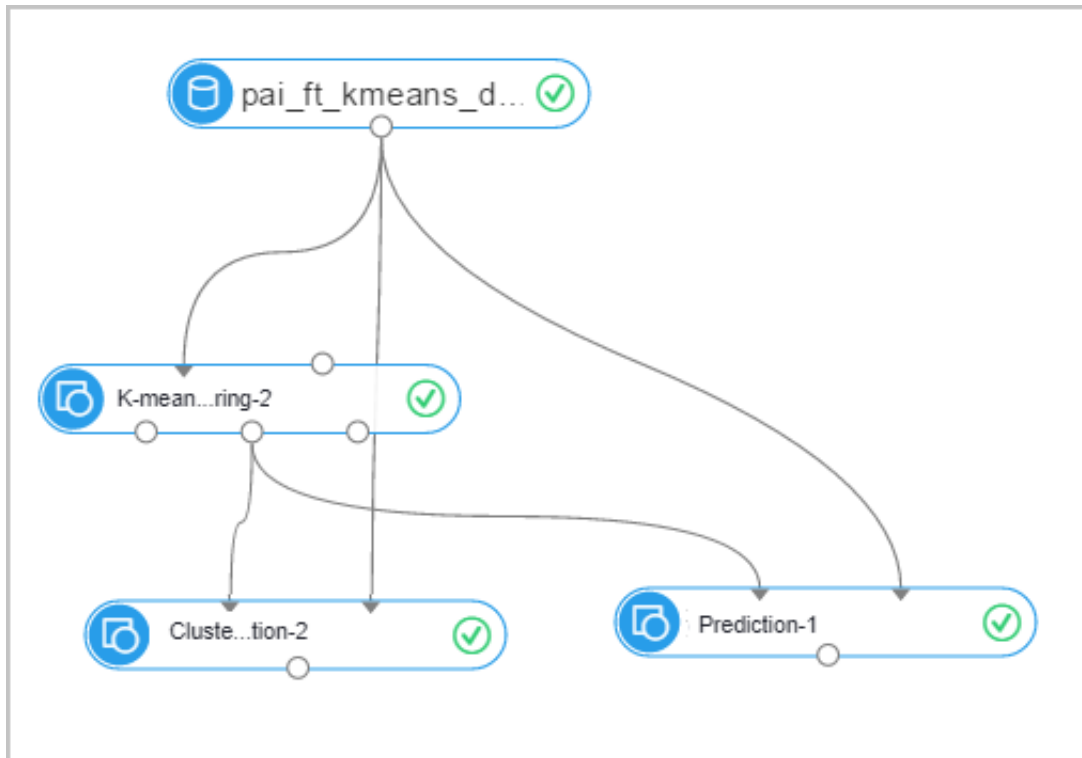
PaiWeb demonstration

Figure 8-3: Clustering model evaluation



## PaiWeb-Pipeline

Figure 8-4: PaiWeb-Pipeline



### 8.3.6.6.3 Binary classification evaluation

You can evaluate a regression algorithm model based on its predicted and actual results. The metrics include MSE, MAE, and MAPE.

PAI command

```

pai -name evaluation
-DinputTableName=input_table
-DlabelColName=label_name
-DpredictionColName=prediction_score
-DoutputTableName=output_table;

```

Algorithm parameters

Table 8-54: Parameters

Parameter	Description	Valid value	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	N/A	N/A

Parameter	Description	Valid value	Default value
<b>inputTablePartition</b>	Optional. The partitions selected from the input table for calculation.	N/A	All partitions of the input table are selected by default.
<b>labelColName</b>	Required. The name of the original label column in the input table. It must be a numerical value.	N/A	N/A
<b>predictionColName</b>	Required. The name of the label column in the prediction result table. It must be a numerical value.	N/A	N/A
<b>outputTableName</b>	Required. The name of the output table.	N/A	N/A
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.

## Output table

Table 8-55: Output column description

Column	Description
<b>MSE</b>	The mean square error, which is used to measure the mean error and evaluate data changes. The smaller the MSE, the more accurately a prediction model describes the test data.
<b>MAE</b>	The mean absolute error, which is used to measure the mean difference between the predicted value and the actual value.
<b>MAPE</b>	The mean absolute percentage error, which is used to measure prediction accuracy. The value is expressed in percentage. If MAPE is set to 15, the mean absolute percentage error is 15%.

### 8.3.6.6.4 Confusion matrix

The Confusion Matrix component is a visualization tool typically used in supervised learning. This tool is used to calculate the classification accuracy of a confusion matrix model by comparing its results with measured values.

#### Procedure

1. Configure the confusion matrix parameters.

The default settings are typically used. You can also select a target column and a prediction probability column. A prediction probability column is the target column generated by the Prediction component.

2. Connect the Confusion Matrix component and the Prediction component.



**Note:**

The parent node of the Confusion Matrix component must be a Prediction component. You can perform confusion matrix analysis only when a classification model is used.

3. Right-click the Confusion Matrix component and choose View Evaluation Report.

#### PAI command

```
PAI -name confusionmatrix
-project algo_public
-DoutputTableName="pai_temp_2954_24178_1"
-DlabelColName="age"
-DpredictionColName="prediction_result"
-DinputTableName="pai_temp_2954_24176_1";
```

Table 8-56: Parameters

Parameter	Description
name	The name of the component.
project	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is algo_public. If you change the name, the system reports an error.
outputTableName	The name of the output table.
labelColName	The name of the label column selected from the input table.
predictionColName	The name of the prediction result column.
inputTableName	The name of the input table for predicting results.

### 8.3.6.6.5 Multiclass classification evaluation

You can evaluate a multiclass classification model based on its predicted and actual results. The indicators include accuracy, kappa, and F1-Score.

#### Component description

**The Multiclass Classification Evaluation component must be connected to a Prediction component and does not support regression models.**

#### PAI command

```
PAI -name MultiClassEvaluation -project algo_public
-DinputTableName="test_input"
-DoutputTableName="test_output"
-DlabelColName="label"
-DpredictionColName="prediction_result"
-Dlifecycle=30;
```

#### Algorithm parameters

Table 8-57: Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	-	-
inputTablePartitions	Optional. The partitions selected from the input table for training.	-	All partitions of the input table are selected by default.
labelColName	Required. The name of the original label column in the input table. It must be a numerical value.	-	-
predictionColName	Required. The name of the label column in the prediction result table. It must be a numerical value.	-	-
outputTableName	Required. The name of the output table.	-	-
predictionColName	Optional. The name of the probability column that lists prediction results. It must be in the {"A":0.2,"B":0.3} format.	-	-

Parameter	Description	Valid values	Default value
lifecycle	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.

### 8.3.6.7 Prediction

The Prediction component is used to make model-based predictions. The component has two inputs (training model and prediction data) and one output (prediction result). Conventional data mining algorithms often use this component for prediction.

#### Procedure

1. Connect all components.
2. Configure column settings.

Table 8-58: Parameters

Parameter	Description
Feature Columns	The feature columns used for prediction. All feature columns are selected by default.
Reserved Columns	The columns reserved and exported to the prediction result .
Output Result Column	The default value is used.
Output Score Column	The default value is used.
Output Detail Column	The default value is used.



#### Note:

Feature columns must be selected if data is in sparse format such as `k1:v1,k2:v2`.



3. After you configure the preceding parameters, right-click the Prediction component and choose View Data from the shortcut menu.

The following three columns are appended to the prediction data:

- **predict\_result**: the prediction result column.
- **predict\_score**: the probability score in prediction results. This column is only appended onto the outputs of binary classification models.
- **predict\_detail**: the prediction result of each category. This column is only appended onto the outputs of binary classification models.

PAI command

```
PAI -name Prediction
-project algo_public
-DdetailColName="prediction_detail"
-DsplitCharacteristic="2"
-DappendColNames="age,campaign,pdays,previous,poutcome,emp_var_rate,
cons_price_idx,cons_conf_idx,euribor3m,nr_employed,y"
-DmodelName="xlab_m_random_forests_6036"
-DresultColName="prediction_result"
-DoutputTableName="pai_temp_675_6048_1"
-DscoreColName="prediction_score"
-DinputTableName="bank_data";
```

Table 8-59: Parameters

Parameter	Description
<b>name</b>	The name of the component.
<b>project</b>	The name of the project. This parameter is used to specify the workspace of the algorithm. The default value is <b>algo_public</b> . If you change the name, the system reports an error.
<b>detailColName</b>	Optional. The name of the detail column in the output table. The default value is <b>prediction_detail</b> .
<b>splitCharacteristic</b>	Optional. The type of classification. The value 1 indicates binary classification. The value 2 indicates multiclass classification.
<b>appendColNames</b>	Optional. The names of columns in the input prediction table to be appended to the output table.
<b>modelName</b>	The name of the random forest model.
<b>resultColName</b>	Optional. The name of the result column in the output table. The default value is <b>prediction_result</b> .

Parameter	Description
outputTableName	The name of the output prediction table.
scoreColName	Optional. The name of the score column in the output table. The default value is prediction_score.
inputTableName	The name of the input prediction table.

## 8.3.7 Deep learning (must be activated separately)

### 8.3.7.1 Activate deep learning

The deep learning service is not a basic function of Apsara Stack Machine Learning Platform for AI. You must purchase it separately.

If you have already deployed the deep learning service, activate it by using the following procedure:

1. Log on to the Apsara Stack Machine Learning Platform for AI console.
2. Click Settings in the left-side navigation pane.
3. Click General. Under Deep Learning, select Enable GPU Compute.

### 8.3.7.2 Read OSS buckets

When using the Read OSS Bucket component on Machine Learning Platform for AI, you must assign the default system role AliyunODPSPAIDefaultRole to your DTplus service account. OSS buckets can be correctly read and written by algorithms of the machine learning platform only when the role is correctly assigned.



#### Note:

The machine learning platform shares service accounts with MaxCompute, because it runs on the MaxCompute framework. During authorization, you must assign the default role to your MaxCompute service account.

You can use RAM authorization to grant OSS access permissions to Machine Learning Platform for AI. Click Settings to grant permission to read and write OSS data. For more information, see [RAM authorization](#).

#### RAM authorization

1. Log on to the Machine Learning Platform For AI console, click Settings in the left-side navigation pane, and select General.

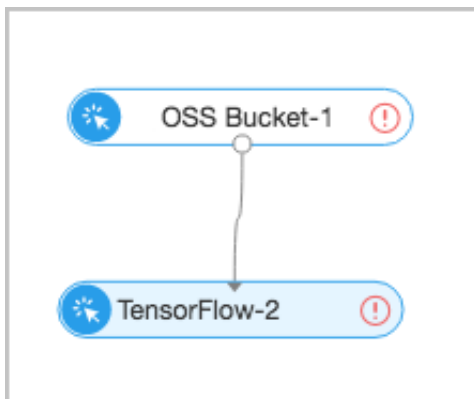
2. Under OSS Authorization, select Authorize Machine Learning Platform for AI to access my OSS resources.
3. The following page is displayed. Click [Click here to authorize access in RAM](#). The RAM page is displayed.
4. Click I Agree.

**Note:**

To view details about the AliyunODPSPAIDefaultRole policy, log on to the [RAM console](#). The default role AliyunODPSPAIDefaultRole contains the following permission information.

Permission (Action)	Description
oss:PutObject	Upload a file or folder object.
oss:GetObject	Obtain a file or folder object.
oss:ListObjects	Query file information.
oss:DeleteObjects	Delete an object.

5. Go back to the machine learning page and click Refresh. RAM information is automatically recorded to the components.
6. Use the deep learning framework. Connect the Read OSS Bucket component to the corresponding deep learning component to obtain permissions to read and write OSS data.



### 8.3.7.3 TensorFlow 1.4

TensorFlow (TF) is an open-source machine learning framework. It is easy to use for algorithm developers. The TF framework is integrated into Apsara Stack Machine Learning Platform for AI. You can write code and adjust computing

resources flexibly by using the TF compute engine. The TF computing engine is a Graphics Processing Unit (GPU) cluster.

#### Parameters

- **Parameter settings**

Table 8-60: Parameter settings

Parameter	Description
Python Code Files	The program execution files. Multiple files can be packaged and uploaded in the tar.gz format.
Primary Python File	Optional. The primary file in a compressed code file package.
Data Source Directory	The path of data sources. You can select Object Storage Service (OSS) data sources.
Configuration File Hyperparameters and Custom Parameters	Machine Learning Platform for AI Tensorflow allows you to use commands to pass in hyperparameter settings and try different learning rates and batch sizes during model testing.
Output Directory	The path of the output model.

- **Tuning**

You can specify the number of GPUs based on the complexity of jobs.

#### PAI command



#### Note:

You do not need to set all parameters. For the definitions of these parameters, see [Table 8-61: Parameters](#). We recommend that you do not directly copy the following command.

```
PAI -name tensorflow_ext140
-Dbuckets="oss://imagenet.oss-cn-shanghai-internal.aliyuncs.com/
smoke_tensorflow/mnist/"
-DgpuRequired="100"
-Darn="acs:ram::166408185518****:role/aliyunodpspaidefaultrole"
-Dscript="oss://imagenet.oss-cn-shanghai-internal.aliyuncs.com/
smoke_tensorflow/mnist_ext.py";
```

The following table lists the descriptions of the parameters.

Table 8-61: Parameters

Parameter	Description	Valid values	Default value
<b>script</b>	<b>Required.</b> The TF algorithm file. This file can be a single file or compressed as a tar.gz format package.	oss://imagenet. oss-cn-shanghai-internal.aliyuncs.com/smoke_tens orflow/mnist_ext. py	N/A
<b>entryFile</b>	<b>Optional.</b> The name of the primary Python file. If the script is a compressed package in the tar.gz format, this parameter is required.	train.py	Null
<b>buckets</b>	<b>Required.</b> The input OSS buckets . You can specify multiple buckets separated with commas (.). Each bucket must end with a forward slash (/).	oss://imagenet. oss-cn-shanghai-internal.aliyuncs.com/smoke_tens orflow/mnist/	Null
<b>arn</b>	<b>Required.</b> The Alibaba Cloud Resource Name ( ARN) of an OSS object.	N/A	Null
<b>gpuRequired</b>	<b>Required.</b> This parameter indicates the number of GPUs to be used.	200	100
<b>checkpointDir</b>	<b>Optional.</b> The TF checkpoint directory.	oss://imagenet. oss-cn-shanghai-internal.aliyuncs.com/smoke_tens orflow/mnist/	Null

Parameter	Description	Valid values	Default value
<b>cluster</b>	<b>Optional.</b>	A JSON format value. Quotation marks must be escaped.	Null
<b>hyperParameters</b>	<b>Optional.</b> The path of the command line hyperparameters.	oss://imagenet.oss-cn-shanghai-internal.aliyuncs.com/smoke_tensorflow/mnist/hyper_parameters.txt	Null

- **script** and **entryFile** are used to specify the TF algorithm script to be executed. If the algorithm is complex and divided into multiple files, you can package the files into a tar.gz file and use **entryFile** to specify the primary Python file.
- **checkpointDir** is used to specify the OSS path to be written by algorithms. You must specify the OSS path when you save TensorFlow models.
- **buckets** is used to specify the OSS path to be read by algorithms. To use OSS, you must specify **arn**.
- **Distributed Machine Learning Platform for AI TensorFlow** supports **cluster**. You can use **cluster** to specify the number of parameter servers and workers. **cluster** is in JSON format, and the quotation marks must be escaped. The JSON code must contain two keys: **ps** and **worker**. Both the **ps** and **worker** parameters contain **count**, **gpu**, **cpu**, and **memory**.

Keyword	Description	Default value	Remarks
<b>count</b>	<b>Required.</b> The number of parameter servers or workers.	-	None

Keyword	Description	Default value	Remarks
<b>gpu</b>	Optional. The number of GPUs allocated to each parameter server or worker. 100 represents a single GPU card.	For parameter servers, the default value is 0 . For workers, the default value is 100.	If the number of GPUs allocated to each worker is set to 0, Machine Learning Platform for AI will reset the value to 100 to ensure the task is scheduled properly.
<b>cpu</b>	Optional. The number of CPUs allocated to each parameter server or worker. 100 represents a single CPU card.	600	None
<b>memory</b>	The memory size allocated to each parameter server or worker. 100 represents 100 MB .	30000	None

#### Examples

The MNIST digit classification set is a set of handwritten digits 1 through 9 that contains training and test sets for machine learning models.

1. Upload the Python execution files and training datasets to OSS. In this case, create a bucket on OSS in China (Shanghai) and name the bucket as tfmnist001. Upload the Python script and training data.
2. Drag and drop the Read OSS Bucket and TensorFlow components onto the canvas to create the following experiment. Set the region for the OSS bucket and configure RAM authorization.
3. Set the TensorFlow parameters. Set the paths for Python Code Files, Primary Python File, and Data Source Directory.
4. Click Run and wait for the experiment to complete running.
5. Right-click the TensorFlow component and view the running log.

## 8.3.8 Time series

### 8.3.8.1 x13\_arima

Autoregressive Integrated Moving Average Model (ARIMA) is a well-known time series prediction method defined by Box and Jenkins in the early 1970s. This model is also called the Box-Jenkins model or the Box-Jenkins method. x13-arima is an ARIMA algorithm for seasonal adjustment based on the open-source X-13ARIMA-SEATS algorithm.

PAI command

```
pai -name x13_arima
 -project algo_public
 -DinputTableName=pai_ft_x13_arima_input
 -DseqColName=id
 -DvalueColName=number
 -Dorder=3,1,1
 -Dstart=1949.1
 -Dfrequency=12
 -Dseasonal=0,1,1
 -Dperiod=12
 -DpredictStep=12
 -DoutputPredictTableName=pai_ft_x13_arima_out_predict
 -DoutputDetailTableName=pai_ft_x13_arima_out_detail
```

Algorithm parameters

Table 8-62: Parameters

Parameter	Description	Valid value	Default value
inputTableName	Required. The name of the input table.	Table name	N/A



Parameter	Description	Valid value	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the <code>partition_name=value</code> format. To specify multiple partitions, use the following format: <code>name1=value1/ name2=value2.</code> Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>seqColName</b>	Required. The name of the time series column.	Column name	This parameter is only used to sort the column specified by <code>valueColName</code> . The value does not affect the calculated results.
<b>valueColName</b>	Required. The name of the value column.	Column name	N/A
<b>groupColNames</b>	Optional. The name of the stratification column. Separate multiple columns with commas (,), such as <code>col0,col1</code> ; . A time series is created for each stratum.	Column name	N/A

Parameter	Description	Valid value	Default value
<b>order</b>	<b>Required.</b> $p$ , $d$ , and $q$ indicate the autoregressive coefficient, difference, and moving regression coefficient, respectively.	$p$ , $d$ , and $q$ must be non-negative integers in the range of $[0, 36]$ .	N/A
<b>start</b>	<b>Optional.</b> The start date of a time series.	A string in the <code>year</code> or <code>year.seasonal</code> format, such as <code>1986.1</code>  For more information, see the time series format section.	1.1
<b>frequency</b>	<b>Optional.</b> The frequency of a time series. Unit: months/year	A positive integer in the range of $(0, 12)$  For more information, see the time series format section.	12
<b>seasonal</b>	<b>Optional.</b> $sp$ , $sd$ , and $sq$ indicate the seasonal autoregressive coefficient, seasonal difference, and seasonal moving regression coefficient, respectively.	$sp$ , $sd$ , and $sq$ must be non-negative integers in the range of $[0, 36]$ .	<code>seasonal</code> is not set by default.
<b>period</b>	<b>Optional.</b> The seasonal period.	A number in the range of $(0, 100]$	<code>frequency</code>

Parameter	Description	Valid value	Default value
<b>maxiter</b>	Optional. The maximum number of iterations.	A positive integer	1500
<b>tol</b>	Optional. The degree of tolerance.	A double type value	1e-5
<b>predictStep</b>	Optional. The number of prediction items.	A number in the range of (0, 365]	12
<b>confidenceLevel</b>	Optional. The prediction confidence level.	A number in the range of (0, 1)	0.95
<b>outputPredictTableName</b>	Required. The name of the output prediction table.	Table name	N/A
<b>outputDetailTableName</b>	Required. The name of the output detail table.	Table name	N/A
<b>outputTablePartition</b>	Optional. The partition in the output table.	Partition name	The output table is non-partitioned by default.
<b>coreNum</b>	Optional. The number of cores.	A positive integer used with memSizePerCore	Automatically calculated.
<b>memSizePerCore</b>	Optional. The memory size of each core. Unit: MB.	A positive integer in the range of [1024, 65536]	Automatically calculated.
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.

#### Time series format

- **The start and frequency parameters specify the two time dimensions of data (valueColName): TS1 and TS2.**
- **The frequency parameter indicates the data frequency within a period, which equals the frequency of TS2 in each TS1.**

- The **start** parameter must be in the **n1.n2** format. This indicates that the start date is the N2 TS2 in the N1 TS1.

Unit time	TS1	TS2	Frequency	Start date
12 months/ year	Year	Month	12	1949.2 indicates the second month of year 1949.
Four quarters/ year	Year	Quarter	4	1949.2 indicates the second quarter of year 1949.
Seven days/ week	Day	Week	7	1949.2 indicates the second day of the 1949th week.
1	Any time unit	1	1	1949.1 indicates the 1949th (year, day, or hour).

Example: value=[1,2,3,5,6,7,8,9,10,11,12,13,14,15]

- **start=1949.3** and **frequency=12** indicate that the data frequency is monthly, and the prediction start date is 1950.06.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949			1	2	3	4	5	6	7	8	9	10
1950	11	12	13	14	15							

- **start=1949.3** and **frequency=4** indicate that the data frequency is quarterly, and the prediction start date is 1953.02.

Year	Qtr1	Qtr2	Qtr3	Qtr4
1949			1	2
1950	3	4	5	6
1951	7	8	9	10
1952	11	12	13	14
1953	14			

- **start=1949.3 and frequency=7 indicate that the data frequency is daily, and the prediction start date is 1951.04.**

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1949			1	2	3	4	5
1950	6	7	8	9	10	11	12
1951	13	14	15				

- **start=1949.1 and frequency=1 indicate that the prediction start date is 1963.00 regardless of the time unit used.**

Cycle	p1
1949	1
1950	2
1951	3
1951	4
1952	5
1953	6
1954	7
1955	8
1956	9
1957	10
1958	11
1959	12
1960	13
1961	14
1962	15

#### Examples

- **Data used for testing: AirPassengers.** The data set contains the number of passengers for international airlines each month from 1949 to 1960. It can be downloaded from <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/AirPassengers.html> .

```
create table pai_ft_x13_arima_input(id bigint,number bigint);
tunnel upload data/airpassengers.csv pai_ft_x13_arima_input -h true;
```

- **PAI command**

```
pai -name x13_arima
 -project algo_public
 -DinputTableName=pai_ft_x13_arima_input
 -DseqColName=id
 -DvalueColName=number
 -Dorder=3,1,1
 -Dseasonal=0,1,1
 -Dstart=1949.1
 -Dfrequency=12
 -Dperiod=12
 -DpredictStep=12
 -DoutputPredictTableName=pai_ft_x13_arima_out_predict
 -DoutputDetailTableName=pai_ft_x13_arima_out_detail
```

- **Output description**

- **The columns of the output table specified by outputPredictTableName are as follows.**

Column	Description
<b>pdate</b>	<b>The prediction date.</b>
<b>forecast</b>	<b>The prediction result.</b>
<b>lower</b>	<b>The lower threshold of the prediction result when the confidence level is specified (default value: 0.95).</b>

Column	Description
upper	The upper threshold of the prediction result when the confidence level is specified (default value: 0.95).

#### Output data

- The columns of the output table specified by outputDetailTableName are as follows.

Column	Description
key	"model" indicates the model.  "evaluation" indicates the evaluation result.  "parameters" indicates the training parameters.  "log" indicates the training log.
summary	The storage details.

#### Output data

##### ■ Model data (key=model)

##### ■ Evaluation metrics (key=evaluation)

### 8.3.8.2 x13\_auto\_arima

ARIMA is described in [x13\\_arima](#). The x13\_auto\_arima algorithm includes a process of automatic model selection.

The x13\_auto\_arima selection process is as follows:

- Default model estimation

In the case of frequency = 1, the default model is (0,1,1).

In the case of frequency > 1, the default model is (0,1,1)(0,1,1).

- **Identification of differencing orders**

**Skip this step if you have configured `diff` and `SeasonalDiff`.**

**Use `Unit root test (wiki)` to determine the difference `d` and the seasonal difference `D`.**

- **Identification of ARMA model orders**

**Select the optimal model based on `BIC (wiki)`. The `maxOrder` and `maxSeasonalOrder` parameters are used in this step.**

- **Comparison of identified model with default model**

**Use `Ljung-Box Q statistic(wiki)` to compare the models. If both models are unacceptable, use the `(3,d,1)(0,D,1)` model.**

- **Final model checks**

PAI command

```
pai -name x13_auto_arima
 -project algo_public
 -DinputTableName=pai_ft_x13_arima_input
 -DseqColName=id
 -DvalueColName=number
 -Dstart=1949.1
 -Dfrequency=12
 -DpredictStep=12
 -DoutputPredictTableName=pai_ft_x13_arima_out_predict2
 -DoutputDetailTableName=pai_ft_x13_arima_out_detail2
```

Algorithm parameters

Table 8-63: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	Table name	-



Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>seqColName</b>	Required. The name of the time series column.	Column name	This parameter is only used to sort <code>valueColNames</code> . It is not relevant to the calculated output.
<b>valueColName</b>	Required. The name of the value column.	Column name	-
<b>groupColNames</b>	Optional. The name of the stratification column. Separate multiple columns with commas (,), such as <code>col0,col1</code> ; . A time series is created for each group.	Column name	-

Parameter	Description	Valid values	Default value
<b>start</b>	<b>Optional. The start date of a time series.</b>	A string in the format of year . seasonal, such as 1986.1  For more information, see the time series format section.	1.1
<b>frequency</b>	<b>Optional. The frequency of a time series.</b>	A positive integer in the range of (0, 12)  For more information, see the time series format section.	The frequency is 12 months/year by default.
<b>maxOrder</b>	<b>Optional. The maximum values of p and q.</b>	A positive integer in the range of [0, 4 ]	2
<b>maxSeasonalOrder</b>	<b>Optional. The seasonal maximum values of p and q.</b>	A positive integer in the range of [0, 2 ]	1
<b>maxDiff</b>	<b>Optional. The maximum value of differential d.</b>	A positive integer in the range of [0, 2 ]	2
<b>maxSeasonalDiff</b>	<b>Optional. The maximum value of seasonal differential d.</b>	A positive integer in the range of [0, 1 ]	1

Parameter	Description	Valid values	Default value
<b>diff</b>	<b>Optional. The differential d.</b>	A positive integer in the range of [0, 2]  If both <code>diff</code> and <code>maxDiff</code> are set, <code>maxDiff</code> is ignored.  If <code>diff</code> is set, then <code>seasonalDiff</code> must also be set.	Default value: -1. This value indicates that <code>diff</code> is not specified by default.
<b>seasonalDiff</b>	<b>Optional. The seasonal differential d.</b>	A positive integer in the range of [0, 1]  If both <code>seasonalDiff</code> and <code>maxSeasonalDiff</code> are set, <code>maxSeasonalDiff</code> is ignored.	Default value: -1. This value indicates that <code>seasonalDiff</code> is not specified by default.
<b>maxiter</b>	<b>Optional. The maximum number of iterations.</b>	A positive integer	1500
<b>tol</b>	<b>Optional. The degree of tolerance.</b>	A double type value	1e-5
<b>predictStep</b>	<b>Optional. The number of prediction items.</b>	A number in the range of (0, 365]	12
<b>confidenceLevel</b>	<b>Optional. The prediction confidence level.</b>	A number in the range of (0, 1)	0.95
<b>outputPredictTable</b>	<b>Required. The name of the output prediction table.</b>	Table name	-

Parameter	Description	Valid values	Default value
<b>outputDetailTableName</b>	<b>Required.</b> The name of the output detail table.	<b>Table name</b>	-
<b>outputTablePartition</b>	<b>Optional.</b> The partitions in the output table.	<b>Partition name</b>	No partition is specified by default.
<b>coreNum</b>	<b>Optional.</b> The number of cores.	A positive integer used with <code>memSizePerCore</code>	Automatically calculated.
<b>memSizePerCore</b>	<b>Optional.</b> The memory size of each core. Unit: MB.	A positive integer in the range of [1024, 65536]	Automatically calculated.
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	A positive integer	No lifecycle is set by default.

#### Time format

- The **start** and **frequency** parameters specify the two time dimensions of data (`valueColName`): TS1 and TS2.
- The **frequency** parameter indicates the data frequency within a period, which equals the frequency of TS2 in each TS1.
- The **start** parameter is in the format of `n1.n2`. This indicates that the start date is the N2 TS2 in the N1 TS1.

Unit time	ts1	ts2	Frequency	Start date
12	Year	Month	12	1949.2 indicates the second month of year 1949.
4	Year	Quarter	4	1949.2 indicates the second quarter of year 1949.

Unit time	ts1	ts2	Frequency	Start date
7	Day	Week	7	1949.2 indicates the second day of a week in year 1949.
1	Any time unit	1	1	1949.1 indicates the 1949th (year, day, or hour).

For example, value=[1,2,3,5,6,7,8,9,10,11,12,13,14,15].

- start=1949.3 and frequency=12 indicate that the data frequency is monthly, and the prediction start date is 1950.06.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949			1	2	3	4	5	6	7	8	9	10
1950	11	12	13	14	15							

- start=1949.3 and frequency=4 indicate that the data frequency is quarterly, and the prediction start date is 1953.02.

Year	Qtr1	Qtr2	Qtr3	Qtr4
1949			1	2
1950	3	4	5	6
1951	7	8	9	10
1952	11	12	13	14
1953	14			

- start=1949.3 and frequency=7 indicate that the data frequency is daily, and the prediction start date is 1951.04.

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1949			1	2	3	4	5
1950	6	7	8	9	10	11	12
1951	13	14	15				

- `start=1949.1` and `frequency=1` indicate that the end date is 1963.00.

Period	p1
1949	1
1950	2
1951	3
1951	4
1952	5
1953	6
1954	7
1955	8
1956	9
1957	10
1958	11
1959	12
1960	13
1961	14
1962	15

#### Examples

- **Data used for testing: AirPassengers.** This data set contains the number of passengers for international airlines each month from 1949 to 1960. It can be downloaded from <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/AirPassengers.html>.

```
create table pai_ft_x13_arima_input(id bigint,number bigint);
tunnel upload data/airpassengers.csv pai_ft_x13_arima_input -h true;
```

- **PAI command**

```
pai -name x13_auto_arima
 -project algo_public
 -DinputTableName=pai_ft_x13_arima_input
 -DseqColName=id
 -DvalueColName=number
 -Dstart=1949.1
 -Dfrequency=12
 -DmaxOrder=4
 -DmaxSeasonalOrder=2
 -DmaxDiff=2
 -DmaxSeasonalDiff=1
 -DpredictStep=12
```

```
-DoutputPredictTableName=pai_ft_x13_arima_auto_out_predict
-DoutputDetailTableName=pai_ft_x13_arima_auto_out_detail
```

• **Output description:**

- **Output table: outputPredictTableName.** The columns are as follows.

Column name	Description
<b>pdate</b>	<b>The prediction date.</b>
<b>forecast</b>	<b>The prediction result.</b>
<b>lower</b>	<b>The lower threshold of the prediction result when the confidence level is confidenceLevel (default value: 0.95).</b>
<b>upper</b>	<b>The upper threshold of the prediction result when the confidence level is confidenceLevel (default value: 0.95).</b>

**Data:**

	key	summary
1	model	{ "comment": { "ma": "arima estimate", "mr": "regress...
2	evaluation	{ "comment": { "aic": "AIC", "aicc": "AICC (F-correcte...
3	paramters	{ "arima": { "d": 1, "isSeasonal": true, "p": 3, "period":...
4	log	1 Log for X-13ARIMA-SEATS program (Version 1.1...

- **Output table: outputDetailTableName.** The columns are as follows.

Column name	Description
<b>key</b>	<b>"model" indicates the model.</b>  <b>"evaluation" indicates the evaluation result.</b>  <b>"parameters" indicates the training parameters.</b>  <b>"log" indicates the training log.</b>
<b>summary</b>	<b>Storage details.</b>

## 8.3.9 Text analysis

### 8.3.9.1 Word splitting

Based on Alibaba Word Segmenter (AliWS), this component performs word splitting on documents specified by columns. Segmented words are separated with spaces. If you have set the part-of-speech (POS) tagging or semantic tagging parameters, the component outputs the word splitting results, POS tagging results, and semantic tagging results. Forward slashes (/) are used as delimiters for POS tagging. Vertical bars (|) are used as delimiters for semantic tagging. Only Chinese Taobao word segmentation and Internet word segmentation are supported.

Parameter settings

**Word segmentation algorithms: CRF and UNIGRAM.**

Table 8-64: Parameters

Parameter	Description
Recognition Options	Specifies whether to recognize nouns with special meanings during word splitting.
Merge Options	Considers the terms used in certain industries as a whole without splitting.
Tokenizer	Allows you to select the Taobao word segmentation or Internet word segmentation. Taobao word segmentation is recommended.
Pos Tagger	Specifies whether to mark the part of speech for each word. If this parameter is specified, the part of speech for each word is marked in the output.

Examples

**The following input table consists of the `id` column (document IDs) and the `text` column (document content).**

PAI command

```
pai -name split_word
-project algo_public
-DinputTableName=doc_test
-DselectedColNames=content1,content2
-DoutputTableName=doc_test_split_word
-DinputTablePartitions="region=cctv_news"
-DoutputTablePartition="region=news"
-Dtokenizer=TAOBAO_CHN
-DenableDfa=true
```



```
-DenablePersonNameTagger=false
-DenableOrgnizationTagger=false
-DenablePostagger=false
-DenableTelephoneRetrievalUnit=true
-DenableTimeRetrievalUnit=true
-DenableDateRetrievalUnit=true
-DenableNumberLetterRetrievalUnit=true
-DenableChnNumMerge=false
-DenableNumMerge=true
-DenableChnTimeMerge=false
-DenableChnDateMerge=false
-DenableSemanticTagger=true
```

## Algorithm parameters

Table 8-65: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	The name of the input table.	-	-
<b>selectedColNames</b>	The names of the columns selected from the input table for word segmentation.	Separate multiple columns with commas (,).	-
<b>outputTableName</b>	The name of the output table.	-	-
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/ name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions in the input table are selected by default.
<b>outputTablePartition</b>	The partition in the output table.	-	The output table is non-partitioned by default.

Parameter	Description	Valid values	Default value
tokenizer	The type of the classifier.	TAOBAO_CHN and INTERNET_CHN	<b>Default value:</b> TAOBAO_CHN. TAOBAO_CHN represents Taobao word segmentation. INTERNET_CHN represents Internet word segmentation.
enableDfa	Specifies whether to enable simple entity recognition.	true and false	true
enablePersonNameTagger	Specifies whether to enable personal name recognition.	true and false	false
enableOrganizationTagger	Specifies whether to enable organization name recognition.	true and false	false
enablePosTagger	Specifies whether to enable part-of-speech tagging.	true and false	false
enableTelephoneRetrieval	Specifies whether to enable retrieval unit configuration for telephone number recognition.	true and false	true
enableTimeRetrieval	Specifies whether to enable retrieval unit configuration for time ID recognition.	true and false	true
enableDateRetrieval	Specifies whether to enable retrieval unit configuration for date ID recognition.	true and false	true

Parameter	Description	Valid values	Default value
<b>enableNumberLetter</b>	Specifies whether to enable retrieval unit configuration for number and letter recognition.	true and false	true
<b>enableChnNumMerge</b>	Specifies whether to merge Chinese numbers into a retrieval unit.	true and false	false
<b>enableNumMerge</b>	Specifies whether to merge regular numbers into a retrieval unit.	true and false	true
<b>enableChnTimeMerge</b>	Specifies whether to merge Chinese time into a semantic unit.	true and false	false
<b>enableChnDateMerge</b>	Specifies whether to merge Chinese dates into a semantic unit.	true and false	false
<b>enableSemanticTagging</b>	Specifies whether to enable semantic tagging.	true and false	false

### 8.3.9.2 Deprecated word filtering

Deprecated word filtering is a preprocessing method in text analysis. This method is used to filter out the noise in word splitting results, such as of, yes, and ah.

Parameter settings

The left and right input ports are as follows:

- **Input table**, which is a word splitting result table for filtering. Parameter: `inputTableName`
- **Deprecated word table**, which is a one-column table with each row containing a deprecated word. Parameter: `noiseTableName`

PAI command

```
PAI -name FilterNoise
```

```
-project algo_public
-DinputTableName="test_input"
-DnoiseTableName="noise_input"
-DoutputTableName="test_output"
-DselectedColNames="words_seg1,words_seg2"
-Dlifecycle=30
```

## Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for calculation.	-	<b>All partitions in the input table are selected by default.</b>
<b>noiseTableName</b>	<b>Required.</b> The name of the deprecated word table.	<b>A one-column table with each row containing a deprecated word</b>	-
<b>noiseTablePartitions</b>	<b>Optional.</b> The partitions selected from the deprecated word table.	-	<b>All partitions in the table are selected by default.</b>
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>selectedColNames</b>	<b>Required.</b> The name of the column to be filtered. Separate multiple columns with commas (.).	-	-
<b>lifecycle</b>	<b>Optional.</b> The lifecycle of the output table.	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional.</b> The number of cores.	<b>A positive integer</b>	<b>Automatically calculated.</b>

Parameter	Description	Valid values	Default value
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB .</b>	<b>A positive integer in the range of (0, 65536)</b>	<b>Automatically calculated.</b>

### 8.3.9.3 String similarity

String similarity calculation is a basic operation in machine learning that is used in information retrieval, natural language processing, and bioinformatics.

This algorithm supports five methods to calculate similarity: Levenshtein distance, longest common substring, string subsequence kernel, cosine, and simhash\_hamming. It also supports two input methods: string-to-string calculation and top N calculation.

PAI command

```
PAI -name string_similarity
-project algo_public
-DinputTableName="pai_test_string_similarity"
-DoutputTableName="pai_test_string_similarity_output"
-DinputSelectedColName1="col0"
-DinputSelectedColName2="col1";
```

Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required. The name of the input table.</b>	-	-
<b>outputTableName</b>	<b>Required. The name of the output table.</b>	-	-
<b>inputSelectedColName1</b>	<b>Optional. The name of the first column for similarity calculation.</b>	-	<b>By default, the first string type column in the table is selected.</b>
<b>inputSelectedColName2</b>	<b>Optional. The name of the second column for similarity calculation.</b>	-	<b>The second string type column in the table is selected by default.</b>

Parameter	Description	Valid values	Default value
<b>inputAppendColName</b>	Optional. The names of columns appended to the output table.	-	No column is appended by default.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for calculation.	-	The whole table is selected by default.
<b>outputColName</b>	Optional. The name of the similarity column in the output table. The column name can be up to 128 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.	-	output
<b>method</b>	Optional. The similarity calculation method.	levenshtein, levenshtein_sim, lcs, lcs_sim, ssk, cosine, simhash_hamming, simhash_hamming_sim, minhash_sim, and hash_jaccard_sim	levenshtein_sim
<b>lambda</b>	Optional. The weight of the matching string. This parameter takes effect when similarityType is set to ssk.	(0, 1)	0.5

Parameter	Description	Valid values	Default value
<b>k</b>	<b>Optional. The length of the substring. This parameter takes effect when similarityType is set to ssk or cosine.</b>	<b>(0, 100)</b>	<b>2</b>
<b>kVec</b>	<b>Optional. The number of MinHash instances.</b>	<b>A positive integer</b>	<b>2</b>
<b>b</b>	<b>Optional. The number of buckets.</b>	<b>A positive integer</b>	<b>1</b>
<b>seed</b>	<b>Optional. The random seed used in a MinHash instance.</b>	<b>A positive integer</b>	<b>0</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB.</b>	<b>A positive integer in the range of (0, 65536)</b>	<b>Automatically calculated.</b>

#### Examples

- **SQL statement to generate data:**

```
create table pai_ft_string_similarity_input
as select * from
(select 0 as id, "Beijing" as col0,
"Beijing" as col1 from dual union all
select 1 as id,
"Beijing" as col0,
"Beijing Shanghai" as col1 from dual union all
select 2 as id,
"Beijing" as col0,
```

```
"Beijing Shanghai Hongkong" as col1 from dual)tmp;
```

- **PAI command**

```
PAI -name string_similarity
-project sre_mpi_algo_dev
-DinputTableName=pai_ft_string_similarity_input
-DoutputTableName=pai_ft_string_similarity_output
-DinputSelectedColName1=col0
-DinputSelectedColName2=col1
-Dmethod=simhash_hamming
-DinputAppendColNames=col0,col1;
```

- **Output description**

- **Output obtained by using the simhash\_hamming method:**

col0 ▲	col1 ▲	output ▲
beijing	beijing	0
beijing	beijing shanghai	6
beijing	beijing shanghai xianggang	13

- **Output obtained by using the simhash\_hamming\_sim method:**

col0 ▲	col1 ▲	output ▲
beijing	beijing	1
beijing	beijing shanghai	0.90625
beijing	beijing shanghai xianggang	0.796875

### 8.3.9.4 Convert row, column, and value to KV pair

This component converts rows, columns, and values into KV pairs. A row, column, and value set is defined as XXD or XXL, where X can represent any type, D represents Double, and L represents Bigint. The row, column, and value set is converted into KV format (row,[col\_id:value]). The row and value types are consistent with the original input data. The col\_id type is Bigint, and the column is mapped to col\_id based on the index table.

PAI command

```
PAI -name triple_to_kv
-project algo_public
-DinputTableName=test_data
-DoutputTableName=test_kv_out
-DindexOutputTableName=test_index_out
-DidColName=id
-DkeyColName=word
-DvalueColName=count
-DinputTablePartitions=ds=test1
```



```
-DindexInputTableName=test_index_input
-DindexInputKeyColName=word
-DindexInputKeyIdColName=word_id
-DkvDelimiter=:
-DpairDelimiter=;
-Dlifecycle=3
```

## Parameters

Table 8-66: Parameters

Parameter	Description	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	The input table cannot be empty.
<b>idColName</b>	<b>Required.</b> The name of the column to be retained after the table is converted into a KV table.	-
<b>keyColName</b>	<b>Required.</b> The name of the key column in the KV table.	-
<b>valueColName</b>	<b>Required.</b> The name of the value column in the KV table.	-
<b>outputTableName</b>	<b>Required.</b> The name of the output KV table.	-
<b>indexOutputTableName</b>	<b>Required.</b> The name of the index table for the output keys.	-
<b>indexInputTableName</b>	<b>Optional.</b> The name of the input index table.	No index table is set by default. The table cannot be empty and it does not need to contain indexes for all of the output keys.
<b>indexInputKeyColName</b>	<b>Optional.</b> The name of the key column in the input index table.	No key column is specified by default. This parameter is required if <b>indexInputTableName</b> is set.

Parameter	Description	Default value
<b>indexInputKeyIdColName</b>	Optional. The name of the index column in the input index table.	No index column is specified by default. This parameter is required if <b>indexInputTableName</b> is set.
<b>inputTablePartitions</b>	Optional. The partitions in the input table.	No partition is specified by default. Only one partition can be input.
<b>kvDelimiter</b>	Optional. The delimiter used to separate the key and value.	The default delimiter is a colon (:).
<b>pairDelimiter</b>	Optional. The delimiter used to separate KV pairs.	The default delimiter is a semicolon (;).
<b>lifecycle</b>	Optional. The lifecycle of the output table.	No lifecycle is set by default.
<b>coreNum</b>	Optional. The number of cores.	-1
<b>memSizePerCore</b>	Optional. The memory size of each core. Valid values: 100 to 65536.	-1

## Examples

- **SQL statement to generate data:**

```
drop table if exists triple2kv_test_input;
create table triple2kv_test_input as
select * from
(
 select '01' as id, 'a' as word,
 10 as count from dual union all
 select '01' as id, 'b' as word,
 20 as count from dual union all
 select '01' as id, 'c' as word,
 30 as count from dual union all
 select '02' as id,
 'a' as word,
 100 as count from dual union all
 select '02' as id, 'd' as word,
 200 as count from dual union all
 select '02' as id, 'e' as word,
 300 as count from dual) tmp;
```

- **PAI command**

```
PAI -name triple_to_kv
 -project algo_public
 -DinputTableName=triple2kv_test_input
```

```
-DoutputTableName=triple2kv_test_input_out
-DindexOutputTableName=triple2kv_test_input_index_out
-DidColName=id
-DkeyColName=word
-DvalueColName=count
-Dlifecycle=1;
```

Input description

### Input table

Table 8-67: Input description

id	word	count
01	a	10
01	b	20
01	c	30

Output description

- The output KV table is as follows, where custom KV delimiters can be used.

Table 8-68: Output description

id	key_value
01	1:10;2:20;3:30

- The output index table that contains indexes for the words is as follows.

Table 8-69: Output index table

key	key_id
a	1
b	2
c	3

### 8.3.9.5 String similarity - Top N

String similarity calculation is a basic operation in machine learning that is used in information retrieval, natural language processing, and bioinformatics.

This algorithm supports five methods to calculate similarity: Levenshtein distance, longest common substring, string subsequence kernel, cosine, and simhash\_hamming. It also supports two input methods: string-to-string calculation and top N calculation.

## PAI command

```
PAI -name string_similarity_topn
 -project algo_public
 -DinputTableName="pai_test_string_similarity_topn"
 -DoutputTableName="pai_test_string_similarity_topn_output"
 -DmapTableName="pai_test_string_similarity_map_topn"
 -DinputSelectedColName="col0"
 -DmapSelectedColName="col1";
```

## Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>mapTableName</b>	<b>Required.</b> The name of the mapping table.	-	-
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>inputSelectedColName</b>	<b>Optional.</b> The name of the column selected from the left table for similarity calculation.	-	The first string type column in the table is selected by default.
<b>mapSelectedColName</b>	<b>Optional.</b> The name of the column selected from the mapping table for similarity calculation. The similarities between each row in the left table and all strings in the mapping table are calculated, and the top N entries are output.	-	The first string type column in the table is selected by default.

Parameter	Description	Valid values	Default value
<b>inputAppendColNames</b>	<b>Optional.</b> The names of columns appended to the output table from the input table.	-	<b>No column is appended by default.</b>
<b>inputAppendRenameColNames</b>	<b>Optional.</b> The aliases of columns appended to the output table from the input table. This parameter takes effect when <b>inputAppendColNames</b> is specified.	-	<b>No alias is specified by default.</b>
<b>mapAppendColNames</b>	<b>Optional.</b> The names of columns appended to the output table from the mapping table.	-	<b>No column is appended by default.</b>
<b>mapAppendRenameColNames</b>	<b>Optional.</b> The aliases of columns appended to the output table from the mapping table.	-	<b>No alias is specified by default.</b>
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table.	-	<b>The whole table is selected by default.</b>
<b>mapTablePartitions</b>	<b>Optional.</b> The partitions in the mapping table.	-	<b>The whole table is selected by default.</b>

Parameter	Description	Valid values	Default value
<b>outputColName</b>	Optional. The name of the similarity column in the output table. The column name can be up to 128 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.	-	<b>output</b>
<b>method</b>	Optional. The similarity calculation method.	levenshtein_sim, lcs_sim, ssk, cosine, simhash_hamming_sim, minhash_sim, and hash_jaccard_sim	<b>levenshtein_sim</b>
<b>lambda</b>	Optional. The weight of the matching string. This parameter takes effect when similarityType is set to ssk.	(0, 1)	<b>0.5</b>
<b>k</b>	Optional. The length of the substring. This parameter takes effect when similarityType is set to ssk or cosine.	(0, 100)	<b>2</b>
<b>kVec</b>	Optional. The number of MinHash instances.	A positive integer	<b>2</b>
<b>b</b>	Optional. The number of buckets.	A positive integer	<b>1</b>

Parameter	Description	Valid values	Default value
<b>seed</b>	<b>Optional. The random seed used in a MinHash instance.</b>	<b>A positive integer</b>	<b>0</b>
<b>topN</b>	<b>Optional. The number of similarity maximums to be output.</b>	<b>(0, +∞)</b>	<b>10</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB</b>	<b>A positive integer in the range of (0, 65536)</b>	<b>Automatically calculated.</b>

#### Examples

- **SQL statement to generate data:**

```
create table pai_ft_string_similarity_topn_input
as select * from
(select 0 as id,
"Beijing" as col0 from dual union all
select 1 as id,
"Beijing Shanghai" as col0 from dual union all
select 2 as id,
"Beijing Shanghai Hongkong" as col0 from dual)tmp;
```

- **PAI command**

```
PAI -name string_similarity_topn
-project sre_mpi_algo_dev
-DinputTableName=pai_ft_string_similarity_topn_input
-DmapTableName=pai_ft_string_similarity_topn_input
-DoutputTableName=pai_ft_string_similarity_topn_output
-DinputSelectedColName=col0
-DmapSelectedColName=col0
-DinputAppendColNames=col0
-DinputAppendRenameColNames=input_col0
-DmapAppendColNames=col0
-DmapAppendRenameColNames=map_col0
```

```
-Dmethod=simhash_hamming_sim;
```

• **Output.**

input_col0 ▲	map_col0 ▲	output ▲
beijing	beijing	1
beiji beijing	beijing shanghai	0.90625
beijing	beijing shanghai xianggang	0.796875
beijing shanghai	beijing shanghai	1
beijing shanghai	beijing	0.90625
beijing shanghai	beijing shanghai xianggang	0.828125
beijing shanghai xianggang	beijing shanghai xianggang	1
beijing shanghai xianggang	beijing shanghai	0.828125
beijing shanghai xianggang	beijing	0.796875

### 8.3.9.6 N-gram counting

N-gram counting is a step in language model training. N-grams are generated based on words. The number of the corresponding N-grams in all corpora is counted. The N-gram counting model counts the number of N-grams in all documents rather than in a single document. For more information, see [ngram-count](#).

PAI command

```
PAI -name ngram_count
 -project algo_public
 -DinputTableName=pai_ngram_input
 -DoutputTableName=pai_ngram_output
 -DinputSelectedColNames=col0
 -DweightColName=weight
 -DcoreNum=2
 -DmemSizePerCore=1000;
```

Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	N/A
outputTableName	Required. The name of the output table.	Table name	N/A



Parameter	Description	Valid values	Default value
<b>inputSelectedColName</b>	Optional. The names of columns selected from the input table.	Column name	The first character type column is selected by default.
<b>weightColName</b>	Optional. The name of the weight column.	Column name	1
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table.	Partition name	The whole table is selected by default.
<b>countTableName</b>	Optional. The name of the former N-gram counting output table. This table is merged into the output result.	Table name	N/A
<b>countWordColName</b>	Optional. The name of the word column in the counting table.	Column name	The second column is selected by default.
<b>countCountColName</b>	Optional. The name of the counting column in the counting table.	Column name	The third column is selected by default.
<b>countTablePartitions</b>	Optional. The partitions in the counting table.	Partition name	N/A
<b>vocabTableName</b>	Optional. The name of the bag-of-words table. The words that are not contained in the bag-of-words table are marked with \<unk\.	Table name	N/A

Parameter	Description	Valid values	Default value
<b>vocabSelectedColName</b>	Optional. The name of the bag-of-words column.	Column name	The first character type column is selected by default.
<b>vocabTablePartitions</b>	Optional. The partitions in the bag-of-words table.	Partition name	N/A
<b>order</b>	Optional. The maximum length of N-grams.	N/A	3
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	N/A
<b>coreNum</b>	Optional. The number of cores.	A positive integer	N/A
<b>memSizePerCore</b>	Optional. The memory size of each core.	A positive integer	N/A

### 8.3.9.7 Text summarization

Automatic summarization uses computers to automatically extract summaries from a source document. A summary is a simple, concise, and short document that completely and accurately describes the content of a certain document. This TextRank-based algorithm generates summaries by extracting existing sentences in the document.

PAI command

```
PAI -name TextSummarization
 -project algo_public
 -DinputTableName="test_input"
 -DoutputTableName="test_output"
 -DdocIdCol="doc_id"
 -DsentenceCol="sentence"
 -DtopN=2
```

```
-Dlifecycle=30;
```

## Algorithm parameters

Table 8-70: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	Required. The name of the input table.	-	-
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for calculation.	-	All partitions in the input table are selected by default.
<b>outputTableName</b>	Required. The name of the output table.	-	-
<b>docIdCol</b>	Required. The name of the document ID column.	-	-
<b>sentenceCol</b>	Required. The sentence column.	Only one column can be specified.	-
<b>topN</b>	Optional. The top N key sentences to be output.	-	3
<b>similarityType</b>	Optional. The method used to calculate sentence similarity.	lcs_sim, levenshtein_sim, cosine, and ssk	lcs_sim
<b>lambda</b>	Optional. The weight of the matching string. This parameter takes effect when similarityType is set to ssk.	(0, 1)	0.5

Parameter	Description	Valid values	Default value
<b>k</b>	Optional. The length of the substring. This parameter takes effect when <b>similarityType</b> is set to <b>ssk</b> or <b>cosine</b> .	(0, 100)	2
<b>dampingFactor</b>	Optional. The damping factor.	(0, 1)	0.85
<b>maxIter</b>	Optional. The maximum number of iterations.	[1, +]	100
<b>epsilon</b>	Optional. The convergence coefficient.	(0, $\infty$ )	0.000001
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	Optional. The number of cores.	A positive integer	Automatically calculated.
<b>memSizePerCore</b>	Optional. The memory size of each core.	A positive integer	Automatically calculated.

The sentence similarity options are as follows:

- **lcs\_sim**: The formula is  $1.0 - (\text{Length of the longest common subsequence}) / \max(\text{len}(A), \text{len}(B))$ .
- **levenshtein\_sim**: The formula is  $1.0 - (\text{Levenshtein distance}) / \max(\text{len}(A), \text{len}(B))$ .
- **cosine**: See Lodhi, Huma; Saunders, Craig; Shawe-Taylor, John; Cristianini, Nello; Watkins, Chris (2002). "Text classification using string kernels". Journal of Machine Learning Research: 419-444.
- **ssk**: See Leslie, C.; Eskin, E.; Noble, W.S. (2002), The spectrum kernel: A string kernel for SVM protein classification 7, pp. 566-575.



#### Note:

**A** and **B** indicate two strings, and `len(A)` indicates the length of string A.

Output format description

**The output table contains the `doc_id` and `abstract` columns, as shown in [Table 8-71](#):**

*Output table example.*

Table 8-71: Output table example

doc_id	abstract
1000894	In 2008, the Shanghai Stock Exchange published disclosure guidelines for the corporate social responsibility of listed companies. Three types of companies were urged to disclose their CSR reports , and other qualified listed companies were encouraged to voluntarily disclose their CSR reports. In 2012, a total of 379 listed companies making up a 40% of all listed companies disclosed CSR reports. Of those companies, 305 were mandated to disclose CSR reports and and 75 voluntarily disclosed CSR reports . According to Hu Ruyin, Shanghai Stock Exchange will explore how to expand the scope of CSR report disclosure , revise and refine the guidelines on disclosure of the CSR reports, and encourage more organizations to promote CSR product innovation.

### 8.3.9.8 Keyword extraction

Keyword extraction is one of the important technologies in natural language processing. It is used to extract keywords from a document. This algorithm is based on TextRank, a variation of the PageRank algorithm used to describe the relationship between webpages. This algorithm uses the relationship between certain words to construct a network, calculate the importance of each word, and determine words with larger weights as keywords.

PAI command

```
PAI -name KeywordsExtraction
-DinputTableName=maple_test_keywords_basic_input
-DdocIdCol=docid -DdocContent=word
```

```
-DoututTableName=maple_test_keywords_basic_output
-DtopN=19;
```

## Algorithm parameters

Table 8-72: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	Table name	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/ name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions are selected by default.
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>docIdCol</b>	<b>Required.</b> The name of the document ID column.	Only one column can be specified.	-
<b>docContent</b>	<b>Required.</b> The word column.	Only one column can be specified.	-

Parameter	Description	Valid values	Default value
<b>topN</b>	<b>Optional. The number of top N keywords to be output. If this number is smaller than the number of keywords, all keywords are output.</b>	-	<b>5</b>
<b>windowSize</b>	<b>Optional. The window size of the TextRank algorithm.</b>	-	<b>2</b>
<b>dumpingFactor</b>	<b>Optional. The damping factor of the TextRank algorithm.</b>	-	<b>0.85</b>
<b>maxIter</b>	<b>Optional. The maximum number of iterations of the TextRank algorithm.</b>	-	<b>100</b>
<b>epsilon</b>	<b>Optional. The convergence residual threshold of the TextRank algorithm.</b>	-	<b>0.000001</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>This parameter is used with memSizePerCore. The value must be a positive integer in the range of [1, 9999].</b>	<b>Automatically calculated.</b>

Parameter	Description	Valid values	Default value
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB .</b>	<b>A positive integer in the range of [ 1024, 65536]</b>	<b>Automatically calculated.</b>

#### Examples

The words in the input table are separated with spaces, and deprecated words and all punctuations are filtered out.



Table 8-73: Examples

docid: string	word: string
doc0	<p>The blended-wing-body aircraft is a new direction for the future development in the aviation field Many research institutions inside and outside China have carried out research on the blended-wing-body aircraft while its fully automated shape optimization algorithm has become a new hot topic Based on the existing research achievements inside and outside China common modeling and flow solver tools have been analyzed and compared The geometric modeling grid flow field solver and shape optimization modules have been designed The pros and cons between different algorithms have been compared to achieve the optimized shape of the blended-wing-body aircraft in the conceptual design stage Geometric modeling and grid generation module are achieved based on the transfinite interpolation algorithm and spline based grid generation method The flow solver module includes the finite difference solver the finite element solver and the panel method solver The finite difference solver includes mathematical modeling of the potential flow the derivation of the Cartesian grid based variable step length difference scheme Cartesian grid generation and indexing algorithm the Cartesian grid based Neumann boundary conditions expression form derivation are achieved based on finite element difference solver The aerodynamic parameters of a two-dimensional airfoil are calculated based on the finite difference solver The finite element solver includes potential flow modeling based on the variational principle of the finite element theory the derivation of the two-dimensional finite element Kutta conditional least squares based speed solving algorithm</p>

## PAI command

```
PAI -name KeywordsExtraction
-DinputTableName=maple_test_keywords_basic_input
-DdocIdCol=docid -DdocContent=word
-DoutputTableName=maple_test_keywords_basic_output
-DtopN=19;
```

## Input/output description

Table 8-74: Output table description

docid	keywords	weight
doc0	Based on	0.041306752223538405
doc0	Algorithm	0.03089845626854151
doc0	Modeling	0.021782865850562882
doc0	Grid	0.020669749212693957
doc0	Solver	0.020245609506360847
doc0	Aircraft	0.019850761705313365
doc0	Research	0.014193732541852615
doc0	Finite element	0.013831122054200538
doc0	Solving	0.012924593244133104
doc0	Module	0.01280216562287212
doc0	Derivation	0.011907588923852495
doc0	Shape	0.011505456605632607
doc0	Difference	0.011477831662367547
doc0	Flow	0.010969269350293957
doc0	Design	0.010830986516637251
doc0	Implementation	0.010747536556701583
doc0	Two-dimensional	0.010695570768457084
doc0	Development	0.010527342662670088
doc0	New	0.010096978306668461

### 8.3.9.9 Sentence splitting

You can split sentences in a document by punctuation. This component is used to preprocess text summarizations. It splits text such that each row contains only a single sentence.

PAI command

```
PAI -name SplitSentences
 -project algo_public
 -DinputTableName="test_input"
 -DoutputTableName="test_output"
 -DdocIdCol="doc_id"
 -DdocContent="content"
 -Dlifecycle=30
```

Parameters

Table 8-75: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for calculation.	-	<b>All partitions in the input table are selected by default.</b>
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>docIdCol</b>	<b>Required.</b> The name of the document ID column.	-	-
<b>docContent</b>	<b>Required.</b> The name of the document content column.	<b>Only one column can be specified.</b>	-

Parameter	Description	Valid values	Default value
<b>delimiter</b>	<b>Optional. A set of characters used to determine the end of a sentence.</b>	-	<b>The default delimiter set contains the period (.), question mark (!), and exclamation mark (?).</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>

Output format description

**The output table contains the doc\_id and sentence columns, as shown in [Table 8-76](#):**

*Output table example.*

Table 8-76: Output table example

doc_id	sentence
1000894	In 2008, the Shanghai Stock Exchange published disclosure guidelines for the corporate social responsibility of listed companies. Three types of companies were urged to disclose their CSR reports , and other qualified listed companies were encouraged to voluntarily disclose their CSR reports.
1000894	In 2012, a total of 379 listed companies making up a 40% of all listed companies disclosed CSR reports. Of those companies, 305 were mandated to disclose CSR reports and and 75 voluntarily disclosed CSR reports.

### 8.3.9.10 Semantic vector distance

You can calculate the extension words or sentences of the specified words or sentences based on the calculated semantic vectors, such as word vectors calculated by the Word2Vec component. The extension words or sentences are a set of vectors closest to a certain vector. The following example shows how to generate a list of words that are most similar to the word that you entered based on the word vectors calculated by the Word2Vec component.

PAI command

```
PAI -name SemanticVectorDistance
 -project algo_public
 -DinputTableName="test_input"
 -DoutputTableName="test_output"
 -DdidColName="word"
 -DvectorColNames="f0,f1,f2,f3,f4,f5"
 -Dlifecycle=30
```

Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for calculation.	-	All partitions in the input table are selected by default.
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>idTableName</b>	<b>Optional.</b> The name of the vector ID table for vector calculation. The table contains only one column and each row stores a vector ID.	-	No vector ID table is specified by default. This means that all vectors in the input table are calculated.

Parameter	Description	Valid values	Default value
<b>idTablePartitions</b>	Optional. The partitions selected from the ID table for calculation.	-	All partitions are selected by default.
<b>idColName</b>	Required. The name of the ID column.	-	3
<b>vectorColNames</b>	Optional. A list of vector column names, such as f1, f2,...	-	-
<b>topN</b>	Optional. The number of the closest vectors to output.	[1, +∞]	5
<b>distanceType</b>	Optional. The distance calculation method.	euclidean, cosine, and manhattan	euclidean
<b>distanceThreshold</b>	Optional. The distance threshold. Only the distances between two vectors that do not exceed this threshold are output.	(0, +∞)	∞
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	Optional. The number of cores.	A positive integer	Automatically calculated.
<b>memSizePerCore</b>	Optional. The memory size of each core.	A positive integer	Automatically calculated.

#### Examples

The output table contains the original\_id, near\_id, distance, and rank columns.

original_id	near_id	distance	rank
hello	hi	0.2	1
hello	xxx	xx	2
Man	Woman	0.3	1
Man	xx	xx	2
..	...	...	...

### 8.3.9.11 Document similarity

This algorithm calculates the similarity between two text documents by comparing the similarities of documents or sentences separated by spaces. This algorithms functions similar to how the similarity of strings is calculated.

PAI command

```
PAI -name doc_similarity
 -project algo_public
 -DinputTableName="pai_test_doc_similarity"
 -DoutputTableName="pai_test_doc_similarity_output"
 -DinputSelectedColName1="col0"
 -DinputSelectedColName2="col1"
```

Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	-	-
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	-	-
<b>inputSelectedColName1</b>	<b>Optional.</b> The name of the first column for similarity calculation.	-	By default, the first string type column in the table is selected.
<b>inputSelectedColName2</b>	<b>Optional.</b> The name of the second column for similarity calculation.	-	The name of the second string type column in the table is selected by default.

Parameter	Description	Valid values	Default value
<b>inputAppendColName</b>	Optional. The names of columns appended to the output table.	-	No column is appended by default.
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table.	-	The whole table is selected by default.
<b>outputColName</b>	Optional. The name of the similarity column in the output table. The column name can be up to 128 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.	-	output
<b>method</b>	Optional. The similarity calculation method.	levenshtein, levenshtein_sim, lcs, lcs_sim, ssk, cosine, simhash_hamming, and simhash_hamming_sim	levenshtein_sim
<b>lambda</b>	Optional. The weight of the matching word pair. This parameter takes effect if similarity Type is set to ssk.	(0, 1)	0.5



Parameter	Description	Valid values	Default value
<b>k</b>	<b>Optional. The length of the substring. This parameter takes effect if similarity Type is set to ssk or cosine.</b>	<b>(0, 100)</b>	<b>2</b>
<b>kVec</b>	<b>Optional. The number of MinHash instances.</b>	<b>A positive integer</b>	<b>2</b>
<b>b</b>	<b>Optional. The number of buckets.</b>	<b>A positive integer</b>	<b>1</b>
<b>seed</b>	<b>Optional. The random seed used in a MinHash instance.</b>	<b>A positive integer</b>	<b>0</b>
<b>lifecycle</b>	<b>Optional. The lifecycle of the output table.</b>	<b>A positive integer</b>	<b>No lifecycle is set by default.</b>
<b>coreNum</b>	<b>Optional. The number of cores.</b>	<b>A positive integer</b>	<b>Automatically calculated.</b>
<b>memSizePerCore</b>	<b>Optional. The memory size of each core. Unit: MB.</b>	<b>A positive integer in the range of (0, 65536)</b>	<b>Automatically calculated.</b>

#### Examples

- **SQL statement to generate data:**

```
drop table if exists pai_doc_similarity_input;
create table pai_doc_similarity_input as
select * from (
select 0 as id,
"Beijing and Shanghai" as col0,
"Beijing and Shanghai" as col1 from dual union all
select 1 as id,
"Beijing and Shanghai" as col0,
"Beijing, Shanghai, and Hong Kong" as col1 from dual)tmp;
```

- **PAI command**

```
drop table if exists pai_doc_similarity_output;
```

```
PAI -name doc_similarity
-project algo_public
-DinputTableName=pai_doc_similarity_input
-DoutputTableName=pai_doc_similarity_output
-DinputSelectedColName1=col0
-DinputSelectedColName2=col1
-Dmethod=levenshtein_sim
-DinputAppendColNames=id,col0,col1;
```

- **Input description: pai\_doc\_similarity\_input**

ID	col0	col1
1	Beijing and Shanghai	Beijing, Shanghai, and Hong Kong
0	Beijing and Shanghai	Beijing and Shanghai

- **Output description: pai\_doc\_similarity\_output**

ID	col0	col1	Output
1	Beijing and Shanghai	Beijing, Shanghai, and Hong Kong	0.6666666666666667
0	Beijing and Shanghai	Beijing and Shanghai	1.0

### 8.3.9.12 PMI

**Mutual information (MI)** is a measure of information in the information theory.

It can be regarded as the amount of information contained in a random variable about another variable, or the reduction in uncertainty of a random variable due to the known random variable.

This algorithm is used to count the co-occurrence of all words in several documents and calculate the point mutual information (PMI) . PMI definition:  $PMI(x,y) = \ln(p(x,y) / (p(x)p(y))) = \ln(\#(x,y)D / (\#x\#y))$ .

$\#(x,y)$  indicates the number of pair(x,y).

- **D** indicates the total number of pairs.
- If x and y appear in the same window, the output is  $\#x+=1; \#y+=1; \#(x,y)+=1$ .

PAI command

```
PAI -name PointwiseMutualInformation
-project algo_public
-DinputTableName=maple_test_pmi_basic_input
-DdocColName=doc
-DoutputTableName=maple_test_pmi_basic_output
-DminCount=0
-DwindowSize=2
```

```
-DcoreNum=1
-DmemSizePerCore=110;
```

## Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	Required. The name of the input table.	Table name	-
<b>outputTableName</b>	Required. The name of the output table.	Table name	-
<b>docColName</b>	Required. The name of the document column after word splitting, where words are separated with spaces.	Column name	-
<b>windowSize</b>	Optional. The window size. For example, the value 5 refers to the five words adjacent on the right of the current word. Words that appear in the window are considered related to the current word.	[1, sentence length]	The whole row is selected by default.
<b>minCount</b>	The minimum word truncation frequency. Words that appear for a number of times less than this value are filtered out.	[0, 2e63]	5

Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	Partition name	All partitions are selected by default.
<b>lifecycle</b>	Optional. The lifecycle of the output table.	A positive integer	No lifecycle is set by default.
<b>coreNum</b>	Optional. The number of cores.	This parameter is used with <code>memSizePerCore</code> . The value must be a positive integer in the range of [1, 9999].	Automatically calculated.
<b>memSizePerCore</b>	The memory size of each core. Unit: MB.	A positive integer in the range of [1024, 65536]	Automatically calculated.

## Examples

- Data generation

<b>doc:string</b>
w1 w2 w3 w4 w5 w6 w7 w8 w8 w9
w1 w3 w5 w6 w9
w0
w0 w0
w9 w1 w9 w1 w9

• **PAI command**

```
PAI -name PointwiseMutualInformation
 -project algo_public
 -DinputTableName=maple_test_pmi_basic_input
 -DdocColName=doc
 -DoutputTableName=maple_test_pmi_basic_output
 -DminCount=0
 -DwindowSize=2
 -DcoreNum=1
 -DmemSizePerCore=110;
```

• **Output description**

Table 8-77: Output table

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w0	w0	2	2	1	2. 0794415416 798357
w1	w1	10	10	1	-1. 1394342831 883648
w1	w2	10	3	1	0. 0645385211 3757116
w1	w3	10	7	2	-0. 0896121586 8968704
w1	w5	10	8	1	-0. 9162907318 74155
w1	w9	10	12	4	0. 0645385211 3757116
w2	w3	3	7	1	0. 4212134650 763035
w2	w4	3	4	1	0. 9808292530 117262

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w3	w4	7	4	1	0.13353139262452257
w3	w5	7	8	2	0.13353139262452257
w3	w6	7	7	1	-0.42608439531090014
w4	w5	4	8	1	0
w4	w6	4	7	1	0.13353139262452257
w5	w6	8	7	2	0.13353139262452257
w5	w7	8	4	1	0
w5	w9	8	12	1	-1.0986122886681098
w6	w7	7	4	1	0.13353139262452257
w6	w8	7	7	1	-0.42608439531090014
w6	w9	7	12	1	-0.9650808960435872
w7	w8	4	7	2	0.8266785731844679
w8	w8	7	7	1	-0.42608439531090014

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w8	w9	7	12	2	-0.2719337154836418
w9	w9	12	12	2	-0.8109302162163288

### 8.3.9.13 Word frequency statistics

Based on the word splitting results, this component outputs the words in their original order and calculates the frequency that a word occurs in the document (docContent) specified by the document ID column (docId).

Parameter settings

**Input parameters:** docId column and docContent column generated by the Word Splitting component.

**Two output parameters:**

- **Output port 1:** The output table contains the id, word, and count columns.  
count: indicates the frequency that a word occurs in each document.
- **Output port 2:** The output table contains the id and word columns.

The table output by the second output port lists words in order of occurrence in the document. The table does not calculate the frequency of the occurrence. Therefore, a word may have multiple table entries in the same document. The output table format is compatible with the Word2Vec component.

Examples

In the Alibaba Cloud word splitting data, the two columns in the output table are used as the input parameters for word frequency calculation.

- **Select the docId column:** id.
- **Select the docContent column:** After the word frequency calculation is performed, the result is displayed by output port 1 on this component.

PAI command

```
pai -name doc_word_stat
-project algo_public
```

```
-DinputTableName=doc_test_split_word
-DdocId=id
-DdocContent=content
-DoutputTableNameMulti=doc_test_stat_multi
-DoutputTableNameTriple=doc_test_stat_triple
-DinputTablePartitions="region=cctv_news"
```

## Algorithm parameters

Table 8-78: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	The name of the input table.	-	-
<b>docId</b>	The name of the document ID column.	Only one column can be specified.	-
<b>docContent</b>	The name of the document content column.	Only one column can be specified.	-
<b>outputTableNameMulti</b>	The name of the output table that lists words in the document content after word splitting. Documents are specified by the <b>docId</b> column and their contents are specified by the <b>docContent</b> column. The words are listed in the order that they occur within the documents.	-	-
<b>outputTableNameTriple</b>	The name of the output table that lists the words and the frequency of the occurrence of these words in the documents. The documents are specified by the <b>docId</b> column and their contents are specified by the <b>docContent</b> column.	-	-



Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for training, in the format of <code>partition_name=value</code> . To specify multiple partitions, use the following format: <code>name1=value1/name2=value2</code> . Separate multiple partitions with commas (,).	-	All partitions in the input table are selected by default.

#### 8.3.9.14 TF-IDF

Term frequency-inverse document frequency (TF-IDF) is typically used as a weighting technology in information retrieval and text mining. TF-IDF is a statistical method to evaluate the importance of a word for a document in a collection or corpus. The importance of a word increases as the frequency that it occurs within the document increases. The importance decreases as the frequency that the word occurs in the corpus increases. TF-IDF is frequently used by search engines as a tool in scoring and ranking the correlation between documents and user queries.

- For more information, see TF-IDF in Wikipedia.
- The TF-IDF component is used to calculate the TF-IDF value of each word that appears in a collection of documents based on word frequency statistics.

#### Examples

The output table in the example of the word frequency statistics component is used as the input table for the TF-IDF component. The corresponding parameter settings are as follows:

- Select the document ID column: `id`
- Select the word column: `word`
- Select the word count column: `count`

The output table contains the following columns: `docid`, `word`, `word_count` (frequency that a certain word occurs in the current document), `total_word_count` (total number of words in the current document), `doc_count` (total number of documents that contain the current word), `total_doc_count` (total number of documents), `tf`, `idf`.

## PAI command

```
pai -name tfidf
-project algo_public
-DinputTableName=rgdoc_split_triple_out
-DdocIdCol=id
-DwordCol=word
-DcountCol=count
-DoutputTableName=rg_tfidf_out;
```

## Algorithm parameters

Table 8-79: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	Table name	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for word splitting.	<b>This value must be in the <code>partition_name=value</code> format. To specify multiple partitions, use the following format: <code>name1=value1/</code> <code>name2=value2</code>. <b>Separate multiple partitions with commas (,).</b></b>	<b>All partitions in the input table are selected by default.</b>
<b>docIdCol</b>	<b>Required.</b> The name of the document ID column.	<b>Only one column can be specified.</b>	-
<b>wordCol</b>	<b>Required.</b> The name of the word column.	<b>Only one column can be specified.</b>	-
<b>countCol</b>	<b>Required.</b> The name of the count column.	<b>Only one column can be specified.</b>	-
<b>outputTableName</b>	<b>Required.</b> The name of the output table.	Table name	-

Parameter	Description	Valid values	Default value
outputTablePartition	The partitions in the output table.	Partition name	The output table is non-partitioned by default.

### 8.3.9.15 PLDA

Latent Dirichlet Allocation (LDA) is a topic model that outputs the topic of a document. It outputs the topic of each document based on probability distribution. LDA is an unsupervised learning algorithm that does not require a manually tagged training set. Instead, it only requires a set of documents and the number of topics (k). It is used in text mining, including text topic recognition, text classification, and text similarity calculation.

Parameter settings

Table 8-80: Parameters

Parameter	Description
Topics	The number of topics output by LDA.
Alpha	The AlphaPrior Dirichlet distribution parameter of $P(z/d)$ .
Beta	The AlphaPrior Dirichlet distribution parameter of $P(w/z)$ .
Burn-in Iterations	The number of burn-in iterations. The parameter value must be less than the total number of iterations. The default value is 100.
Total Iterations	Optional. The total number of iterations. The parameter value must be a positive integer. The default value is 150.



**Note:**

**z** represents topics, **w** represents words, and **d** represents documents.

Input and output settings

- **Input:**

The data must be in the sparse matrix format. For more information about the format, see the data format description section. You can use the Convert Row,

Column, and Value to KV Pair component to convert the data. Input format as shows the following picture.

id	features
2	38:3.0,39:1.0,40:3.0,41:1.0,42:1.0,43:2.0,44:1.0,45:1.0,46:1.0,47:1.0,48:1.0,49:2.0,50:1.0,51:1.0,52:1.0,53:1.0,54:1.0,55:1.0,56:1.0,57:1.0,58:1.0,59:1.0,60:1.0,61:1.0,62:1.0,63:1.0,64:1.0,65:1.0,66:1.0,67:1.0,68:1.0,69:1.0,70:1.0,71:1.0,72:1.0,73:1.0,74:1.0,75:1.0,76:1.0,77:2.0
1	0:1.0,1:2.0,3:1.0,4:1.0,5:1.0,6:1.0,7:1.0,8:1.0,9:1.0,10:1.0,11:1.0,12:1.0,13:1.0,14:2.0,15:1.0,16:1.0,17:1.0,18:1.0,19:1.0,20:1.0,21:1.0,22:1.0,23:1.0,24:1.0,25:1.0,26:1.0,27:1.0,28:1.0,29:1.0,30:1.0,31:1.0,32:1.0,33:1.0,34:1.0,35:1.0,36:2.0,39:2.0,77:3.0

- Column 1: the ID of a document.
- Column 2: KV data of the word and how frequently it occurs.

• **Output:**

The following tables are generated in sequence: topic-word frequency contribution table, P(w/z) table, P(z/w) table, P(d/z) table, P(z/d) table, and P(z) table.

The following picture shows the output format of the topic-word frequency contribution table.

wordid	topic_0	topic_1
0	1	0
1	2	0
2	0	0
3	1	0
4	1	0
5	1	0
6	1	0
7	1	0
8	0	1
9	1	0
10	1	0
11	1	0
12	1	0

PAI command

```
pai -name PLDA
-project algo_public
-DinputTableName=lda_input
-DtopicNum=10
-topicWordTableName=lda_output;
```

Algorithm parameters

Table 8-81: Parameters

Parameter	Description	Valid values	Default value
inputTableName	Required. The name of the input table.	Table name	-

Parameter	Description	Valid values	Default value
<b>inputTablePartitions</b>	Optional. The partitions selected from the input table for word splitting.	This value must be in the <code>partition_name=value</code> format. To specify multiple partitions, use the following format: <code>name1=value1/</code> <code>name2=value2</code> . Separate multiple partitions with commas (,).	All partitions in the input table are selected by default.
<b>selectedColNames</b>	Optional. The names of the columns selected from the input table for LDA.	Separate multiple columns with commas (,).	All columns in the input table are selected by default.
<b>topicNum</b>	Required. The number of topics.	[2, 500]	-
<b>kvDelimiter</b>	Optional. The delimiter used to separate the key and value.	Space, comma (,), and colon (:)	The default delimiter is a colon (:).
<b>itemDelimiter</b>	Optional. The delimiter used to separate keys.	Space, comma (,), and colon (:)	The default delimiter is a space .
<b>alpha</b>	Optional. The prior Dirichlet distribution parameter of $P(z/d)$ .	$(0, \infty)$	0.1
<b>beta</b>	Optional. The prior Dirichlet distribution parameter of $P(w/z)$ .	$(0, \infty)$	0.01
<b>topicWordTableName</b>	Required. The name of the topic-word frequency contribution table.	Table name	-

Parameter	Description	Valid values	Default value
<b>pwzTableName</b>	Optional. The name of the P(w/z) table.	Table name	No P(w/z) table is output by default.
<b>pzwTableName</b>	Optional. The name of the P(z/w) table.	Table name	No P(z/w) table is output by default.
<b>pdzTableName</b>	Optional. The name of the P(d/z) table.	Table name	No P(d/z) table is output by default.
<b>pzdTableName</b>	Optional. The name of the P(z/d) table.	Table name	No P(z/d) table is output by default.
<b>pzTableName</b>	Optional. The name of the P(z) table.	Table name	No P(z) table is output by default.
<b>burnInIterations</b>	Optional. The number of burn-in iterations.	A positive integer	This value must be smaller than the total number of iterations. The default value is 100.
<b>totalIterations</b>	Optional. The number of iterations.	A positive integer	150

### 8.3.9.16 Word2Vec

Word2Vec is an open-source algorithm used to convert words into vectors. By training neural networks, Word2Vec can map words to K-dimensional space vectors and map word vectors to semantics.

For information about the Google Word2Vec toolkit, visit <https://code.google.com/p/word2vec/>.

Parameter settings

- **Dimension of Word Features:** We recommend a value from 0 to 1000.
- **Downsampling Threshold:** We recommend a value from 1e-3 to 1e-5.
- **Input:** inputs a word column and a vocabulary.
- **Output:** generates a word vector table and a vocabulary.

## PAI command

```
pai -name Word2Vec
-project algo_public
-DinputTableName=w2v_input
-DwordColName=word
-DoutputTableName=w2v_output;
```

## Algorithm parameters

Table 8-82: Parameters

Parameter	Description	Valid values	Default value
<b>inputTableName</b>	<b>Required.</b> The name of the input table.	Table name	-
<b>inputTablePartitions</b>	<b>Optional.</b> The partitions selected from the input table for word splitting.	The parameter value must be in the <code>partition_name=value</code> format. To specify multiple partitions, use the following format: <code>name1=value1/ name2=value2</code> . Separate multiple partitions with commas (,).	All partitions in the input table are selected by default.
<b>wordColName</b>	<b>Required.</b> The name of the word column. Each row in the word column contains a single word. <code>&lt;/s&gt;</code> is used to break lines in the corpus.	Column name	-
<b>inVocabularyTableName</b>	<b>Optional.</b> The name of the input word list, which contains the wordcount output of <code>inputTableName</code> .	Table name	Word count is performed for the input table by default.

Parameter	Description	Valid values	Default value
<b>inVocabularyPartition</b>	Optional. The partitions in the input word list.	Partition name	By default, all partitions in the table specified by <b>inVocabularyTableName</b> are selected.
<b>layerSize</b>	Optional. The dimension of word features.	0 to 1000	100
<b>cbow</b>	Optional. The language model.	1: cbow. 0: skip-gram.	0
<b>window</b>	Optional. The size of the word window.	A positive integer	5
<b>minCount</b>	Optional. The minimum frequency of word truncation.	A positive integer	5
<b>hs</b>	Optional. This parameter specifies whether to use hierarchical softmax.	1: Hierarchical softmax is used . 0: Hierarchical softmax is not used .	1
<b>negative</b>	Optional. The negative sampling.	0: Negative sampling is unavailable. Recommended value range: 5 to 10 .	0
<b>sample</b>	Optional. The downward sampling threshold .	0 or smaller values: downward sampling is unavailable. Recommended value range: 1e-3 to 1e-5.	0
<b>alpha</b>	Optional. The initial learning rate .	A value greater than 0	0.025



Parameter	Description	Valid values	Default value
iterTrain	Optional. The number of training iterations.	A value greater than or equal to 1	1
randomWindow	Optional. This parameter specifies whether to randomly set the size of the window.	1: The window size is randomly generated. The window size value will range from 1 to 5. 0: The window size is determined by the window parameter.	1
outVocabularyTable	Optional. The name of the output word list.	Table name	No Output Word List is generated by default.
outVocabularyPartition	Optional. The partition in the output word list.	Partition name	The output word list is non-partitioned by default.
outputTableName	Required. The name of the output table.	Table name	-
outputPartition	Optional. The information about partitions in the output table.	Partition name	The output table is non-partitioned by default.

## 8.3.10 Network analysis

### 8.3.10.1 K-core

The k-core of a graph is the largest subgraph in which every vertex is connected to at least k other vertices within the subgraph. The coreness of a vertex is k if it belongs to the k-core but is not included in the (k+1)-core. Therefore, the coreness of a vertex whose degree is 1 must be 0. The graph coreness is equal to that of the vertex with the largest coreness.

Parameter settings

**k:** Required. The value of the coreness. Default value: 3.

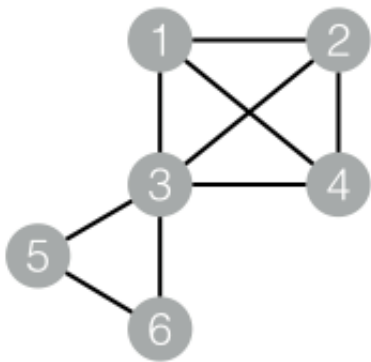
## Examples - Testing data

**SQL statement to generate data:**

```
drop table if exists KCore_func_test_edge;
create table KCore_func_test_edge as
select * from (
 select '1' as flow_out_id,
 '2' as flow_in_id from dual union all
 select '1' as flow_out_id,
 '3' as flow_in_id from dual union all
 select '1' as flow_out_id,
 '4' as flow_in_id from dual union all
 select '2' as flow_out_id,
 '3' as flow_in_id from dual union all
 select '2' as flow_out_id,
 '4' as flow_in_id from dual union all
 select '3' as flow_out_id,
 '4' as flow_in_id from dual union all
 select '3' as flow_out_id,
 '5' as flow_in_id from dual union all
 select '3' as flow_out_id,
 '6' as flow_in_id from dual union all
 select '5' as flow_out_id,
 '6' as flow_in_id from dual)tmp;
```

*Figure 8-5: Graph structure* shows the group structure.

Figure 8-5: Graph structure



Set K to 2. *Figure 8-6: Output* shows the output.

Figure 8-6: Output

node1	node2
1	2
1	3
1	4
2	1
2	3
2	4
3	1
3	2
3	4
4	1
4	2
4	3

PAI command

```
pai -name KCore
-project algo_public
-DinputEdgeTableName=KCore_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=KCore_func_test_result
-Dk=2;
```

Algorithm parameters

Table 8-83: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the edge table.	Yes	-

Parameter	Description	Required	Default value
<b>toVertexCol</b>	The end vertex column in the edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64
<b>k</b>	The number of cores.	Yes	3

### 8.3.10.2 Single-source shortest path

The single-source shortest path (SSSP) refers to the shortest path between a vertex and all other vertices as calculated by the Dijkstra algorithm.

Parameter settings

**Start Vertex ID:** Required. The ID of the start vertex used to calculate the shortest paths.

Examples - Testing data

**SQL statement to generate data:**

```
drop table if exists SSSP_func_test_edge;
create table SSSP_func_test_edge
as select
flow_out_id,flow_in_id,edge_weight from (
select "a" as flow_out_id,
"b" as flow_in_id,
1.0 as edge_weight from dual union all
select "b" as flow_out_id,
"c" as flow_in_id,
2.0 as edge_weight from dual union all
select "c" as flow_out_id,
"d" as flow_in_id,
1.0 as edge_weight from dual union all
select "b" as flow_out_id,
"e" as flow_in_id,
```

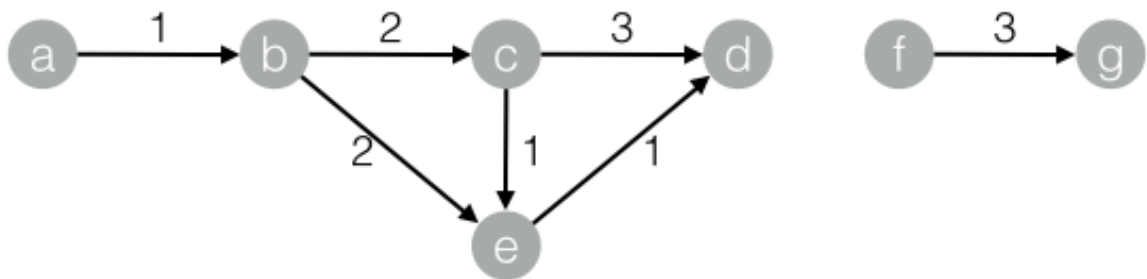
```

2.0 as edge_weight from dual union all
select "e" as flow_out_id,
"d" as flow_in_id,
1.0 as edge_weight from dual union all
select "c" as flow_out_id,
"e" as flow_in_id,
1.0 as edge_weight from dual union all
select "f" as flow_out_id,
"g" as flow_in_id,
3.0 as edge_weight from dual union all
select "a" as flow_out_id,
"d" as flow_in_id,
4.0 as edge_weight from dual) tmp ;

```

Figure 8-7: *Graph structure* shows the graph structure.

Figure 8-7: Graph structure



#### Output

start_node	dest_node	distance	distance_cnt
a	b	1.0	1
a	c	3.0	1
a	d	4.0	3
a	a	0.0	0
a	e	3.0	1

#### PAI command

```

pai -name SSSP
-project algo_public
-DinputEdgeTableName=SSSP_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=SSSP_func_test_result
-DhasEdgeWeight=true
-DedgeWeightCol=edge_weight

```

-DstartVertex=a;

## Algorithm parameters

Table 8-84: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64
<b>startVertex</b>	The ID of the start vertex.	Yes	-
<b>hasEdgeWeight</b>	Specifies whether the edges in the input edge table have weights.	No	false
<b>edgeWeightCol</b>	The edge weight column in the input edge table.	No	-

### 8.3.10.3 PageRank

The PageRank algorithm is used to sort and calculate the rankings of web pages based on their link sources.

#### Features

The basic principle of the PageRank algorithm is as follows: The more web pages that direct to a web page, the more importance or higher quality the web page has. In addition to the number of links directing to a web page, the weight of the web page and the number of outgoing links are also considered during page ranking. For a social network of users, the edge weight is an important factor in addition to the influence of the users. For example, a Sina Weibo user is more likely to have influence on their family, friends, classmates, and colleagues than they will on followers with a weaker relationship. In the social network, the edge weight is equivalent to the user-to-user relationship strength index. The PageRank formula with connection weight is as follows:

$$W(A) = (1 - d) + d * (\sum_i W(i) * C(Ai))$$

In the formula,  $W(i)$  represents the weight of node  $i$ ,  $C(A,i)$  represents the link weight, and  $d$  represents the damping coefficient.  $W$  is the influence index of each user and represents the node weight after the algorithm iteration becomes stable.

#### Parameter settings

**Maximum Iterations: Optional.** The number of iterations performed before the algorithm automatically converges. Default value: 30.

#### Examples - Testing data

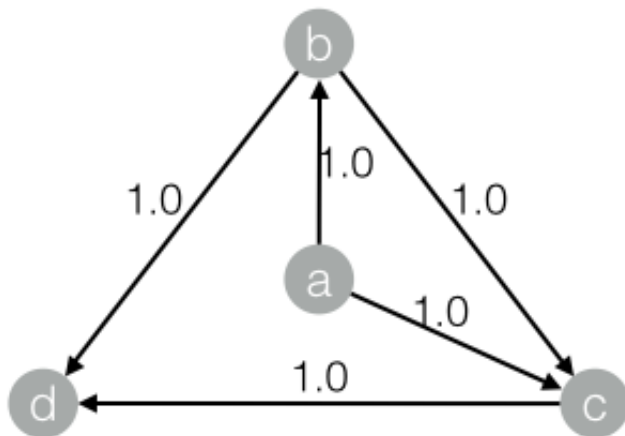
**SQL statement to generate data:**

```
drop table if exists PageRankWithWeight_func_test_edge;
create table PageRankWithWeight_func_test_edge
as select * from (
select 'a' as flow_out_id,
'b' as flow_in_id,
1.0 as weight from dual union all
select 'a' as flow_out_id,
'c' as flow_in_id,
1.0 as weight from dual union all
select 'b' as flow_out_id,
'c' as flow_in_id,
1.0 as weight from dual union all
select 'b' as flow_out_id,
```

```
'd' as flow_in_id,
1.0 as weight from dual union all
select 'c' as flow_out_id,
'd' as flow_in_id,1.0 as weight from dual)tmp ;
```

Figure 8-8: Graph structure shows the graph structure.

Figure 8-8: Graph structure



Output

node	weight
a	0.0375
b	0.06938
c	0.12834
d	0.20556

PAI command

```
pai -name PageRankWithWeight
-project algo_public
-DinputEdgeTableName=PageRankWithWeight_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=PageRankWithWeight_func_test_result
-DhasEdgeWeight=true
-DedgeWeightCol=weight
```



```
-DmaxIter 100;
```

## Algorithm parameters

Table 8-85: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64
<b>hasEdgeWeight</b>	Specifies whether the edges in the input edge table have weights.	No	false
<b>edgeWeightCol</b>	The edge weight column in the input edge table.	No	-
<b>maxIter</b>	The maximum number of iterations.	No	30

### 8.3.10.4 Label propagation clustering

Graph clustering is used to divide a graph into subgraphs based on the topology of the graph so that the links between the nodes in a subgraph are more than the links between the subgraphs. The label propagation algorithm (LPA) is a graph-based semi-supervised machine learning algorithm. The labels of a node (community) depend on those of the neighboring nodes. The degree of dependence is determined by the similarity between nodes. Data becomes stable by iterative propagation update.

#### Parameters

**Maximum Iterations: Optional.** The maximum number of iterations. Default value: 30.

#### Examples - Testing data

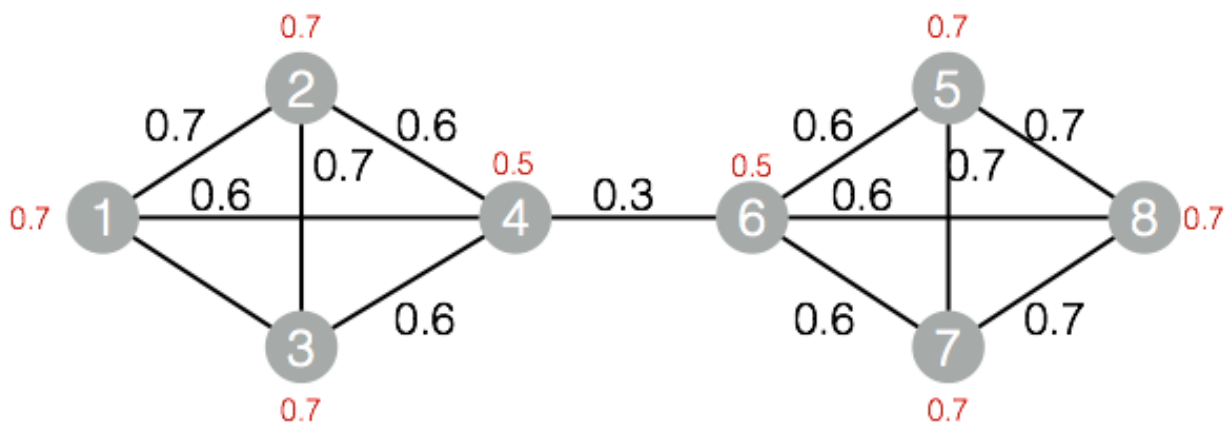
##### SQL statement to generate data:

```
drop table if exists LabelPropagationClustering_func_test_edge;
create table LabelPropagationClustering_func_test_edge
as select * from (
select '1' as flow_out_id,
'2' as flow_in_id,
0.7 as edge_weight from dual union all
select '1' as flow_out_id,
'3' as flow_in_id,
0.7 as edge_weight from dual union all
select '1' as flow_out_id,
'4' as flow_in_id,
0.6 as edge_weight from dual union all
select '2' as flow_out_id,
'3' as flow_in_id,
0.7 as edge_weight from dual union all
select '2' as flow_out_id,
'4' as flow_in_id,
0.6 as edge_weight from dual union all
select '3' as flow_out_id,
'4' as flow_in_id,
0.6 as edge_weight from dual union all
select '4' as flow_out_id,
'6' as flow_in_id,
0.3 as edge_weight from dual union all
select '5' as flow_out_id,
'6' as flow_in_id,
0.6 as edge_weight from dual union all
select '5' as flow_out_id,
'7' as flow_in_id,
0.7 as edge_weight from dual union all
select '5' as flow_out_id,
'8' as flow_in_id,
0.7 as edge_weight from dual union all
select '6' as flow_out_id,
'7' as flow_in_id,
0.6 as edge_weight from dual union all
select '6' as flow_out_id,
```

```
'8' as flow_in_id,
0.6 as edge_weight from dual union all
select '7' as flow_out_id,
'8' as flow_in_id,
0.7 as edge_weight from dual)tmp ;
drop table if exists LabelPropagationClustering_func_test_node;
create table LabelPropagationClustering_func_test_node
as select * from (
select '1' as node,
0.7 as node_weight from dual union all
select '2' as node,
0.7 as node_weight from dual union all
select '3' as node,
0.7 as node_weight from dual union all
select '4' as node,
0.5 as node_weight from dual union all
select '5' as node,
0.7 as node_weight from dual union all
select '6' as node,
0.5 as node_weight from dual union all
select '7' as node,
0.7 as node_weight from dual union all
select '8' as node,
0.7 as node_weight from dual)tmp ;
```

*Figure 8-9: Group structure* shows the group structure.

Figure 8-9: Group structure



## Output

node	group_id
1	1
2	1
3	1
4	1
5	5
6	5
7	5
8	5

## PAI command

```
pai -name LabelPropagationClustering
-project algo_public
-DinputEdgeTableName=LabelPropagationClustering_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DinputVertexTableName=LabelPropagationClustering_func_test_node
-DvertexCol=node
-DoutputTableName=LabelPropagationClustering_func_test_result
-DhasEdgeWeight=true
-DedgeWeightCol=edge_weight
-DhasVertexWeight=true
-DvertexWeightCol=node_weight
-DrandSelect=true
-DmaxIter=100;
```

## Algorithm parameters

Table 8-86: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-

Parameter	Description	Required	Default value
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>inputVertexTableName</b>	The name of the input vertex table.	Yes	-
<b>inputVertexTablePartitions</b>	The partitions in the input vertex table.	No	The whole table is selected by default.
<b>vertexCol</b>	The vertex column in the input vertex table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64
<b>hasEdgeWeight</b>	Specifies whether the edges in the input edge table have weights.	No	false
<b>edgeWeightCol</b>	The edge weight column in the input edge table.	No	-
<b>hasVertexWeight</b>	Specifies whether the vertices in the input vertex table have weights.	No	false
<b>vertexWeightCol</b>	The vertex weight column in the input vertex table.	No	-

Parameter	Description	Required	Default value
<b>randSelect</b>	<b>Specifies whether the maximum label value is to be randomly selected.</b>	No	false
<b>maxIter</b>	<b>The maximum number of iterations.</b>	No	30

### 8.3.10.5 Label propagation classification

Label propagation classification is a semi-supervised classification algorithm. It uses the label information of labeled nodes to predict the label information for unlabeled nodes.

#### Features

During algorithm execution, the labels of each node are propagated to the neighboring nodes based on the similarity between the nodes. In each step of propagation, a node updates its labels based on the labels of the neighboring nodes so that the node is more similar to the neighboring nodes. The higher the similarity, the more labeling influences the neighboring nodes have on that node, and the easier it is for the labels to be propagated. During label propagation, the labels of the labeled data remain unchanged. These labels serve as sources for propagation to the unlabeled data.

After the iterations end, the probability distributions of similar nodes tend to be similar. These nodes can be classified into the same category. This completes the label propagation.

#### Parameter settings

**Damping factor:** The default value is 0.8. **Convergence factor:** The default value is 0.000001.

#### Examples - Testing data

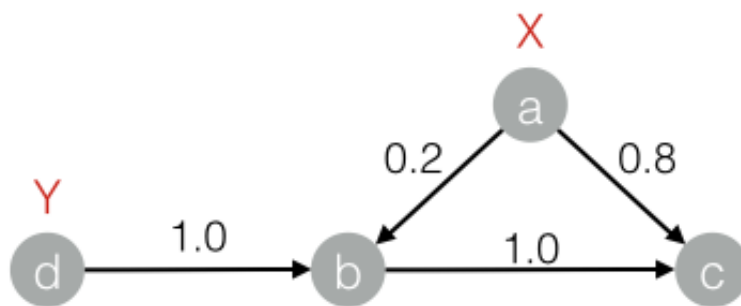
##### SQL statement to generate data:

```
drop table if exists LabelPropagationClassification_func_test_edge;
create table LabelPropagationClassification_func_test_edge
as select * from (
select 'a' as flow_out_id,
'b' as flow_in_id,
0.2 as edge_weight from dual union all
select 'a' as flow_out_id,
```

```
'c' as flow_in_id,
0.8 as edge_weight from dual union all
select 'b' as flow_out_id,
'c' as flow_in_id,
1.0 as edge_weight from dual union all
select 'd' as flow_out_id,
'b' as flow_in_id,
1.0 as edge_weight from dual)tmp ;
drop table if exists LabelPropagationClassification_func_test_node;
create table LabelPropagationClassification_func_test_node
as select * from (
select 'a' as node,
'X' as label,
1.0 as label_weight from dual union all
select 'd' as node,
'Y' as label,
1.0 as label_weight from dual)tmp ;
```

*Figure 8-10: Graph structure* shows the graph structure.

Figure 8-10: Graph structure



#### Output

node	tag	weight
a	X	1.0
b	X	0.16667
b	Y	0.83333
c	X	0.53704
c	Y	0.46296
d	Y	1.0

#### PAI command

```
pai -name LabelPropagationClassification
-project algo_public
-DinputEdgeTableName=LabelPropagationClassification_func_test_edge

-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
```

```
-DinputVertexTableName=LabelPropagationClassification_func_test_node
-DvertexCol=node
-DvertexLabelCol=label
-DoutputTableName=LabelPropagationClassification_func_test_result
-DhasEdgeWeight=true
-DedgeWeightCol=edge_weight
-DhasVertexWeight=true
-DvertexWeightCol=label_weight
-Dalpha=0.8
-Depsilon=0.000001;
```

## Algorithm parameters

Table 8-87: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>inputVertexTableName</b>	The name of the input vertex table.	Yes	-
<b>inputVertexTablePartitions</b>	The partitions in the input vertex table.	No	The whole table is selected by default.
<b>vertexCol</b>	The vertex column in the input vertex table.	Yes	-
<b>vertexLabelCol</b>	The vertex label column in the input vertex table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-



Parameter	Description	Required	Default value
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64
<b>hasEdgeWeight</b>	Specifies whether the edges in the input edge table have weights.	No	false
<b>edgeWeightCol</b>	The edge weight column in the input edge table.	No	-
<b>hasVertexWeight</b>	Specifies whether the vertices in the input vertex table have weights.	No	false
<b>vertexWeightCol</b>	The vertex weight column in the input vertex table.	No	-
<b>alpha</b>	The damping coefficient.	No	0.8
<b>epsilon</b>	The convergence coefficient.	No	0.000001
<b>maxIter</b>	The maximum number of iterations.	No	30

### 8.3.10.6 Modularity

Modularity is used to measure the structure of the community network. It measures the closeness of the communities divided from a network structure. A value larger than 0.3 represents an obvious community structure.

Examples - Testing data

For more information, see [Label propagation clustering](#).

## Output

```
+-----+
| val |
+-----+
| 0.4230769 |
+-----+
```

## PAI command

```
pai -name Modularity
-project algo_public
-DinputEdgeTableName=Modularity_func_test_edge
-DfromVertexCol=flow_out_id
-DfromGroupCol=group_out_id
-DtoVertexCol=flow_in_id
-DtoGroupCol=group_in_id
-DoutputTableName=Modularity_func_test_result;
```

## Algorithm parameters

Table 8-88: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>fromGroupCol</b>	The start vertex group in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>toGroupCol</b>	The end vertex group in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-

Parameter	Description	Required	Default value
outputTablePartitions	The partitions in the output table.	No	-
lifecycle	The lifecycle of the output table.	No	-
workerNum	The number of workers.	No	-
workerMem	The memory size per worker.	No	4096
splitSize	The data split size.	No	64

### 8.3.10.7 Maximum connected subgraph

In an undirected graph G, vertex A is connected to vertex B if a path exists between the two vertices. Graph G contains several subgraphs. Each vertex is connected to other vertices in the same subgraph. Vertices in different subgraphs are not connected. In this case, the subgraphs of graph G are called maximum connected subgraphs.

Features

Examples - Testing data

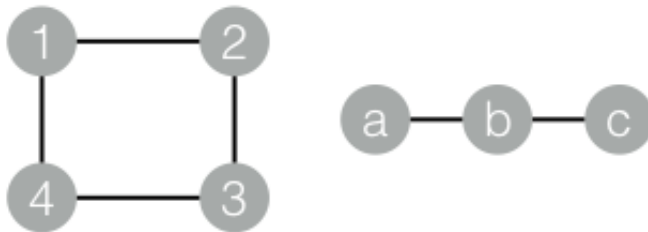
**SQL statement to generate data:**

```
drop table if exists MaximalConnectedComponent_func_test_edge;
create table MaximalConnectedComponent_func_test_edge
as select * from (
select '1' as flow_out_id,
'2' as flow_in_id from dual union all
select '2' as flow_out_id,
'3' as flow_in_id from dual union all
select '3' as flow_out_id,
'4' as flow_in_id from dual union all
select '1' as flow_out_id,
'4' as flow_in_id from dual union all
select 'a' as flow_out_id,
'b' as flow_in_id from dual union all
select 'b' as flow_out_id,
'c' as flow_in_id from dual)tmp;
drop table if exists MaximalConnectedComponent_func_test_result;
```

```
create table MaximalConnectedComponent_func_test_result (node
string, grp_id string);
```

*Figure 8-11: Graph structure* shows the graph structure.

Figure 8-11: Graph structure



Output

node	grp_id
1	4
2	4
3	4
4	4
a	c
b	c
c	c

PAI command

```
pai -name MaximalConnectedComponent
-project algo_public
-DinputEdgeTableName=MaximalConnectedComponent_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=MaximalConnectedComponent_func_test_result;
```

Algorithm parameters

Table 8-89: Parameters

Parameter	Description	Required	Default value
inputEdgeTableName	The name of the input edge table.	Yes	-

Parameter	Description	Required	Default value
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64

### 8.3.10.8 Vertex clustering coefficient

This coefficient is used to calculate the peripheral density of a vertex in an undirected graph G. The density of a star network is 0, and that of a fully meshed network is 1.

Parameter settings

**maxEdgeCnt:** Optional. If the node degree is larger than the value of this parameter, sampling is required. Default value: 500.

Examples - Testing data

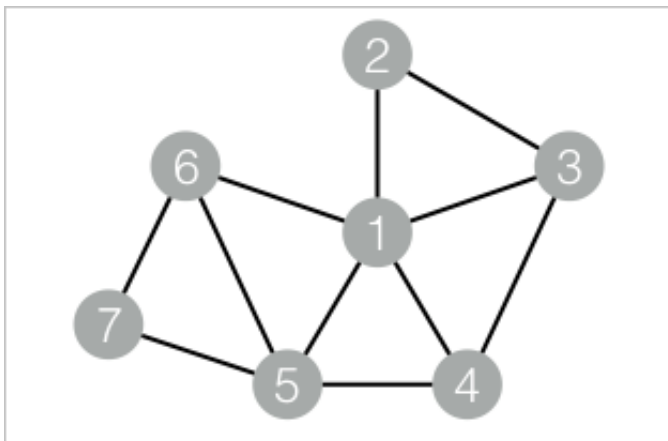
**SQL statement to generate data:**

```
drop table if exists NodeDensity_func_test_edge;
create table NodeDensity_func_test_edge
as select * from (
select '1' as flow_out_id,
'2' as flow_in_id from dual union all
select '1' as flow_out_id,
'3' as flow_in_id from dual union all
```

```
select '1' as flow_out_id,
'4' as flow_in_id from dual union all
select '1' as flow_out_id,
'5' as flow_in_id from dual union all
select '1' as flow_out_id,
'6' as flow_in_id from dual union all
select '2' as flow_out_id,
'3' as flow_in_id from dual union all
select '3' as flow_out_id,
'4' as flow_in_id from dual union all
select '4' as flow_out_id,
'5' as flow_in_id from dual union all
select '5' as flow_out_id,
'6' as flow_in_id from dual union all
select '5' as flow_out_id,
'7' as flow_in_id from dual union all
select '6' as flow_out_id,
'7' as flow_in_id from dual)tmp;
drop table if exists NodeDensity_func_test_result;
create table NodeDensity_func_test_result (node string, node_cnt
bigint, edge_cnt bigint, density double, log_density double);
```

Figure 8-12: Graph structure shows the graph structure.

Figure 8-12: Graph structure



## Output

```
1,5,4,0.4,1.45657
2,2,1,1.0,1.24696
3,3,2,0.66667,1.35204
4,3,2,0.66667,1.35204
5,4,3,0.5,1.41189
6,3,2,0.66667,1.35204
7,2,1,1.0,1.24696
```

## PAI command

```
pai -name NodeDensity
-project algo_public
-DinputEdgeTableName=NodeDensity_func_test_edge
```

```
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=NodeDensity_func_test_result
-DmaxEdgeCnt=500;
```

## Algorithm parameters

Table 8-90: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>maxEdgeCnt</b>	If the node degree is larger than the value of this parameter, sampling is required.	No	500
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64

### 8.3.10.9 Edge clustering coefficient

This coefficient is used to calculate the peripheral density of each edge in an undirected graph G.

Examples - Testing data

**SQL statement to generate data:**

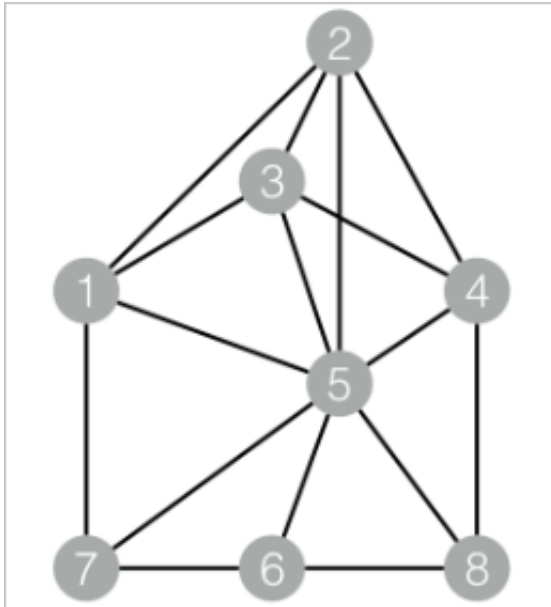
```
drop table if exists EdgeDensity_func_test_edge;
create table EdgeDensity_func_test_edge
as select * from (
select '1' as flow_out_id,
'2' as flow_in_id from dual union all
select '1' as flow_out_id,
'3' as flow_in_id from dual union all
select '1' as flow_out_id,
'5' as flow_in_id from dual union all
select '1' as flow_out_id,
'7' as flow_in_id from dual union all
select '2' as flow_out_id,
'5' as flow_in_id from dual union all
select '2' as flow_out_id,
'4' as flow_in_id from dual union all
select '2' as flow_out_id,
'3' as flow_in_id from dual union all
select '3' as flow_out_id,
'5' as flow_in_id from dual union all
select '3' as flow_out_id,
'4' as flow_in_id from dual union all
select '4' as flow_out_id,
'5' as flow_in_id from dual union all
select '4' as flow_out_id,
'8' as flow_in_id from dual union all
select '5' as flow_out_id,
'6' as flow_in_id from dual union all
select '5' as flow_out_id,
'7' as flow_in_id from dual union all
select '5' as flow_out_id,
'8' as flow_in_id from dual union all
select '7' as flow_out_id,
'6' as flow_in_id from dual union all
select '6' as flow_out_id,
'8' as flow_in_id from dual)tmp;
drop table if exists EdgeDensity_func_test_result;
```



```
create table EdgeDensity_func_test_result (node1 string, node2
string, node1_edge_cnt bigint, node2_edge_cnt bigint, triangle_c
nt bigint, density double);
```

Figure 8-13: Graph structure shows the graph structure.

Figure 8-13: Graph structure



## Output

```
1,2,4,4,2,0.5
2,3,4,4,3,0.75
2,5,4,7,3,0.75
3,1,4,4,2,0.5
3,4,4,4,2,0.5
4,2,4,4,2,0.5
4,5,4,7,3,0.75
5,1,7,4,3,0.75
5,3,7,4,3,0.75
5,6,7,3,2,0.66667
5,8,7,3,2,0.66667
6,7,3,3,1,0.33333
7,1,3,4,1,0.33333
7,5,3,7,2,0.66667
8,4,3,4,1,0.33333
8,6,3,3,1,0.33333
```

## PAI command

```
pai -name EdgeDensity
-project algo_public
-DinputEdgeTableName=EdgeDensity_func_test_edge
-DfromVertexCol=flow_out_id
```

```
-DtoVertexCol=flow_in_id
-DoutputTableName=EdgeDensity_func_test_result;
```

### Algorithm parameters

Table 8-91: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64

### 8.3.10.10 Counting triangle

All triangles can be output to an undirected graph G.

#### Parameter settings

**maxEdgeCnt:** Optional. If the node degree is larger than the value of this parameter, sampling is required. Default value: 500.

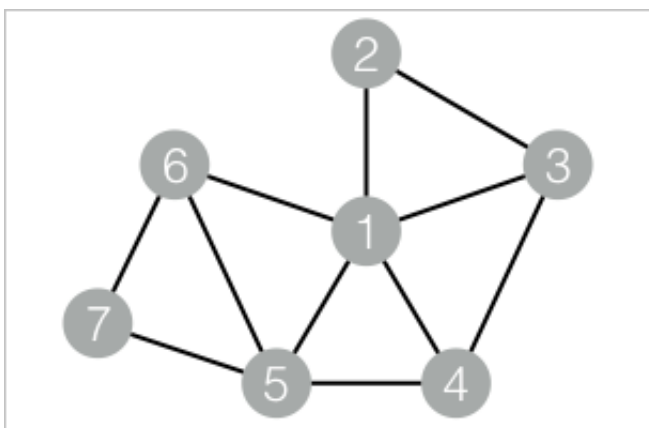
## Examples - Testing data

**SQL statement to generate data:**

```
drop table if exists TriangleCount_func_test_edge;
create table TriangleCount_func_test_edge
as select * from (
select '1' as flow_out_id,
'2' as flow_in_id from dual union all
select '1' as flow_out_id,
'3' as flow_in_id from dual union all
select '1' as flow_out_id,
'4' as flow_in_id from dual union all
select '1' as flow_out_id,
'5' as flow_in_id from dual union all
select '1' as flow_out_id,
'6' as flow_in_id from dual union all
select '2' as flow_out_id,
'3' as flow_in_id from dual union all
select '3' as flow_out_id,
'4' as flow_in_id from dual union all
select '4' as flow_out_id,
'5' as flow_in_id from dual union all
select '5' as flow_out_id,
'6' as flow_in_id from dual union all
select '5' as flow_out_id,
'7' as flow_in_id from dual union all
select '6' as flow_out_id,
'7' as flow_in_id from dual)tmp;
drop table if exists TriangleCount_func_test_result;
create table TriangleCount_func_test_result (node1 string, node2
string, node3 string);
```

*Figure 8-14: Graph structure* shows the graph structure.

Figure 8-14: Graph structure



## Output

```
1,2,3
1,3,4
1,4,5
1,5,6
5,6,7
```

## PAI command

```
pai -name TriangleCount
-project algo_public
-DinputEdgeTableName=TriangleCount_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
-DoutputTableName=TriangleCount_func_test_result;
```

## Algorithm parameters

Table 8-92: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-

Parameter	Description	Required	Default value
maxEdgeCnt	If the node degree is larger than the value of this parameter , sampling is required.	No	500
workerNum	The number of workers.	No	-
workerMem	The memory size per worker.	No	4096
splitSize	The data split size.	No	64

### 8.3.10.11 Tree depth

In a tree network, this component outputs the depth of each node in a tree and the tree ID.

Examples - Testing data

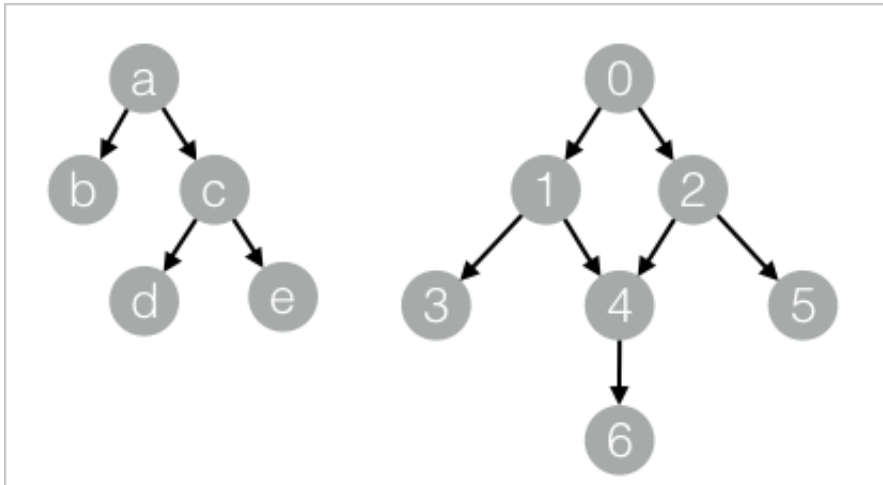
**SQL statement to generate data:**

```
drop table if exists TreeDepth_func_test_edge;
create table TreeDepth_func_test_edge
as select * from (
select '0' as flow_out_id,
'1' as flow_in_id from dual union all
select '0' as flow_out_id,
'2' as flow_in_id from dual union all
select '1' as flow_out_id,
'3' as flow_in_id from dual union all
select '1' as flow_out_id,
'4' as flow_in_id from dual union all
select '2' as flow_out_id,
'4' as flow_in_id from dual union all
select '2' as flow_out_id,
'5' as flow_in_id from dual union all
select '4' as flow_out_id,
'6' as flow_in_id from dual union all
select 'a' as flow_out_id,
'b' as flow_in_id from dual union all
select 'a' as flow_out_id,
'c' as flow_in_id from dual union all
select 'c' as flow_out_id,
'd' as flow_in_id from dual union all
select 'c' as flow_out_id,
'e' as flow_in_id from dual)tmp;
drop table if exists TreeDepth_func_test_result;
```

```
create table TreeDepth_func_test_result (node string, root string
, depth bigint);
```

*Figure 8-15: Graph structure* shows the graph structure.

Figure 8-15: Graph structure



#### Output

```
0,0,0
1,0,1
2,0,1
3,0,2
4,0,2
5,0,2
6,0,3
a,a,0
b,a,1
c,a,1
d,a,2
e,a,2
```

#### PAI command

```
pai -name TreeDepth
-project algo_public
-DinputEdgeTableName=TreeDepth_func_test_edge
-DfromVertexCol=flow_out_id
-DtoVertexCol=flow_in_id
```

```
-DoutputTableName=TreeDepth_func_test_result;
```

Algorithm parameters

Table 8-93: Parameters

Parameter	Description	Required	Default value
<b>inputEdgeTableName</b>	The name of the input edge table.	Yes	-
<b>inputEdgeTablePartitions</b>	The partitions selected from the input edge table.	No	The whole table is selected by default.
<b>fromVertexCol</b>	The start vertex column in the input edge table.	Yes	-
<b>toVertexCol</b>	The end vertex column in the input edge table.	Yes	-
<b>outputTableName</b>	The name of the output table.	Yes	-
<b>outputTablePartitions</b>	The partitions in the output table.	No	-
<b>lifecycle</b>	The lifecycle of the output table.	No	-
<b>workerNum</b>	The number of workers.	No	-
<b>workerMem</b>	The memory size per worker.	No	4096
<b>splitSize</b>	The data split size.	No	64

## 8.3.11 Tools

### 8.3.11.1 SQL script

You can use the SQL script editor to write SQL statements.

1. Drag and drop the SQL Script component onto the canvas.
2. Connect the input table to the SQL Script component, and then click SQL Script.

The following configuration pane is displayed.

### 3. Write an SQL script in the text box.

- An SQL script supports one to four inputs and one output.
- You can write only one SQL statement.
- The input data is automatically mapped to tables t1 through t4. You can directly call `#{t1}`, `#{t2}`, `#{t3}`, and `#{t4}` without specifying the table names.
- The sample SQL script calculates the number of rows in the input table.

## 8.4 Automatic parameter tuning with AutoML

### 8.4.1 Automatic parameter tuning with AutoML

This topic describes the automatic parameter tuning feature of AutoML.

Parameter

1. [Log in to machine learning console](#). In the left-side navigation pane, click Experiments.
2. Click an experiment to go to the canvas of the experiment.



**Note:**

This topic uses air quality prediction as an example.

3. In the upper-left corner of the canvas, choose Auto ML > Auto Parameter Tuning.
4. On the Auto Parameter Tuning page, select an algorithm for parameter tuning, and click Next.



**Note:**

You can select only one algorithm to tune at a time.

5. In the Configure Parameter Tuning module, set the Parameter Tuning Method parameter and click Next.

Alibaba Cloud Machine Learning Platform for AI provides four parameter tuning methods. For more information, see [Parameter adjustment method](#).



6. In the Configure Model Output module, set the model output parameters and click Next.

Parameter	Description
Evaluation Criteria	You can select one evaluation standard from the following four dimensions: AUC, F1-score, PRECISION, and RECALL.
Saved Models	You can save up to five models. The system ranks models based on the Evaluation Criteria setting you select and save the top ranked models according to the number entered in the Saved Models field.
Pass Down Model	This switch is turned on by default. If the switch is turned off, the model generated by the default parameters of the current component are passed down to the node of the subsequent component. If the switch is turned on, the optimal model generated by automatic parameter tuning are passed down to the node of the subsequent component.

7. In the upper-left corner of the canvas, click Run to run the automatic parameter tuning algorithm, as shown in the following figure.



**Note:**

After the preceding configuration is complete, the Auto ML switch of the related algorithm is turned on. You can turn the switch on or off as needed.

8. Right-click a model component and choose Edit AutoML Parameters from the shortcut menu to modify its AutoML configuration parameters.

#### Output model display

1. During parameter tuning, right-click the target model component and choose Parameter Tuning Details from the shortcut menu.
2. On the AutoML-Parameter Tuning Details page, click the Metrics tab to view the current tuning progress and the running status of each model.

3. You can sort candidate models according to indicators (AUC, F1-score, Accuracy, and Recall Rate).
4. In the View Details column, you can click Log or Parameter to view the logs and parameters of each candidate model.

#### Parameter tuning effect display

1. On the AutoML-Parameter Tuning Details page, you can click the Charts tab to view the Model Evaluation and Comparison and Hyperparameter Iteration Result Comparison charts.
2. You can view the growth trend of the evaluation indicators of updated parameters in the Hyperparameter Iteration Result Comparison chart.

#### Model storage

1. [Log in to machine learning console](#). In the left-side navigation pane, click Models.
2. Click Experiment Model to open the experiment model folder.
3. Click the corresponding experiment folder to view the model saved with Auto ML.
4. (Optional) You can apply a model to other experiments by dragging the model to the canvas of the target experiment.

## 8.4.2 Parameter tuning methods

AutoML supports four parameter tuning methods.

#### Evolutionary Optimizer

##### Principle:

1. Randomly selects A parameter candidate sets (where A indicates the number of exploration samples).
2. Takes the N parameter candidate sets with higher evaluation indicators as the parameter candidate sets of the next iteration.
3. Continues the exploration within R times (where R indicates the convergence coefficient) as the standard deviation range around these parameters to explore new parameter sets. The new parameter sets replace the last A-N parameter sets by the evaluation indicator in the previous round.
4. Iterates the exploration for M rounds (where M indicates the number of explorations) until the optimal parameter set is found, according to the preceding logic.

Based on the preceding principle, the final number of models is  $A + (A - N) \times M$ .



**Note:**

The first value of N is  $A/2 - 1$ . During iteration, the default value is  $A/2 - 1$  (rounded up).

Parameter	Description
Data Splitting Ratio	Splits input data sources into training and evaluation sets. 0.7 indicates that 70 % of the data is used for model training and 30% for evaluation.
Exploration Samples	The number of parameter sets of each iteration. The higher the number, the greater the accuracy, the larger the calculation. This parameter must be set to a positive integer in the range of 5 to 30.
Explorations	The number of iterations. The higher the number of iterations, the greater the search accuracy, the larger the calculation. This parameter must be set to a positive integer in the range of 1 to 10.
Convergence Coefficient	Tunes the exploration ranges (R times the standard deviation range search ). The smaller the range, the faster the convergence (however, optimal parameters may be missed). Valid values : 0.1 to 1 (one floating point after the decimal point).



**Note:**

You must enter the tuning range for each parameter. If the current parameter range is not configured, the parameter range is set by default.

Random Search

**Principle:**

1. Randomly selects a value for each parameter within the parameter range.

2. Enters random values into a set of parameters for model training.
3. Performs M rounds (where M indicates the number of iterations) and then sorts the output models.

Parameter	Description
Data Splitting Ratio	Splits input data sources into training and evaluation sets. 0.7 indicates that 70 % of the data is used for model training and 30% for evaluation.
Iterations	The number of searches in the configured range. Valid values: 2 to 50.

**Note:**

You must enter the tuning range for each parameter. If the current parameter range is not configured, the parameter range is set by default.

## Grid Search

**Principle:**

1. Splits the value range of each parameter into N segments (where N indicates the number of split grids).
2. Randomly takes a value from the N segments. Assuming that there are M parameters,  $N^M$  parameter sets can be combined.
3. According to the  $N^M$  parameter sets,  $N^M$  models are generated by training. The models are then sorted.

Parameter	Description
Data Splitting Ratio	Splits input data sources into training and evaluation sets. 0.7 indicates that 70 % of the data is used for model training and 30% for evaluation.
Grids	The number of split grids. Valid values: 2 to 10.

**Note:**

**You must enter the tuning range for each parameter. If the current parameter range is not configured, the parameter range is set by default.**

#### Custom Parameters

- **You can enumerate parameter candidate sets. The system then helps score all the combinations of the candidate sets.**
- **You can define enumeration ranges and separate parameters with commas (.). If the ranges are not configured, the default ranges of parameters are tuned.**

## 8.5 Terms and acronyms

### 8.5.1 Terms

**This topic lists the basic terms used in machine learning.**

#### experiment

**A user-created data mining workflow.**

#### project

**The basic object in MaxCompute. A project is also known as a workspace. A project contains other objects, such as tables and instances.**

#### component

**The minimum operating unit that you can invoke and execute on Apsara Stack Machine Learning Platform for AI. You can use components to import and export data, process data, analyze data, train models, and make predictions.**

### 8.5.2 Acronyms

**This topic describes the acronyms used in the Machine Learning Platform for AI User Guide.**

#### MaxCompute

**MaxCompute (formerly known as ODPS) is a data processing platform developed by Alibaba Cloud for large-scale data warehousing. MaxCompute can store and compute structured data in batches to meet the requirements of most big data modeling and analysis scenarios.**

MaxCompute source and target tables

**Tables are data storage objects in MaxCompute. Similar to relational database tables, tables in MaxCompute have a two-dimensional logical structure. A source table is the input of an algorithm node, while a target table is the output of an algorithm node.**

## 8.6 FAQ

**This topic describes the problems that you may encounter when using Apsara Stack Machine Learning Platform for AI and the corresponding solutions.**

How do I log on to the Apsara Stack Machine Learning Platform for AI console?

**See [Log in to machine learning console](#).**

How do I create an experiment?

**You can use one of the following methods to create an experiment:**

- **Go to the Apsara Stack Machine Learning Platform for AI homepage and choose New > New Experiment.**
- **Go to the Experiments pane. Right-click My Experiments and choose New Experiment from the shortcut menu.**
- **Go to the Experiments pane, and click New Experiment.**

How do I prepare data?

- 1. Switch to the Components pane. Click Data Source/Target, and drag and drop the Read MaxCompute Table component onto the canvas. Click this component and select your MaxCompute table in the configuration pane on the right side of the page.**
- 2. On the Column Information tab, you can view the data type of the input table and the values in the first 100 data entries.**

How do I preprocess data?

- 1. Search for the Missing Value Imputation component in the search box, and drag and drop this component onto the canvas.**

2. Click the Missing Value Imputation component and set its parameters. Click Columns and select the columns to be imputed. Impute the columns with the minimum values.

Missing value imputation is an important step in data preprocessing before model training.

3. Click Data Preprocessing, and then drag and drop the Split component onto the canvas and set its parameters.

The purpose of this step is to split data in half into model training data and model prediction data.

How do I visualize data?

1. Click Statistical Analysis, and then drag and drop the Whole Table Statistics component onto the canvas. Connect the components, and then click Run at the top of the canvas.
2. After the experiment stops running, right-click Whole Table Statistics and choose View Data from the shortcut menu. The statistics about the whole table are displayed, as shown in the following .

How do I create a model?

1. Choose Machine Learning > Binary Classification. Drag and drop the Logistic Regression for Binary Classification component onto the canvas, connect the components, and start the experiment.
2. Click the Logistic Regression for Binary Classification component. On the Column Settings tab on the right, set the Training Feature Columns and Target Columns parameters.
3. On the Parameter Settings tab on the right, set the Regularization Type, Maximum Iterations, Regularization Coefficient, and Minimum Convergence Deviance parameters.
4. Click Models in the left-side navigation pane and view the model generated by the experiment under Experiment Model.

## 9 Dataphin

---

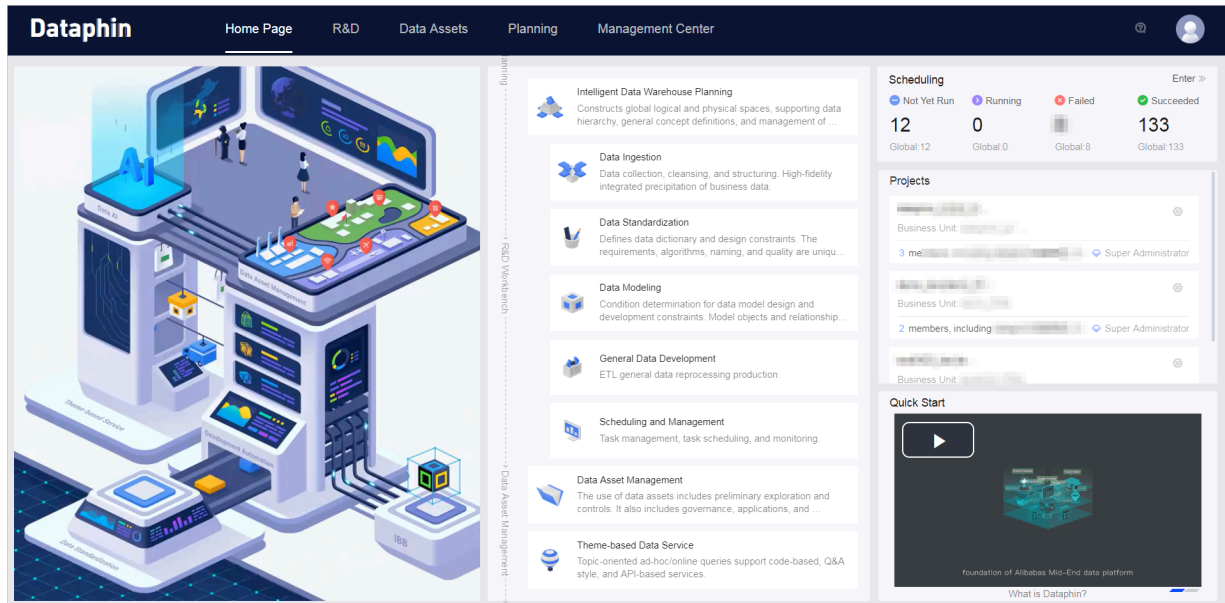
### 9.1 What is Dataphin?

Dataphin is an engine for creating intelligent big data platforms. It is designed to meet the requirements of big data development, management, and utilization across multiple industries. It adopts an OneData, OneEntity, OneService (product, technology, methodology) big data lifecycle management system. The system is developed by Alibaba Cloud and has been proven by years of practice. Dataphin provides an end-to-end intelligent data creation and management solution covering data ingestion, data standardization, data modeling, data development, data distilling, data asset management, and data services. These features help governments and enterprises build an asset-oriented, service-oriented, closed-loop, and self-optimizing intelligent data system with unified standards to stimulate and drive innovation.

Dataphin is integrated with a large amount of compute and storage environments, which enables you to use a single console to process data from various data sources . By using Dataphin, you can import data, produce standard data by data modeling, and create a tag system by extracting tags from entities. This allows you to generate and manage data assets by using your business data knowledge. Dataphin also provides several types of data services including data table search and intelligent voice search.

The following figure shows the Dataphin R&D Workbench.





## 9.2 Limits

To use Dataphin, you must have the necessary knowledge and expertise. This document is intended for:

- Application developers
- Analysts
- Data developers
- Technical architects
- System administrators

To ensure stable running of Dataphin, observe the following limits and recommendations.

Operation	Limit/Recommendation
Computing engine type: select a computing engine type	Select a computing engine type and configure the cluster where your computing engine is located. For example, you need to specify the endpoint of the cluster. MaxCompute and Hadoop are the available computing engine types. The system uses the computing engine to support data construction. Select the computing engine type and configure the computing engine settings based on your computing engine cluster.

Operation	Limit/Recommendation
Data source management: add data sources	<ul style="list-style-type: none"> <li>• We recommend that you set an AccessKey with administrative privileges for data source management.</li> <li>• We do not recommend that you configure one physical database as two data sources. Two data sources cannot have the same configuration.</li> </ul>
Project management: configure project names	<ul style="list-style-type: none"> <li>• When the data source type is MaxCompute, the project name must be the same as your MaxCompute project name.</li> <li>• The project name must not start with LD_ or ld_. If project names start with LD_ or ld_, project names conflict with business unit names. In this case, errors may occur during table query because the system cannot differentiate between a logical table and physical table. In Dataphin, when querying a logical table, you must prefix the table name with the corresponding business unit name. When querying a physical table, you must prefix the table name with the corresponding project name.</li> </ul>
Project management: configure computing engine sources	<ul style="list-style-type: none"> <li>• If a physical database has been configured as a data source, we do not recommend that you add, delete, or modify data in the database from non-Dataphin consoles.</li> <li>• We do not recommend that you use computing engine sources from different clusters.</li> </ul>
R&D Workbench: process data	Computing engine sources cannot read data from different clusters.

Operation	Limit/Recommendation
R&D Workbench: standard modeling	<ul style="list-style-type: none"><li>• We recommend that you use caution when naming data standardization elements and logical tables. Set lowercase names for data standardization elements to ensure easy reading. Make sure that names are valid and easy to read. The names cannot be changed when they have downstream dependencies.</li><li>• Use abbreviations whenever possible to avoid data production errors. Errors may occur because the field name length exceeds the limit imposed by the database.</li></ul>
R&D Workbench: ad hoc query	When querying a logical table, prefix the table name with the corresponding business unit name. When querying a physical table, prefix the table name with the corresponding project name.
Data distilling (coming soon): ID Engine	We recommend that you set IDs based on user information to ensure an accurate match between an ID and user.

## 9.3 Quick start

### 9.3.1 Instructions for the system administrator

This topic is intended for the system administrator. Before using Dataphin, the system administrator and the deployment engineers must ensure that the environment is ready and required user roles are created.

#### Procedure

## 1. Make sure that the hardware environments are ready:

- **Apsara Infrastructure Management Framework is deployed and DTCenter is accessible.**
- **MaxCompute, OSS, DAuth, SLB, ECS, three physical machines, and PostgreSQL database resources are available.**



### Note:

**The PostgreSQL database in your environment must be accessible.**

**PostgreSQL can be installed on an RDS instance or ECS instance:**

- **PostgreSQL on RDS: A 6U model is required for Apsara Stack 3.7.0 and earlier versions. For versions later than 3.7.0, a 7U model is required. We do not recommend that you use miniRDS because it soon to become obsolete.**
- **PostgreSQL on ECS: Dataphin provides this configuration in Apsara Stack 3.7.0 and later versions. You can obtain this baseline configuration if you purchase any of these versions. If you purchase a version earlier than 3.7.0, we recommend that you contact the Dataphin team to obtain specific ECS resources and deploy PostgreSQL in a non-standard way.**

## 2. Obtain the computing cluster information.

- **Endpoint of MaxCompute: This information is required when you configure the computing engine.**
- **AccessKey ID and AccessKey Secret generated while project [Dataphin\_Meta] is initialized in MaxCompute: This MaxCompute project is used for MaxCompute metadata computation and storage.**



### Note:

**After the Apsara Stack environment and MaxCompute are deployed, a project is generated to obtain the MaxCompute metadata in the Apsara Stack environment. The system administrator and deployment engineers must verify that project [Dataphin\_Meta] exists and obtain the AccessKey ID and AccessKey Secret. If project [Dataphin\_Meta] does not exist, contact the deployment engineers to manually create project [Dataphin\_Meta] and grant Dataphin the permission to obtain MaxCompute metadata.**

### 3. Generate accounts.

When the deployment is complete, the account system provides the following three types of user roles in the Apsara Stack environment:

- **O&M super administrator:** an independent metadata management tenant of the Dataphin system. The O&M super administrator can obtain and parse the metadata of the customer's cluster. Each system has only one O&M super administrator. Contact the deployment engineers to obtain the account and password of the O&M super administrator.



**Note:**

The system administrator must keep the account and password of the O&M super administrator strictly confidential, and use caution when performing operations after you log on as the O&M super administrator.

- **Super administrator:** a tenant of the customer, who performs development and construction operations. For example, a super administrator can manage users, design and build the data architecture of a specific business data system. In the Apsara Stack environment, the role of a department account (or Apsara Stack tenant account) is a super administrator.
- **Common user:** a member of a department. A common user performs building and development operations, including the detailed design of a specific business data system. In the Apsara Stack environment, a RAM user can be added under a department account. That is, a RAM user can be added as a tenant member of the super administrator (department account).



**Note:**

Currently, each user can only belong to one tenant, and a user from one department cannot be added as a tenant member of another department. We recommend that each Dataphin system can only be used for one department (one tenant).

### 9.3.2 Instructions for quick start

Before using Dataphin, you need to apply for an account from the system administrator. After logging on to the system, you can start using Dataphin by following the process described in this topic.

1. **Data ingestion: The Management Center and data warehouse planning modules** allow you to ingest the source data. Your physical data sources serve as the foundation for building the data platform. For more information, see [Management Center](#) and [Data warehouse planning](#).
2. **Data modeling and data distilling: During data modeling and data distilling,** you need to standardize data and extract data to build a data model. For more information, see [Data modeling and development](#).
3. **Asset management and data service: The data asset and theme-based data service modules** allow you to inventory and assess the data assets across your enterprise and run ad hoc queries. For more information, see [Data assets](#) and [Theme-based data service](#).

You can control the entire process by using the scheduling center. For more information, see [Scheduling center](#).

### 9.3.3 Log on to the Dataphin console

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

#### Context

After you log on to the Dataphin console, you can use all modules of Dataphin.

#### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.

**3. Enter the correct username and password.**

- The system has a default super administrator with the username super. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
- You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).

**4. Click LOGIN to go to the Dashboard page.**

### 9.3.4 Management Center

You can manage member accounts and configure system settings in Management Center.

Choose Management Center > Members Management. On the page that appears, you can add members, delete members, synchronize account systems, and search for members.

Choose Management Center > Computing Engine Computation. On the page that appears, you can configure the computing engine, cluster, and metadata. You can also verify the specified cluster endpoint. For more information, see [Management Center](#).

### 9.3.5 Data warehouse planning

Data warehouse planning is a key step in architectural design (during data production).

To complete this step, you need to define logical space divisions based on your business characteristics. The logical space divisions include business units, data domains, namespaces, and global objects. Then, create projects based on your development cooperation and management mode. When creating a project, configure project and member settings, and register the required underlying data sources (physical databases). For more information, see [Data warehouse planning](#).

### 9.3.6 Data ingestion

This module selects the required business data for storage based on the source data layer design in the global architecture of enterprise data. In addition, data ingestion module also formulates data synchronization, cleansing, and structuring policies based on requirements for data storage, timeliness, and quality.

As an initial stage in data construction, the data synchronization suite is developed based on Alibaba's years of practice in the synchronization and exchange of business data, log data, and other types of data. This achieves high efficiency of raw business data ingestion. Through the pipeline, it can support metadata transmission, acquisition and statistics, as well as simple rule checking and custom fault-tolerant mechanism for data transmission volume and content. This can achieve flexible management and high-quality of data synchronization.

#### Data source configuration

This module supports data source access and management. The data source list allows you to manage accessed data sources conveniently and add data sources of various types. Currently, the data synchronization center supports data sources including MaxCompute, MySQL, SQL Server, PostgreSQL, and Hive.

#### Data sync

This module allows you to select source data and target data, configure parameters for incremental or full data synchronization, identify the mappings between source data fields and target data fields, configure transmission traffic and concurrent transmissions, and schedule task nodes after creation.

### 9.3.7 Data modeling and development

Dataphin offers systematic modeling and development to implement and help advance the data warehouse theory.

You can define dimensions and business processes by using a top-down approach. Then, you can further develop dimension tables, fact tables, aggregate tables, the application data store layer, and the common dimensional model layer. This facilitates the use of layered business data and optimizes computing and storage. For more information, see [Data modeling and development](#).



### 9.3.8 Scheduling center

The scheduling center manages the running of all tasks.

Rule-based and ordered publishing of code relies on a publishing and scheduling system. The scheduling center manages the running of all tasks. It provides directed acyclic graphs (DAGs) to show the dependencies between scheduled tasks. The DAGs can help you learn and adjust task running progress to improve the stability of data production. After you build a data model or configure data distilling rules, the coding module automatically controls the scheduling system.

Based on the computation of global metadata, the coding module automatically generates optimal code to reduce costs. You only need to design the computing logic. You can ignore storage and computation issues. In the scheduling center, you can view the code tasks configured by yourself and tasks created by the system. For more information, see [Tasks](#).

### 9.3.9 Data assets

Dataphin allows you to inventory and assess the data assets across your enterprise based on the standards and methodology of enterprise data asset management.

You can manage all data and APIs created by data modeling, development, and distilling. Dataphin supports searching by keywords and provides a data catalog. This allows you to spend less time searching for data. The data catalog helps enterprise executives discover and understand the value of data assets. It also supports automatic metadata extraction and analysis. Dataphin allows you to inventory and analyze data assets throughout the entire data creation process covering computation, storage, security, and application. When problems are found, Dataphin provides several intelligent solutions to optimize data governance. This helps enterprises reduce costs and improve data analysis efficiency.

Dataphin can provide a visual display of created data and can also display the data in detailed data tables. For more information, see [Global mode](#).

### 9.3.10 Theme-based data services

Dataphin provides theme-based data services to support the ad hoc query feature so that you can query data during development or query data in your data sources.

When writing query code in the code editor, you only need to enter a logical table name and one or more characters of a field name. The code editor can display

the fields of logical tables associated with your specified logical table in the star schema or snowflake schema. You can also use SQL statements to obtain all data fields in a model when fields are not duplicate.

For more information, see [Ad hoc query](#).

## 9.4 Management Center

### 9.4.1 Overview

On the Management Center page, the system administrator can manage member accounts and configure system settings. These operations must be complete before Dataphin is used. This topic describes how to use the O&M super administrator and super administrator accounts to configure the metadata warehouse, set the computing engine type, and manage members.

For more information about the O&M super administrator and super administrator, see [Instructions for the system administrator](#).

### 9.4.2 Initialize metadata

Once you have obtained the computing cluster information, you can log on to the Dataphin system as the operations and maintenance (O&M) super administrator to configure the metadata warehouse. This topic uses Apsara Stack Dataphin (MaxCompute) as an example.

#### Procedure

1. [Log on to the Dataphin console](#) as the O&M super administrator. For more information about the O&M super administrator, see [Instructions for the system administrator](#).

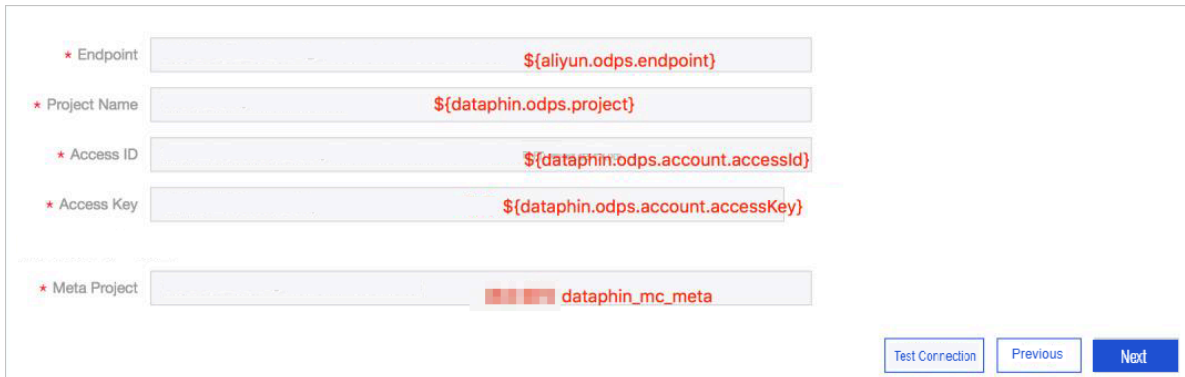


#### Note:

The system administrator must keep the account and password of the O&M super administrator strictly confidential, and use caution when performing operations after you log on as the O&M super administrator.

2. Choose Management Center > Metadata Warehouses Configuration and click Start.
3. Select a computing engine and click Next. In this example, select MaxCompute.

4. On the Parameter Configuration page, follow the instructions in the following figure to configure the parameters. Click Test Connection. After the test is passed, click Next.



The screenshot shows a 'Parameter Configuration' form with the following fields and values:

- Endpoint: `#{alibabacloud.endpoint}`
- Project Name: `#{dataphin.odps.project}`
- Access ID: `#{dataphin.odps.account.accessId}`
- Access Key: `#{dataphin.odps.account.accessKey}`
- Meta Project: `dataphin_mc_meta`

Buttons at the bottom right: Test Connection, Previous, Next.

5. Perform the subsequent steps as prompted.

**Note:**

The subsequent steps, including configuring MaxCompute at the backend, take about 15 minutes. If the system displays information indicating that the subsequent steps are successfully performed, click Complete.

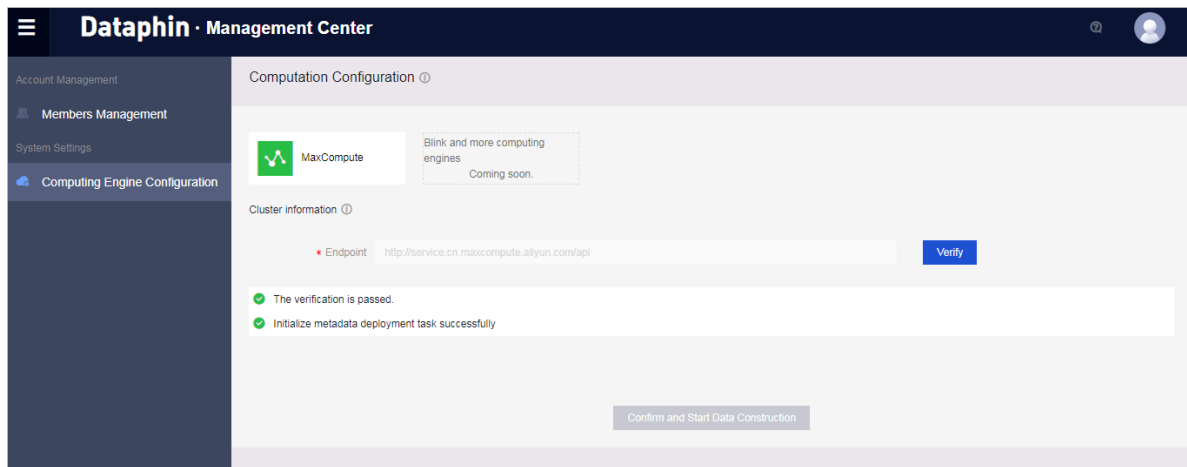
### 9.4.3 Set the computing engine type

After the metadata is initialized, you can log on to the Dataphin system as the super administrator to set the computing engine type. This topic uses MaxCompute as an example.

#### Procedure

1. [Log on to the Dataphin console](#) as the super administrator. For more information about the super administrator, see [Instructions for the system administrator](#).

2. Choose Management Center > Computing Engine Configuration. After your specified endpoint is verified, click Confirm and Start Data Construction, as shown in the following figure.



After setting the computing engine type, you can proceed to create a computing engine source as the super administrator. For more information, see [Computing engine sources](#).

#### 9.4.4 Member management

The super administrator can manage the members in the system and control their access permissions. On the Members Management page, you can view each member's account name, nickname, and time that each member is added to the Dataphin system. Only the super administrator can add and delete members. The synchronization between account systems is available to all users. This supports synchronous and asynchronous update of account information. In Dataphin, the member session timeout is set to 6 hours.

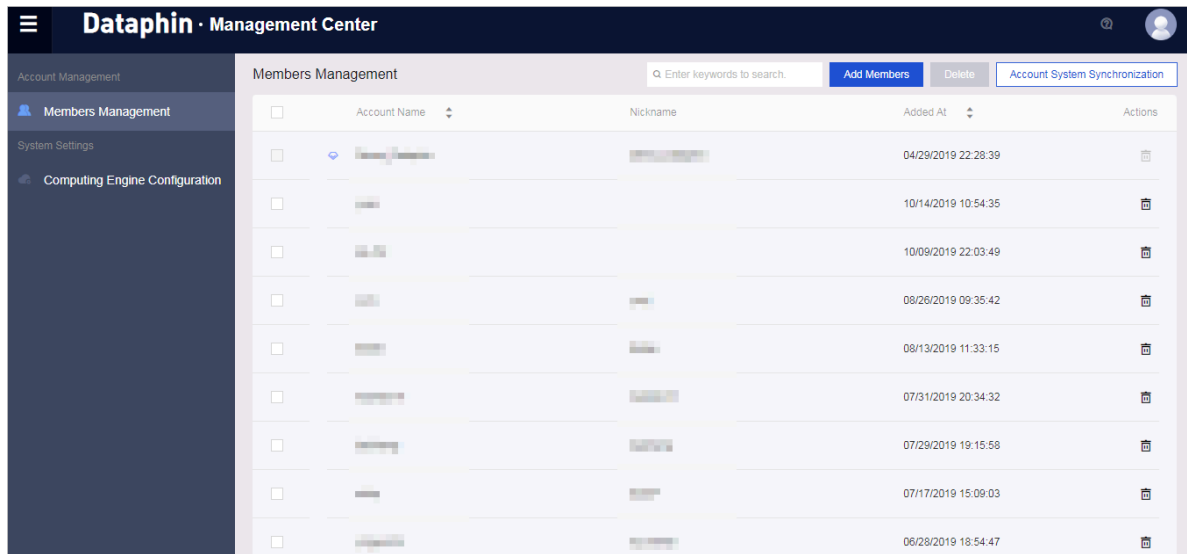
##### Context

You can perform the preceding operation provided that Dataphin is connected to the account system of the environment where you deploy Dataphin.

##### Procedure

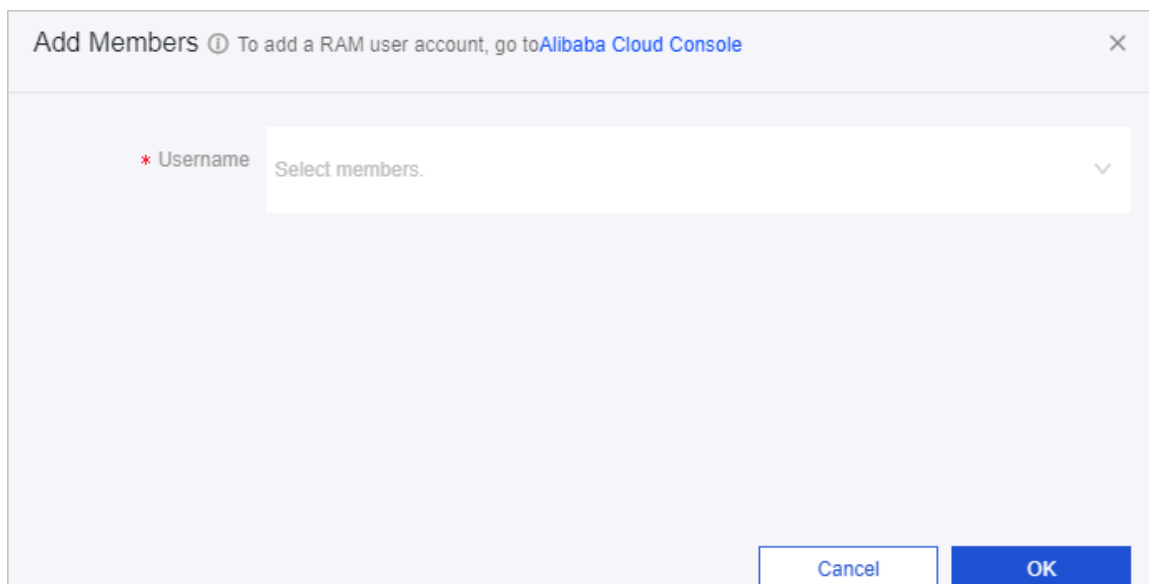
1. [Log on to the Dataphin console](#) as the super administrator. For more information about the super administrator, see [Instructions for the system administrator](#).
2. Choose Management Center > Members Management. On the page that is displayed, you can view the account name, nickname, and the time that each member is added to the Dataphin system. You can sort the members in ascending or descending alphabetical order of account names. You can also sort the

members in ascending or descending order of the time that each member is added to the system.



- **Add a member**

Click **Add Members**. In the dialog box that appears, select a member name and click **OK**, as shown in the following figure.

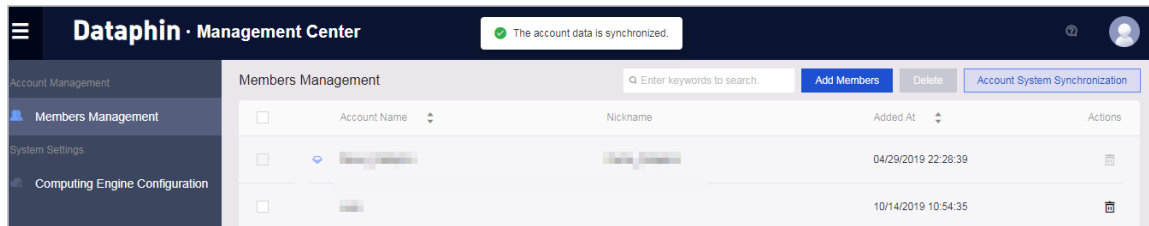


- **Delete members**

Select the members to be deleted and click **Delete > OK**.

- **Synchronize account systems**

Click **Account System Synchronization** to update account information, as shown in the following figure.



## 9.5 Data warehouse planning

### 9.5.1 Overview

Data warehouse planning is a key step in architectural design (during data production). To complete this step, you need to define logical space divisions based on your business characteristics. The logical space divisions include business units, data domains, namespaces, and global objects. Then, create projects based on your development cooperation and management mode. When creating a project, configure project and member settings, and register the required underlying data sources (physical databases).

### 9.5.2 Business units

A business unit defines the namespace of a data warehouse in Dataphin. When your business data has different definitions, you can create business units to separately manage each type of business data. Then, you need to build the data warehouse based on the created business units. A business unit can contain multiple projects.

#### Prerequisites

Assume that your business focuses on retail. Various business systems are less isolated. In this scenario, only one business unit is required.

#### Procedure

1. Log on to the [Log on to the Dataphin console](#) as the super administrator.



#### Note:

Only the super administrator can create and modify business units.

2. On the Dataphin homepage, click Planning in the top navigation bar or Intelligent Data Warehouse Planning in the middle section to go to the Planning page.
3. On the Planning page that appears, click Business Units in the left-side navigation pane. On the Business Units page, click + Create Business Unit.

4. In the Create Business Unit dialog box that appears, select Dev-Prod Mode or Basic Mode as required and click Next. Then, enter the common name, common display name, and description, select an icon, and click OK.

**Note:**

By default, business unit names are prefixed with LD\_. Therefore, project names cannot have this prefix. This enables Dataphin to distinguish between logical tables and physical tables in data queries. When logical tables and physical tables are referenced, the former is prefixed with the corresponding business unit name, and the latter is prefixed with the corresponding project name. If project names also start with LD\_, Dataphin cannot distinguish between physical tables and logical tables that are both prefixed with LD\_.

5. On the Business Units page, click the business unit that you created in preceding steps, for example, the business unit named LD\_retail. The Data Domain section in the lower part of the page lists all data domains in this business unit. Then, click + Create Data Domain.
6. In the Create Data Domain dialog box that appears, enter the data domain display name, data domain name, alias, and description. Then, click OK.

**Note:**

A data domain is used to categorize business concepts in a business unit. For example, you can create the commodity domain, transaction domain, and membership domain to store different types of business concepts.

7. In the Business Unit Parameters section of the target business unit, view the time-based partition field parameters of the business unit.
8. Click the Change icon for Time-based Partitioning Field. In the dialog box that appears, modify the display name, name, data type, default value, or description.

**Note:**

- When you create a business unit, Dataphin sets parameters for Time-based Partitioning Field by default. You can modify these default parameters as required, such as the name and data type.
- However, if logical tables are created in the business unit, you cannot modify the name and data type of the time-based partition field.

### 9.5.3 Global objects

Global objects are concepts that can be universally referenced. By defining global objects, you can ensure that the definitions of the concepts are consistent throughout the entire system.

#### Procedure

1. *Log on to the Dataphin console.* Choose **Intelligent Data Warehouse Planning > Global Objects**.
2. Click **Create Statistical Period**.
3. Specify the start time, end time, name, and expression of the statistical period, as shown in *Figure 9-1: Statistical period*.

The statistical periods, period names, and period expressions of some frequently used metrics have been initialized.

Figure 9-1: Statistical period

Create Statistical Period

\* Statistical Period Enter a display name. A display name must be 1 to 64 characters in length, for example, last 7 days. It can contain letters, r

\* Alias Enter a statistical period alias. An alias must be 1 to 10 characters in length, such as 7d, and can contain letters, numbers, ,

Description Enter a statistical period description. The description must be 0 to 128 characters in length. 0/128

Expression Expression Parameter Description

Start Time ☐ Parameter Enter a parameter. ☒ Function Expression lastNDate \${bizdate}', 7

End Time ☐ Parameter Enter a parameter. ☒ Function Expression lastNDate \${bizdate}', 7

Cancel OK

### 9.5.4 Project management

A project is used to isolate physical resources and developers during Data Mid-End construction. A business unit can contain multiple projects. Each Dataphin



member can join multiple projects. By creating projects, you can isolate physical resources and group developers.

### Prerequisites

- You must configure the computing engine in the Management Center before managing projects. For more information, see [Set the computing engine type](#).
- Only the super administrator can create projects.
- When you create a project, you need to bind a computing engine to the project. Therefore, you need to create a computing engine before creating a project. For more information, see [Computing engine sources](#).
- A computing engine can be bound to only one project. A project can have only one computing engine. You cannot change the computing engine of a project after they are bound.

### Procedure

1. Log on to the [Log on to the Dataphin console](#) as the super administrator.
2. On the Dataphin homepage, click Planning in the top navigation bar or Intelligent Data Warehouse Planning in the middle section to go to the Planning page.
3. On the Planning page that appears, click Project Management in the left-side navigation pane. On the Project Management page, click + Create Project in the upper-right corner.
4. In the Create Project dialog box that appears, set parameters as required.
  - a) Configure the basic information about the project, including the computing engine, common display name, common name, and description.



#### Note:

- If the computing engine type is MaxCompute, we recommend that you set Common Name to the MaxCompute project name.
- By default, business unit names are prefixed with LD\_. Therefore, project names cannot have this prefix. This enables Dataphin to distinguish between logical tables and physical tables in data queries. When logical tables and physical tables are referenced, the former is prefixed with the corresponding business unit name, and the latter is prefixed with the corresponding project name. If project names also start with LD\_,

Dataphin cannot distinguish between physical tables and logical tables that are both prefixed with LD\_.

- After selecting MaxCompute for Computing Engine, you need to use the authorization code copied from the Computing Engines page to grant required permissions to your AccessKey by following the authorization mechanism of MaxCompute. This guarantees that you can pass the authentication when you access physical databases across projects in Dataphin.

b) Configure the namespace of the project. The business unit is optional. If you do not select a business unit, you cannot proceed with data standardization or data modeling for logical tables. Instead, you can only write scripts to develop data. The project type is used to categorize the tasks and generated data of projects. The default project type is Application Data Store.



**Note:**

Valid values of the Project Type parameter are described as follows:

- **Source Data:** stores raw data of business databases. This layer serves as the source and basis in follow-up data development and is also called the vertical data center.
- **Common Dimensional Modeling:** extracts themes, standards, and common data from business data. This layer connects the source data layer and application data store layer and is also called the common data center.
- **Application Data Store:** defines diversified and distinct metrics based on business scenarios.

5. After you complete the preceding configuration, click OK to create the project. You can find this project on the Projects Joined tab of the Project Management page.
6. On the Project Management page, click Members Management for the project that you created in preceding steps.

7. In the Manage Members dialog box that appears, add or remove project members as required.

- a) To add a member, click + Add Members. In the dialog box that appears, select the target member and specify the role for the member.
- b) To remove a member, click the Delete icon in the Actions column for the target member.



**Note:**

- The project administrator can modify the role for members.
- By default, the super administrator also plays the administrator role in all projects. You cannot delete or change the super administrator role.

8. On the Project Management page, click Enter Workbench for the target project. On the Develop tab that appears, you can develop data in the project.

### 9.5.5 Physical data sources

You can register your physical databases to Dataphin. Physical databases serve as the underlying sources of data for projects and data synchronization. The physical data sources supported by Dataphin include MaxCompute, MySQL, DRDS, SQL Server, PostgreSQL, and Oracle.

#### Procedure

1. [Log on to the Dataphin console](#). Choose Intelligent Data Warehouse Planning > Physical Data Sources.
2. Click Create Data Source.

3. Select MaxCompute as the data source type, as shown in [Figure 9-2: Data source type](#).

Figure 9-2: Data source type

The screenshot shows a 'Create Data Source (Computing Engine)' dialog box. It contains the following fields and controls:

- Type:** A dropdown menu with 'MaxCompute' selected.
- Name:** A text input field with the placeholder 'Enter a data source name.'
- Description:** A text input field with the placeholder 'Enter a data source description.' and a character count '0/128'.
- Endpoint:** A text input field with the value 'http://service.cn.maxcompute.aliyun.com/api' and an information icon.
- Project Name:** A text input field with the placeholder 'Enter a project name.'
- Access ID:** A text input field with the placeholder 'Enter an Access ID for authentication. To ensure that the task is executed normally, make sure that you have the required data permission.'
- Access Key:** A text input field with the placeholder 'Enter an AccessKey for authentication.' and an information icon.

At the bottom right, there are three buttons: 'Test Connection' (disabled), 'Cancel', and 'Submit'.

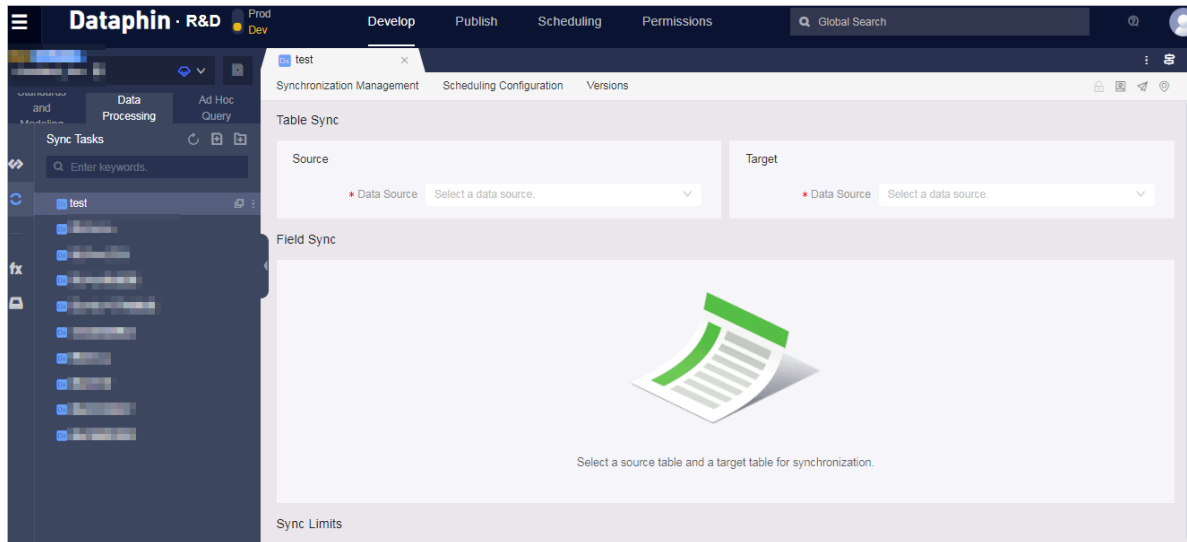
4. Enter a data source name and description. You can use the name of your Dataphin project as the data source name.

5. To add another type of data source, repeat the preceding procedure.

After you have designed the data architecture and determine the data required, you can start ingesting data. Specifically, you can use Dataphin to complete data collection, cleansing, conversion to structured data, integration, and

**synchronization. On the R&D workbench, you can synchronize data between databases that are used as data sources, as shown in [Figure 9-3: Data synchronization](#).**

Figure 9-3: Data synchronization



**Now, you have finished global data planning, which is the first step for data development.**

### 9.5.6 Computing engine sources

**Currently, data from heterogeneous databases cannot be computed together. Therefore, you must specify a computing engine type for the entire system to ensure the proper transfer of source data and normal use of Dataphin functionalities.**

#### Context



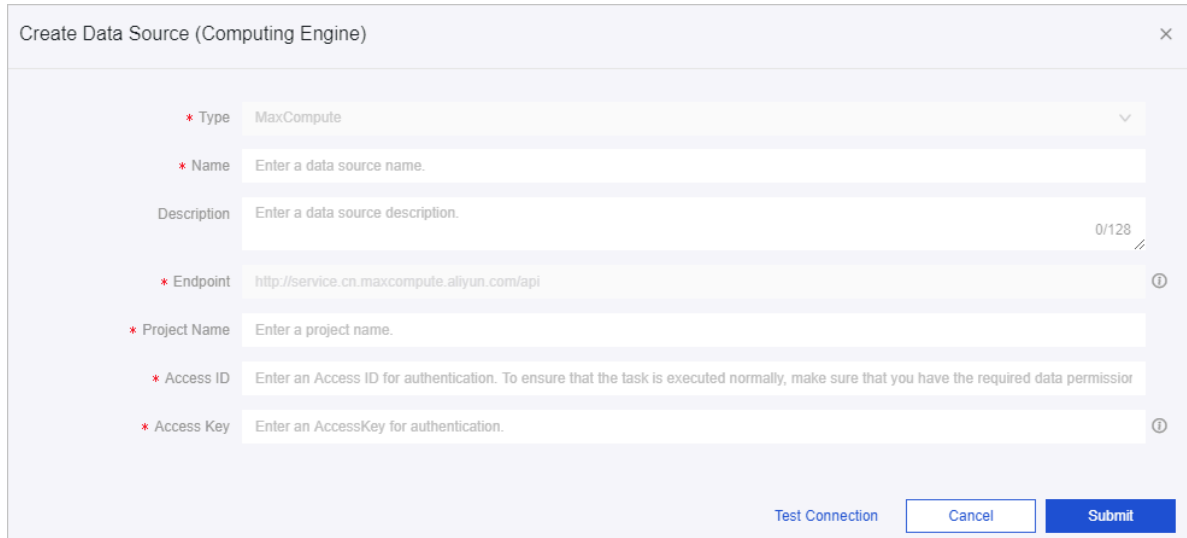
#### Note:

**The roles in Dataphin can be classified into super administrator and developer. The super administrator is responsible for configuring basic information.**

#### Procedure

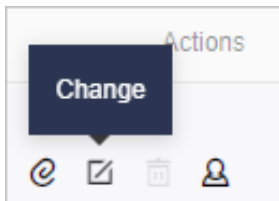
1. [Log on to the Dataphin console](#) as the super administrator.
2. On the homepage, click Intelligent Data Warehouse Planning or click Planning in the top navigation bar to go to the data warehouse planning page.

3. Click **Computing Engine Sources** in the left-side navigation pane. On the page that appears, click **Create Data Source**. The data source creation dialog box appears, as shown in the following figure.



The image shows a 'Create Data Source (Computing Engine)' dialog box. It contains several input fields with red asterisks indicating required fields: 'Type' (set to 'MaxCompute'), 'Name' (placeholder: 'Enter a data source name.'), 'Description' (placeholder: 'Enter a data source description.', with a character count '0/128'), 'Endpoint' (placeholder: 'http://service.cn.maxcompute.aliyun.com/api'), 'Project Name' (placeholder: 'Enter a project name.'), 'Access ID' (placeholder: 'Enter an Access ID for authentication. To ensure that the task is executed normally, make sure that you have the required data permission'), and 'Access Key' (placeholder: 'Enter an AccessKey for authentication.'). At the bottom right, there are three buttons: 'Test Connection', 'Cancel', and 'Submit'.

4. In the Create Data Source dialog box, configure the required parameters and click **Test Connection**. After the connection test is passed, click **Submit**.
5. After the data source is created, you can click **Change** to modify the data source.



## 9.6 Data ingestion

Data ingestion is achieved through data synchronization. Data synchronization is the process of importing data from a table in the source database to a table in the target database. For example, importing data of table A in a MySQL database to table B in a PostgreSQL database.

1. Go to the R&D workbench. [Log on to the Dataphin console](#). Click **R&D** in the top navigation bar or **Data Ingestion** on the homepage.
2. Go to the data ingestion page. On the R&D page, click **Data Processing**, and then click the **Sync Tasks** submenu. You can move the pointer over an icon to expand its corresponding submenu. On the **Sync Tasks** page, you can create folders, and create, save, and publish sync tasks.

## 9.6.1 Manage sync task folders

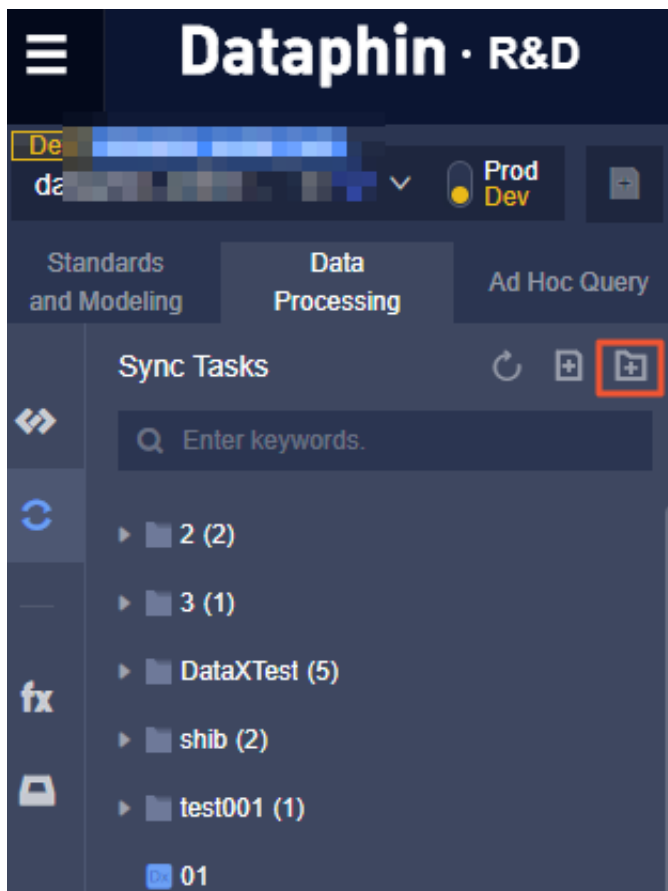
This topic describes how to create, move, rename, and delete folders and subfolders.

### Context

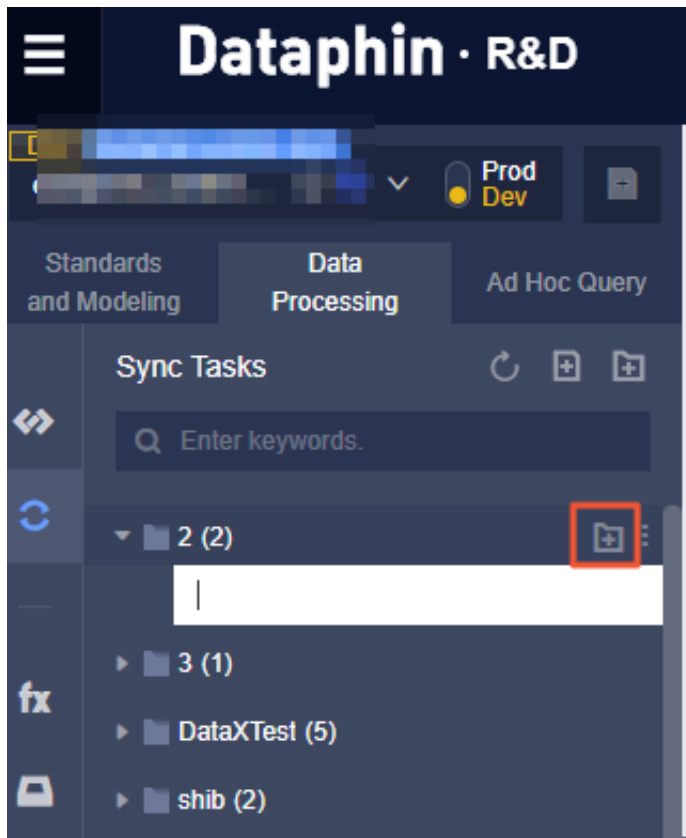
Data ingestion is achieved through data synchronization. Data synchronization is the process of importing data from a table in the source database to a table in the destination database. To categorize and manage sync tasks, you need to create, move, rename, and delete folders and subfolders as required.

### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.
3. On the Develop tab of the R&D page, click the Data Processing tab, and then click Sync Tasks on the left-side navigation menu. In the left-side navigation pane, click the Create Folder icon, enter the folder name as required, and then press Enter.

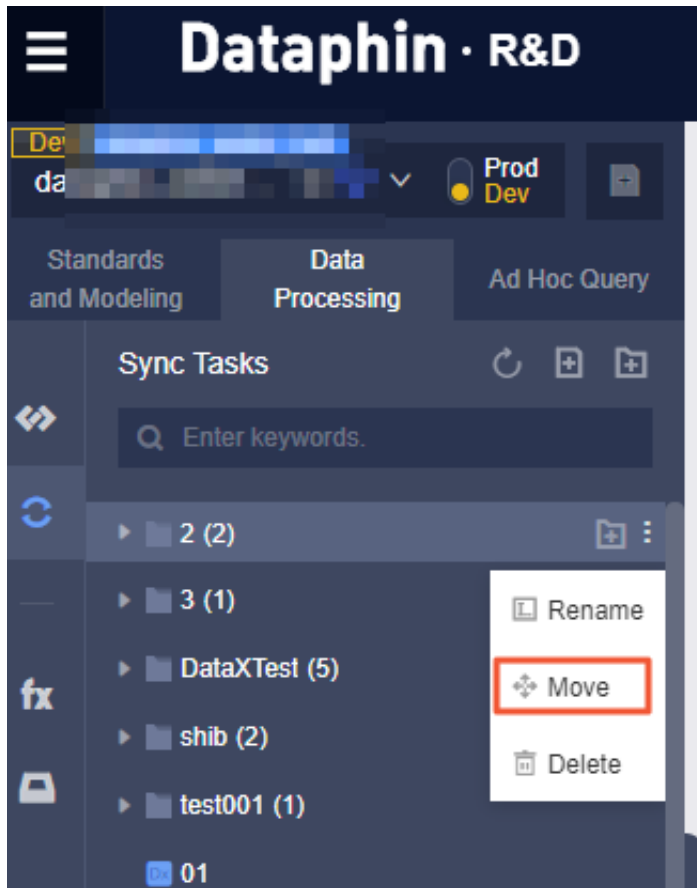


4. Move the pointer over the folder that you created in preceding steps, click the Create Folder icon, enter the subfolder name as required, and then press Enter.

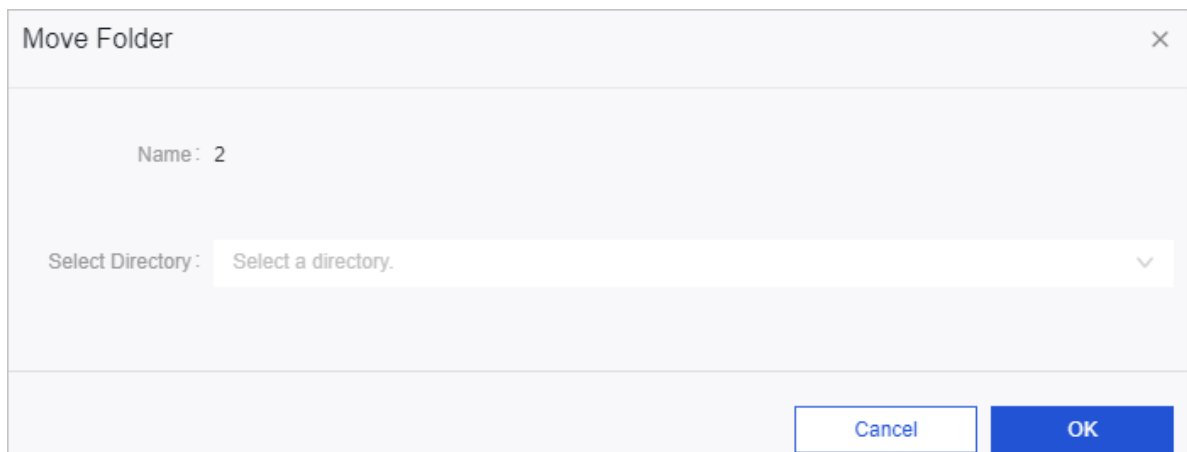





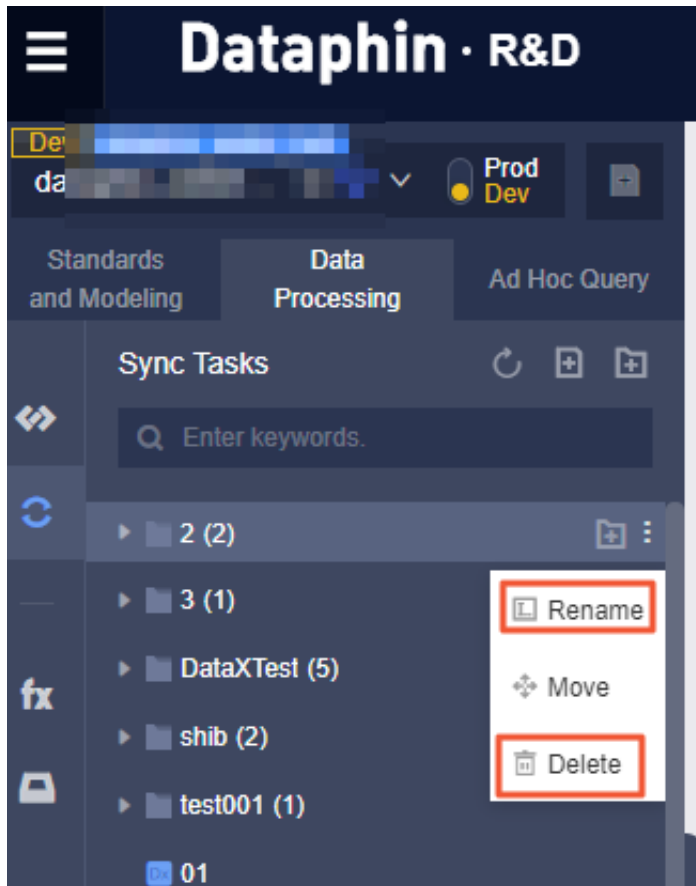
5. To move a folder, move the pointer over the  icon next to the target folder and select Move.



6. In the Move File dialog box that appears, select the destination directory from the drop-down list and click OK.



7. To rename or delete a folder, move the pointer over the  icon next to the target folder and select Rename or Delete.

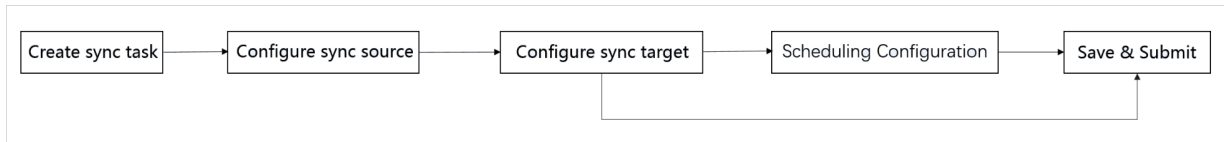
**Note:**

You cannot delete a folder that contains items.

## 9.6.2 Manage sync tasks

A sync task imports data from a source table into the target table. The process of creating a sync task is as follows:

1. Create a sync task.
2. Configure the information about the source table and target table for the sync task.
3. Save or publish the task.
  - A one-time task can be directly saved or published.
  - A recurring task can be saved or published only after you have configured the scheduling information.

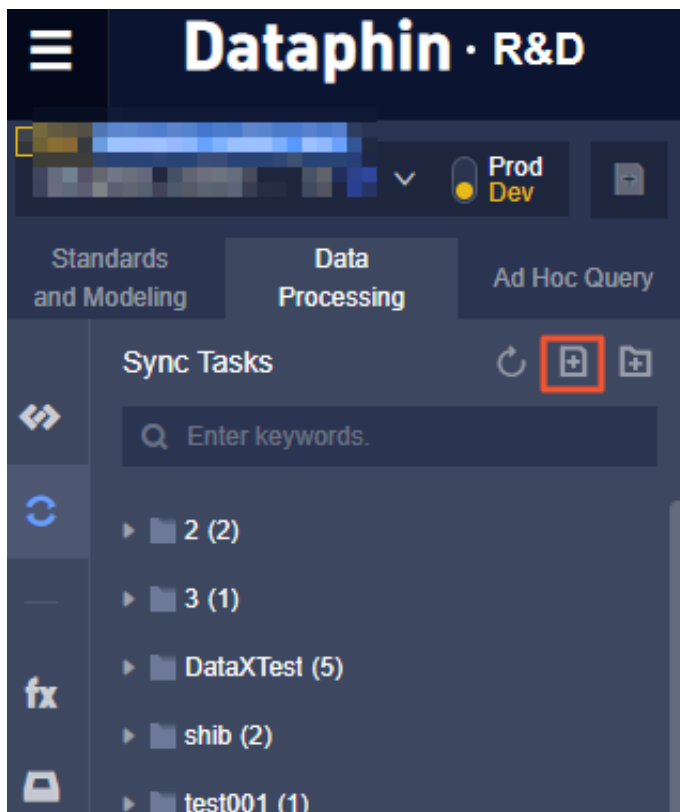


### 9.6.3 Create a sync task

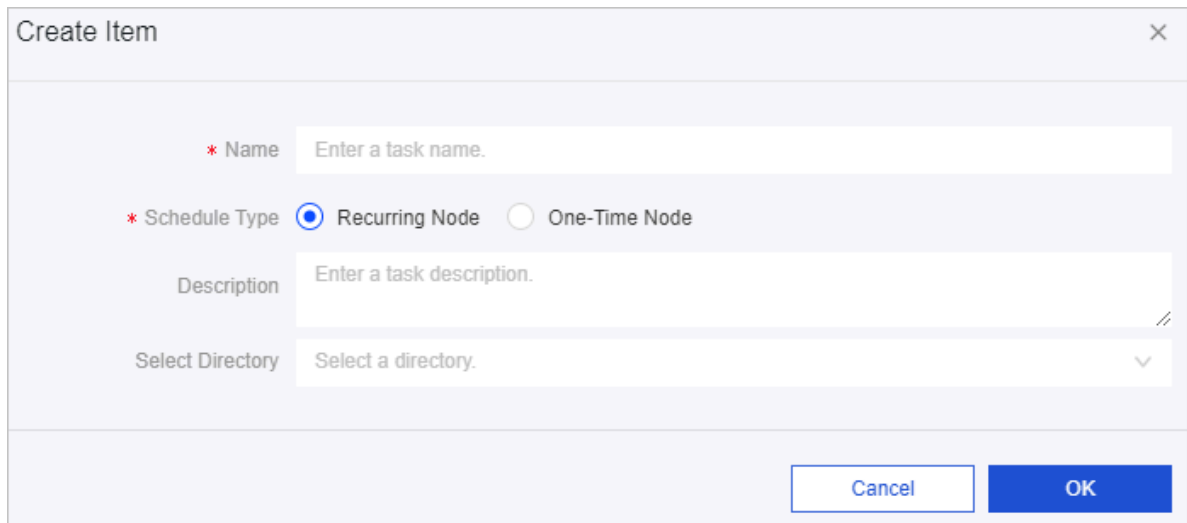
This topic describes how to create a sync task.

#### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.
3. On the Develop tab of the R&D page, click the Data Processing tab, and then click Sync Tasks on the left-side navigation menu.
4. In the left-side navigation pane, click the Create File icon.



5. In the Create File dialog box that appears, enter the name and description, select the scheduling type and directory, and then click OK.



The 'Create Item' dialog box contains the following fields and controls:

- Name:** A text input field with the placeholder 'Enter a task name.' and a red asterisk indicating it is required.
- Schedule Type:** Two radio buttons: 'Recurring Node' (selected) and 'One-Time Node'.
- Description:** A text input field with the placeholder 'Enter a task description.' and a red asterisk indicating it is required.
- Select Directory:** A dropdown menu with the placeholder 'Select a directory.'
- Buttons:** 'Cancel' and 'OK' buttons at the bottom right.

**Note:**

You can set Schedule Type to Recurring Node or One-Time Node.

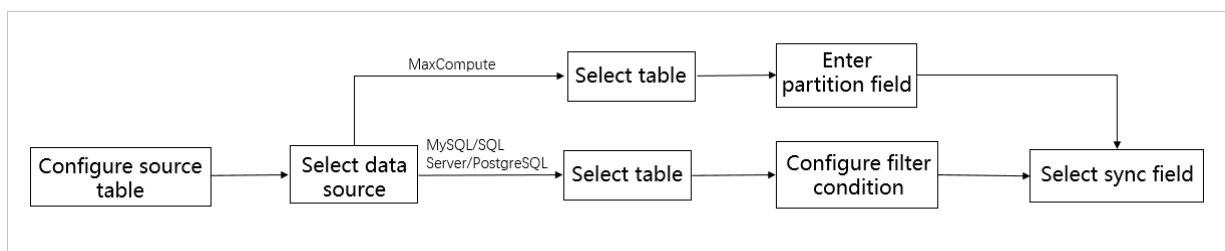
- **Recurring Node:** runs the task on a specified schedule.
- **One-Time Node:** runs the task after you click Run in the code editor.

## 9.6.4 Configure a sync task

This topic describes how to configure a sync task.

Configuration process

The following figure shows the process of configuring a sync task.



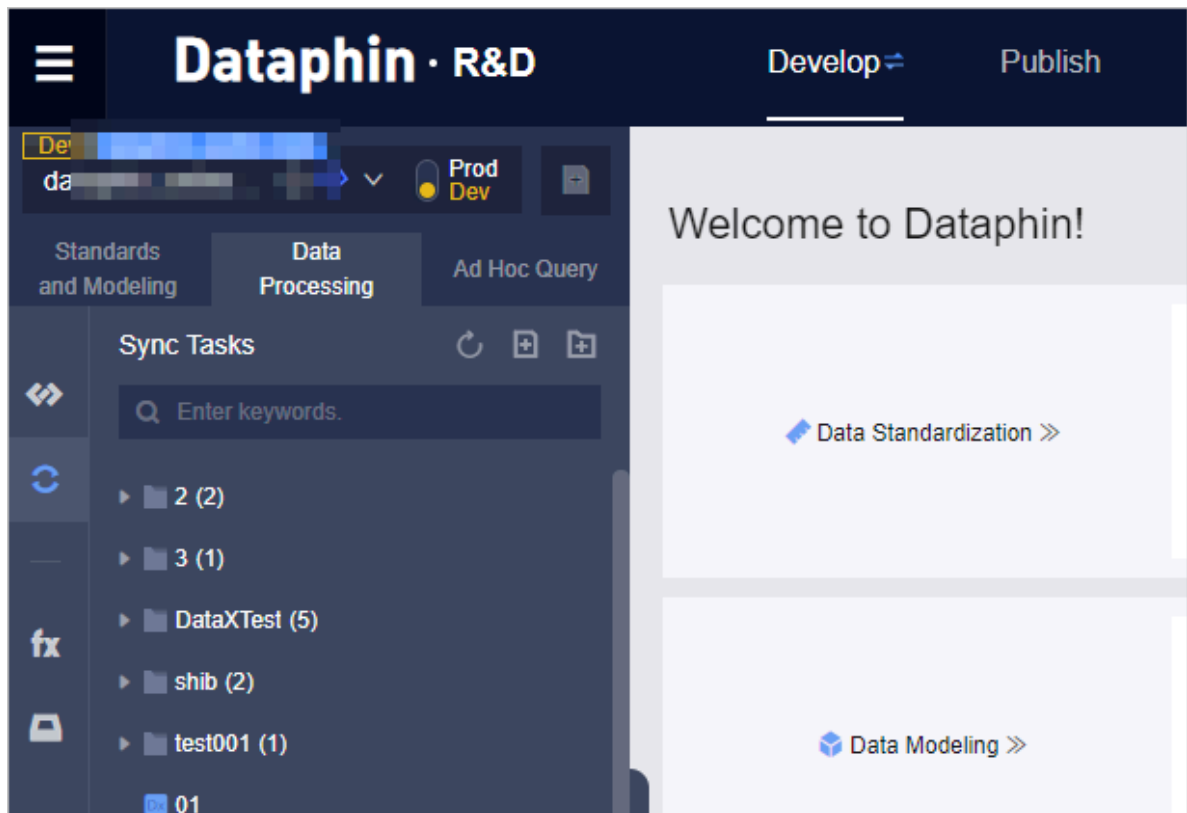
Currently, Dataphin supports the following data sources for the source table and destination table, which are sorted by type:

- **Relational databases:** MySQL, Vertica, Oracle, SQL Server, PostgreSQL, AnalyticDB, and Distributed Relational Database Service (DRDS)
- **Alibaba Cloud big data warehouse:** MaxCompute
- **Open-source big data warehouse:** Hive

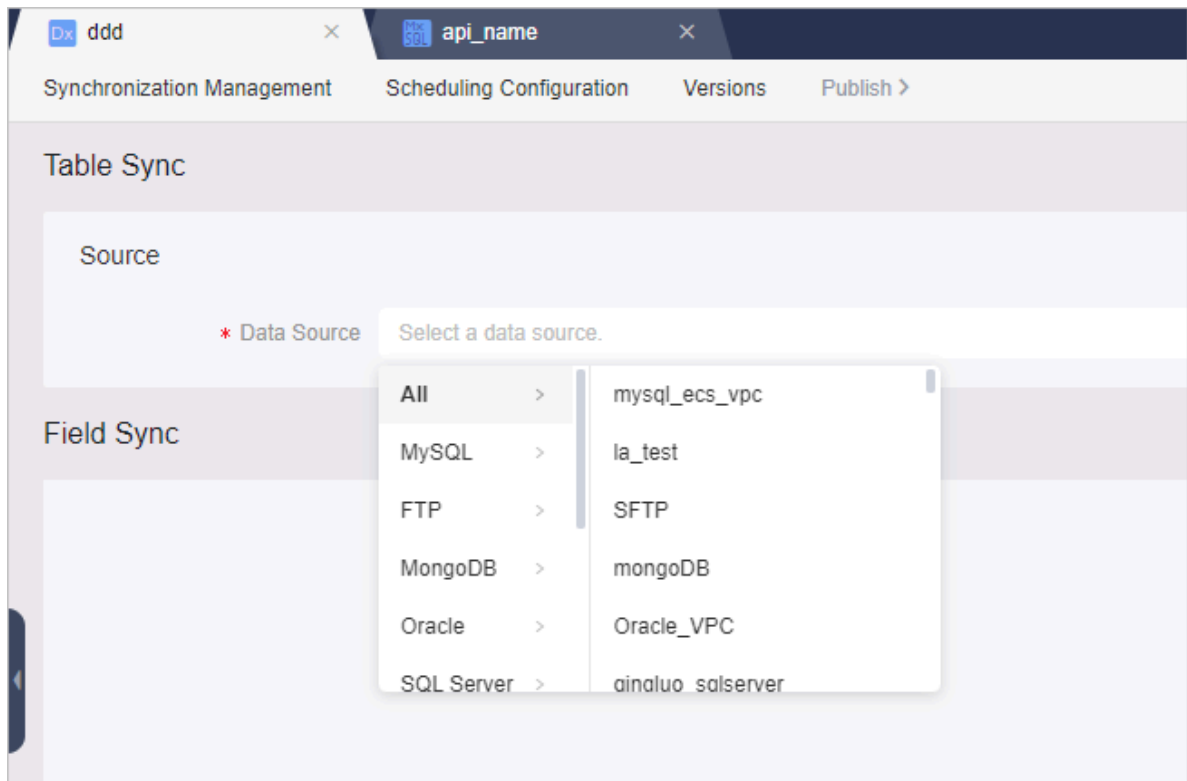
- **Unstructured databases: FTP and Hadoop Distributed File System (HDFS)**

Procedure

1. [Log on to the Dataphin console.](#)
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.
3. On the Develop tab of the R&D page, click the Data Processing tab, and then click Sync Tasks on the left-side navigation menu.



4. In the left-side sync task list, click a sync task. On the Synchronization Management tab that appears for the task, select the source database in the Source section.



5. Select a table from the source database. If the source database is a MySQL, PostgreSQL, SQL Server, or Oracle database, you need to select the Single Table or Multiple Tables mode.

- **Single Table mode**

If you select Single Table, you can search for a table by entering the table name or prefix and select the target table from the drop-down list.

The screenshot shows the 'Table Sync' configuration window. Under the 'Source' section, the 'Data Source' is set to 'la\_test'. The 'Mode' is set to 'Single Table' (indicated by a selected radio button). The 'Table' field contains the placeholder text 'Select a table.'. The 'Filter Conditions' field contains the placeholder text 'Enter filter conditions. For example, ds=\${bizdate}. Separating multiple parameters with commas (,).'

- **Multiple Tables mode**

a. If you select Multiple Tables, you can search for tables that have the same schema by using an expression. For example, enter `a2019061[0-5]` in the Table field.

The screenshot shows the 'Table Sync' configuration window with 'Multiple Tables' mode selected. The 'Data Source' is 'la\_test'. The 'Mode' is 'Multiple Tables' (indicated by a selected radio button). The 'Table' field contains the placeholder text 'Enter an expression.' and a blue icon for 'Confirm Match Details'. Below the 'Table' field, a note states: 'This supports enumeration, regex, and hybrid forms. For example, table\_[001-100]; table\_102.' The 'Filter Conditions' field contains the same placeholder text as in the previous screenshot.

b. After you enter the expression, click the Confirm Match Details icon next to the Table field to match tables with the metadata of the source database. In the Confirm Match Details dialog box that appears, the matching results

of the searched tables are listed, such as a20190610, a20190611, a20190613, and a20190614.

Table Sync

Source

\* Data Source: la\_test

\* Mode: ☐ Single Table ☒ Multiple Tables

\* Table: ddd

Filter Conditions: Enter filter conditions. For example, 'ddd'.

Target

Confirm Match Details

Expression: ddd

Tables Matched: 0

No data

Cancel OK

- c. If a table does not exist in the source database, the table name is highlighted. In this case, OK is dimmed and source fields cannot be loaded.
  - d. If all tables exist in the source database, click OK. Source fields of the first matched table are loaded.
6. Edit the source fields that are loaded after you select the source table. You can delete fields. If you want to add a field, click +Create Field.

Field Sync

Source Fields (0/0)

\* Source Table: ddd

+ Create Field

Name	Data Type	Actions
------	-----------	---------

7. Configure partition information or set filter conditions.



**Note:**

- If the source table is a partitioned table, you must configure partition information. You can use Linux shell wildcards to configure partition information in the Filter Conditions field. The asterisk (\*) matches zero or more characters. The question mark (?) matches a character. For example, enter month=201701,ds=20170101,/\*query\*/ month>=201701 for a



partitioned table or enter `ds=${bizdate}` for a dynamic partitioned table in the Filter Conditions field.

- If the source database is a MySQL, SQL Server, or PostgreSQL database, tables from this database have no partitions. You can set filter conditions in a WHERE clause and enter filter conditions excluding the WHERE keyword. For example, if you want to import data entries whose ID is greater than 2 and whose name is dataphin from the source table, enter `id>2 and name="dataphin"` in the Filter Conditions field.

## 8. Configure the destination table.

- a. Select a data source and a table, set the ingestion mode to insert or overwrite, and then configure partition information.

The screenshot shows the 'Target' configuration panel. It contains three dropdown menus: 'Data Source' (selected: mysql\_ecs\_nj), 'Table' (selected: Select a table.), and 'Conflict Resolution Policy' (selected: Select Conflict Resolution Policy). Below these is a 'Parse Solution' section with two options: 'Append Data' and 'Overwrite Data', both with an information icon.

- b. Select fields to be synchronized. Source fields and destination fields are automatically matched based on the field name. If source fields and destination fields have different names, you need to adjust fields to match them manually. To match fields, delete destination fields whose names are different from source fields. Then, move the pointer over a row in the destination field list and select a field from the pop-up list.

9. Set required limits for the sync task to control the concurrency and fault tolerance. In most cases, you can use the default settings. The following table describes the parameters.

Sync Limits

Speed Limit 1MB/s ▾ \* Concurrent Tasks 3 Error Threshold 0 The task will be terminated once it exceeds the error threshold.

Parameter	Description
Speed Limit	The maximum data transmission rate during synchronization. The default limit is 1 MB/s. Dataphin will try to reach but will not exceed this rate. This setting determines the size of scheduled resources when the sync task is running. The higher the speed limit is, the more resources are scheduled.
Concurrent Tasks	The number of concurrent data extraction tasks.
Error Threshold	The number of errors that are allowed during synchronization before the sync task is terminated. The default value is 0, which indicates that no errors are allowed.

10. Configure the scheduling policy. For more information, see [Configure the scheduling policy](#).

11. Click the Save icon and then the Submit icon to save and submit the sync task.

### 9.6.5 Configure the scheduling policy

Scheduling configuration is a common feature for code tasks and sync tasks. You must configure the scheduling policy for all scheduled tasks.

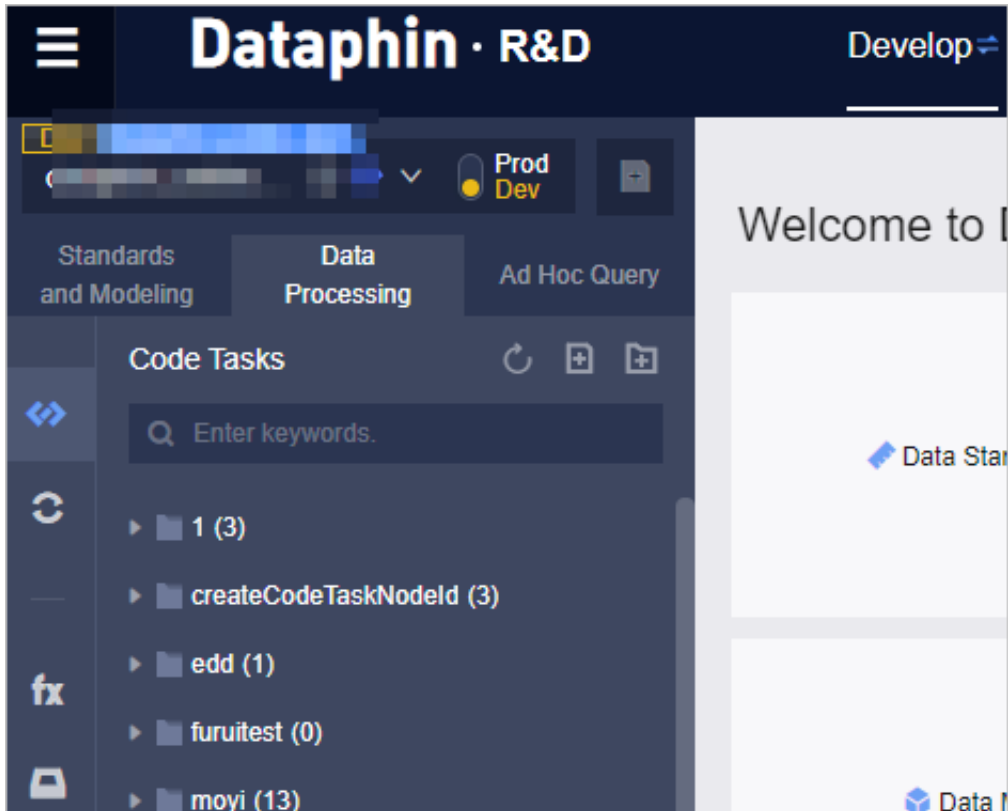
#### Context

This topic describes how to configure the scheduling policy for a code task.

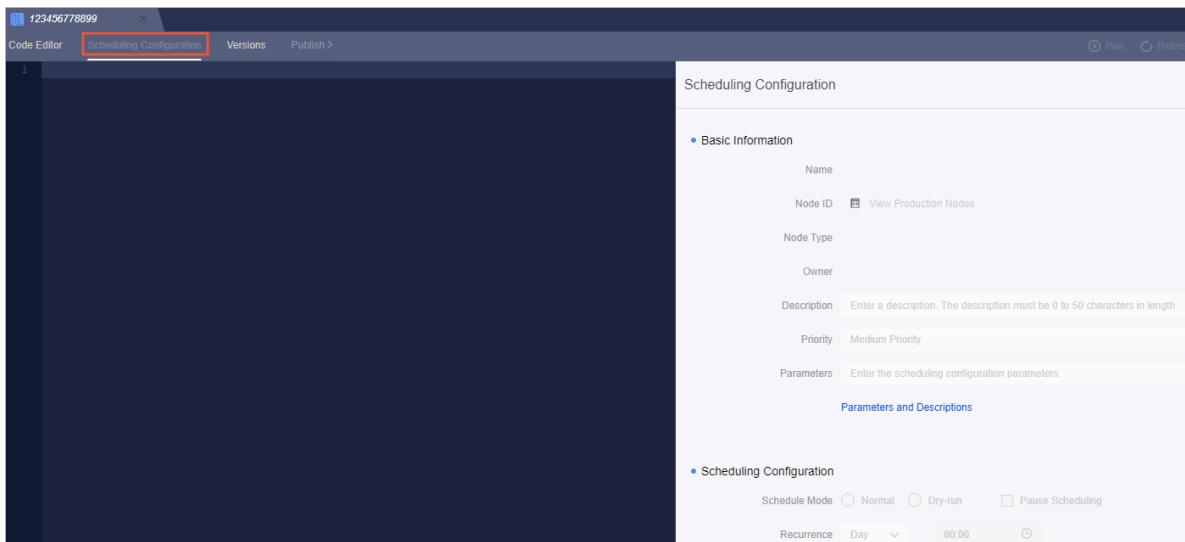
#### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.

3. On the Develop tab of the R&D page, click the Data Processing tab, and then click Code Tasks on the left-side navigation submenu.



4. In the left-side code task list, click a code task. On the configuration tab of the code task, click Scheduling Configuration in the top navigation bar.



5. In the Scheduling Configuration dialog box that appears, set parameters in the Basic Information section. You can click Node parameter configuration description to view the parameter description. Then, you can set parameters based on your requirements and the parameter description.

## 6. Set parameters in the Scheduling Configuration section.

### Schedule Mode

- **Normal:** runs the task based on the configured recurrence.
- **Dry-run:** runs the task based on the configured recurrence. However, when this task is scheduled to run, Dataphin directly returns a success message without running the node script.
- **Pause Scheduling:** specifies whether to pause the scheduling of the task. When this task is scheduled to run based on the configured recurrence, Dataphin directly returns a failure message without running the node script. You can select this check box when you temporarily do not need to run the task.

### Recurrence

You can set Recurrence to Day, Week, Month, Hour, or Minute.

- **Day:** runs the task at the specified time every day after its upstream nodes are run. That is, the task is run only after all of its upstream nodes are run

and when the specified running time arrives. The default running time is 00:00:00.

- **Week:** runs the task on the specified day or days of each week. During the days not specified, dry-run is implemented every day.
- **Month:** runs the task on the specified day or days of each month. During the days not specified, dry-run is implemented every day.
- **Hour:** runs the task at intervals of the specified hours every day
- **Minute:** runs the task at intervals of the specified minutes every day.

### Depend on Previous Instance

If you select Depend on Previous Instance, the current node is run after the previous instance of another node or the current node is run.

The screenshot shows the 'Scheduling Configuration' dialog box. It includes the following elements:

- Schedule Mode:** Three radio buttons for 'Normal' (selected), 'Dry-run', and 'Pause Scheduling'.
- Recurrence:** A dropdown menu set to 'Day' and a time input field set to '00:00' with a clock icon.
- Error Message:** A red text message stating 'Sorry,there has syntax error in Cron Expression.'
- cron expression:** A text input field.
- Depend on Previous Instance:** A checkbox that is currently unchecked.

## 7. Set parameters in the Dependency section.

The screenshot shows the 'Dependency' section in the Dataphin interface. It features two main sections: 'Upstream Dependency' and 'Current Node'. Each section contains a table with columns: Output Name, Node Name, Node ID, Owner, and Actions. The 'Upstream Dependency' table has a 'Create Upstream Dependency' button above it. The 'Current Node' table has an 'Add' button above it. Both tables currently display 'No data'.

a) Configure the upstream dependency. Specify the upstream nodes on which the current node depends. The specified nodes must exist.

- For an SQL-based task, click Start Parsing to automatically use the source tables referenced in the SQL script as the upstream dependency of the current node.
- Alternatively, you can click Create Upstream Dependency. In the dialog box that appears, search for a node based on the output name and add the node as the upstream dependency of the current node. Then, click OK to add the upstream dependency.



### Note:

- Each node output name is globally unique in Dataphin. In most cases, the output name of an SQL node is a table name.
- When Dataphin is initialized for each tenant, which is an enterprise, Dataphin generates a virtual node whose name starts with `virtual` as a root node for the tenant.

b) Add output names for the current node. In the Current Node section, click Add. In the dialog box that appears, enter an output name for the node. Then, click OK.



### Note:

- When you configure the upstream dependency, you search for a node based on the output name. The output name of a node contains the output information of the node.
- A node can have multiple output names. We recommend that you set the output name for the current node by observing uniform rules. This helps other users find this node when they configure the upstream dependency for their nodes.

## 9.6.6 Run sync tasks

Sync tasks are categorized into one-time tasks and recurring tasks. You can manage one-time and recurring tasks on the Scheduling page. For more information, see [Scheduling center](#).

## 9.7 Data modeling and development

### 9.7.1 Overview

This topic describes how to go to the workbench of a project.

#### Context

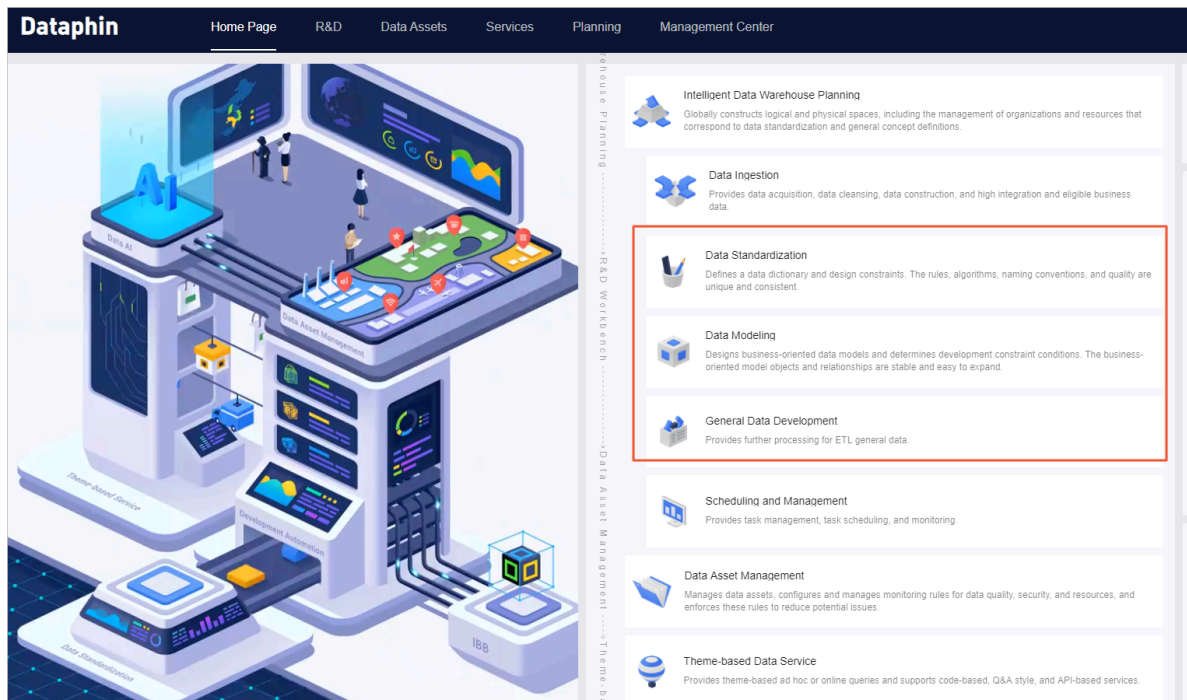
Dataphin provides systematic modeling and development features to build a data warehouse by using tools. Using the data modeling module, you can:

- Create dimensions and business processes in a top-down way.
- Refine the development of dimension tables, fact tables, aggregate tables, and the application data store layer, and then accumulate data assets at the common dimensional model layer. This facilitates business data application by hierarchy and optimizes computing and storage performance.

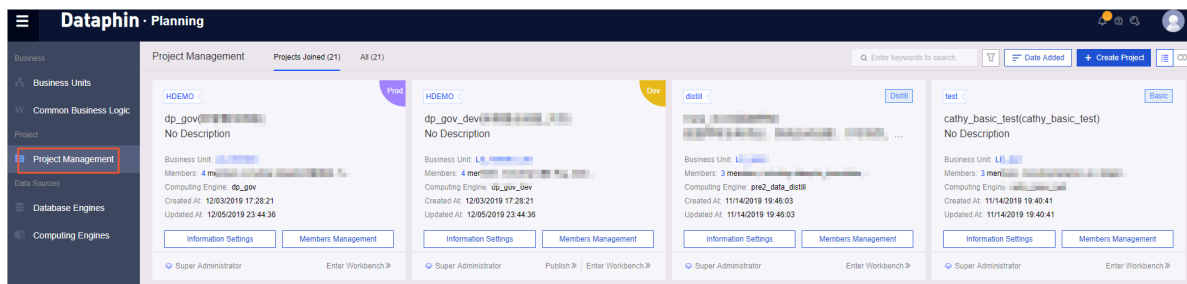
#### Procedure

1. [Log on to the Dataphin console](#).

2. On the Dataphin homepage, click **Planning** in the top navigation bar or **Intelligent Data Warehouse Planning** in the middle section to go to the **Planning** page.



3. On the Planning page that appears, click **Project Management** in the left-side navigation pane. On the Project Management page, click **Enter Workbench** for the target project. On the Develop tab that appears, you can develop data in the project.



## 9.7.2 Data standardization: Dimensions

A dimension is a statistical object. It is an entity that actually exists. By creating a dimension, you can standardize your business entities (or master data) during architectural design to ensure that they are unique.

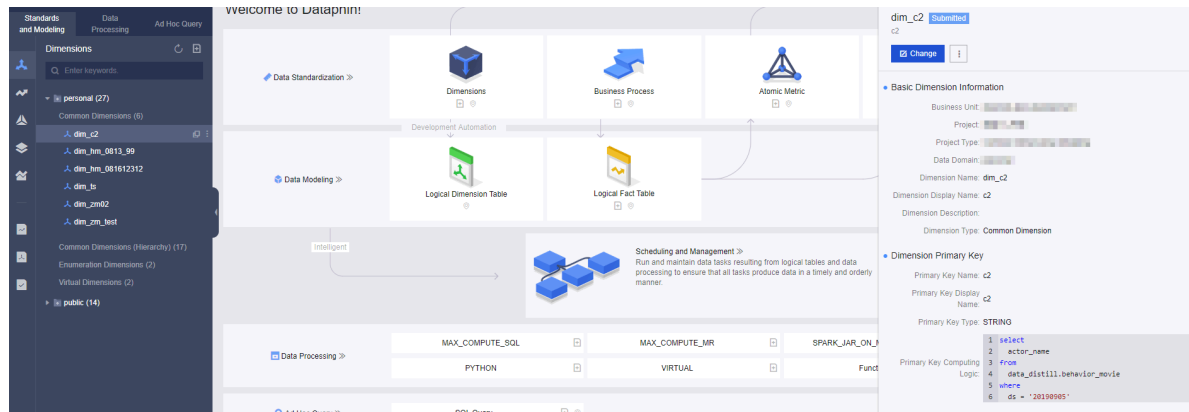
### Procedure

1. [Log on to the Dataphin console](#). Click **R&D** in the top navigation bar to go to the **R&D** page.

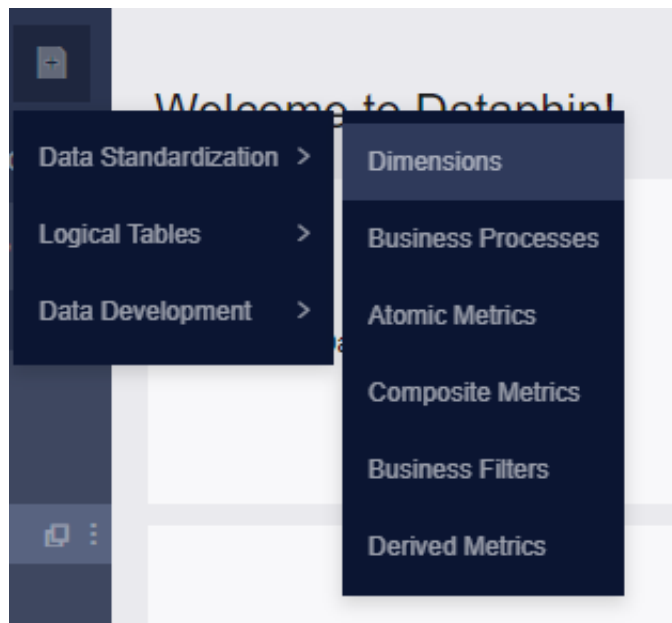


2. Choose Data Standardization > Dimensions. A list of dimensions that are created in the current project appears. Click a dimension to display its details on the right side of the console, as shown in *Figure 9-4: Dimension list*.

Figure 9-4: Dimension list



3. Choose Create > Data Standardization > Dimensions.



4. On the dimension creation page, configure the required parameters and click the Save icon.

Create Dimension

Basic Dimension Information

Business Unit  Project dataphin\_poc\_02 Project Type Application Data ...

\* Data Domain  Select a domain... \* Dimension Name dim\_ Enter a name \* Dimension Display ... Enter a name.

Dimension Description  Enter a dimension description. 0/128

Dimension Logic

Common Dimension Primary Key and Recursive Hierarchy Definition

\* Primary Key Name  Enter the dimension primary k \* Primary Key Display...  Enter the dimension primary k \* Primary Key Type STRING

\* Primary Key Computing Logic  Beautify  Example  Code Check  ?

1

Parent Dimension ☒ No ☐ Yes

Change Dimension Type

5. After creating a dimension, you can also click Change to modify the dimension settings. This includes the name, display name, primary key definition, and

parent-child relationship definition, as shown in [Figure 9-5: Modifying dimension settings](#).

Figure 9-5: Modifying dimension settings

dim\_c2

Submitted

c2

Change

Basic Dimension Information

Business Unit:

Project:

Project Type:

Data Domain:

Dimension Name: dim\_c2

Dimension Display Name: c2

Dimension Description:

Dimension Type: Common Dimension

Dimension Primary Key

Primary Key Name: c2

Primary Key Display Name: c2

Primary Key Type: STRING

Primary Key Computing Logic:

```
1 select
2 actor_name
3 from
4 data_distill.behavior_movie
5 where
6 ds = '20190905'
```

The screenshot displays the 'Basic Dimension Information' and 'Dimension Logic' sections of the Dataphin console. In the 'Basic Dimension Information' section, the 'Business Unit' is 'devprod\_test\_development', the 'Project' is 'hm\_0813', and the 'Data Domain' is 'personal'. The 'Dimension Name' is 'dim\_hm\_0813\_99' and the 'Dimension Display Name' is 'hm\_0813'. The 'Dimension Description' field is empty. In the 'Dimension Logic' section, the 'Primary Key Name' is 'hm\_0813\_99\_pk' and the 'Primary Key Display Name' is 'hm\_0813\_99\_cpk'. The 'Primary Key Type' is 'STRING'. The 'Primary Key Computing Logic' is shown in a code editor with the following SQL query: 

```
select actor_name
from data_distill.behavior_movie
```

. The 'Parent Dimension' is set to 'No'.

### 9.7.3 Data standardization: Business processes

A business process is a collection of all events in a business activity. By creating a business process, you can standardize a type of transaction event in business to ensure that it is unique.

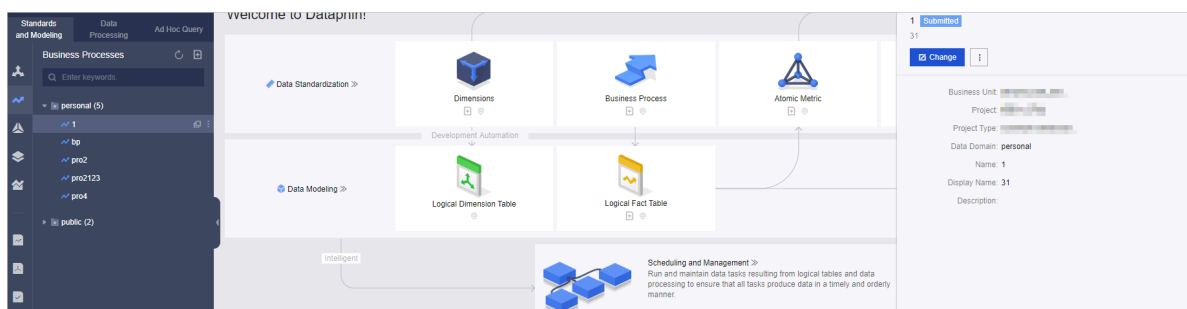
#### Context

We recommend that you define a business process that best suits your business development to avoid changing the business process in the future.

#### Procedure

1. [Log on to the Dataphin console](#). Click R&D in the top navigation bar to go to the R&D page.
2. Choose Data Standardization > Business Processes. Click a business process to display the business details on the right side of the console, as shown in [Figure 9-6: Business process list](#).

Figure 9-6: Business process list



3. Choose **Create > Data Standardization > Business Processes**. In the dialog box that appears, you can create a business process such as payment, order placement, and return of goods.
4. Click the **More** icon on the right of the business process you want to edit. From the drop-down menu, select **Change**. In the dialog box that appears, you can edit the business process settings such as name and display name.

#### 9.7.4 Logical tables: Logical dimension tables

A logical dimension table describes the attributes of a dimension. After you publish a dimension, Dataphin automatically creates a corresponding logical dimension table. Logical dimension tables are used to filter out and extract the common detail data of objects from business data.

##### Prerequisites

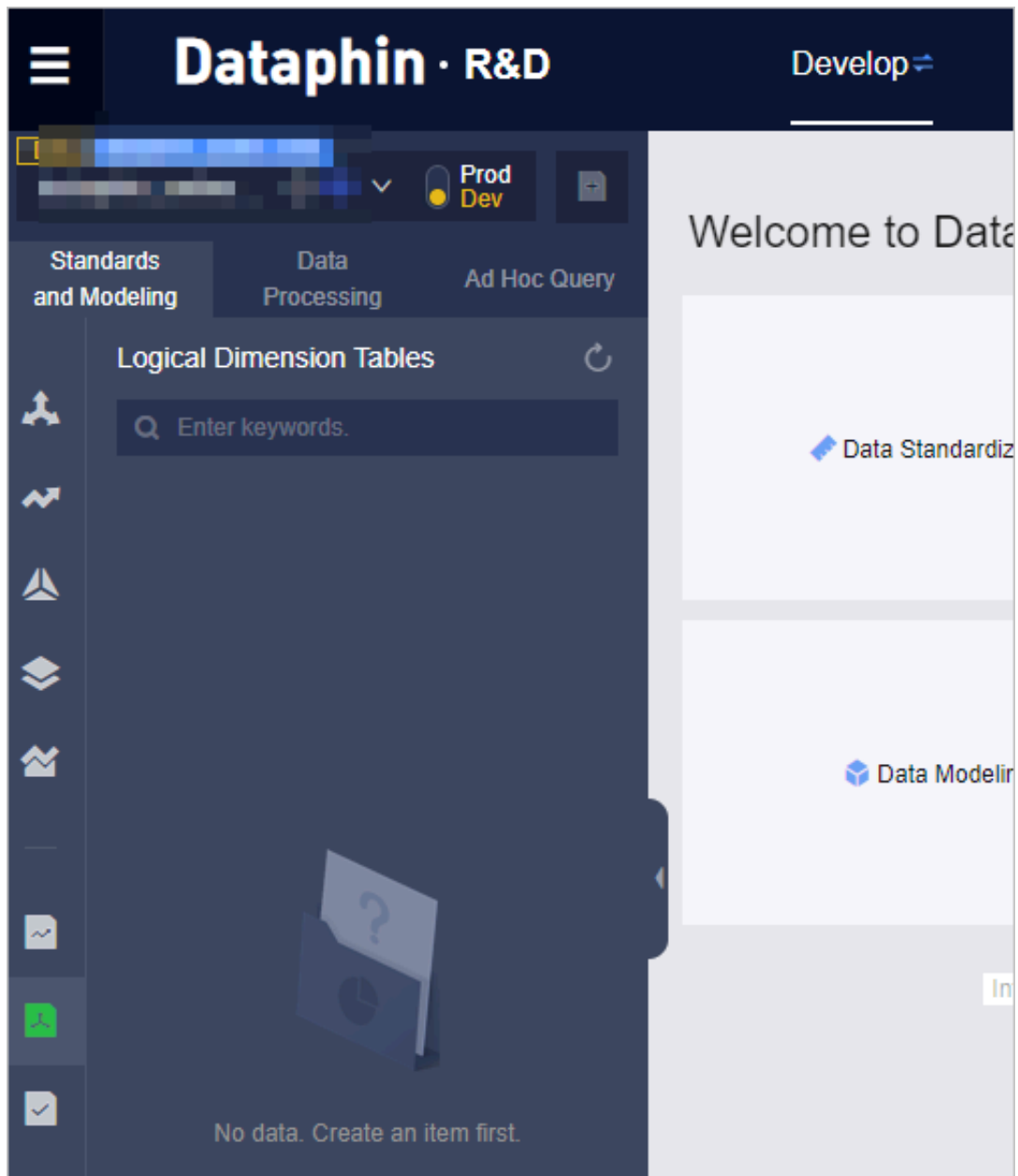
When you edit a logical dimension table, pay attention to the following two points:

- You only need to store persistent attributes of a dimension in its corresponding logical dimension table.
- You do not need to define fields for child dimensions if these fields are defined in the published parent dimension. Dataphin can automatically reference such fields.

##### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click **R&D** in the top navigation bar to go to the R&D page.

3. On the Develop tab of the R&D page, click the Standards and Modeling tab, and then click Logical Dimension Tables on the left-side navigation submenu.



4. In the left-side logical dimension table list, click a logical dimension table.

5. On the table configuration tab that appears, edit the logical dimension table as required. On the table configuration tab, you can perform the following operations:

- In the top navigation bar, you can click Table Information, Scheduling Configuration, Logical Table Conversion Task Settings, or Central Table Settings to edit the logical dimension table.
- In the upper-right corner, you can click the corresponding icon to save or submit the logical dimension table.
- In the Central Table section, you can view the model information, associate dimensions, add attributes, and add child dimensions.

### 9.7.5 Logical tables: Logical fact tables

A logical fact table models a specific business process and provides detailed information of transactions in the business process. Logical fact tables are used to extract details of common transactions from business data.

#### Context



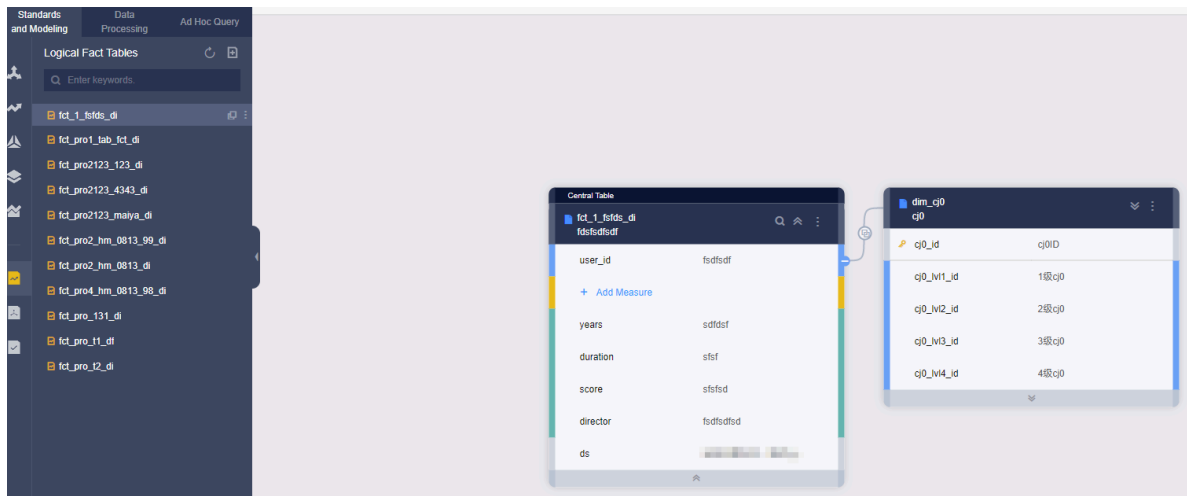
#### Note:

- Logical fact tables extract descriptive information about transactions without the need to contain attribute information of dimensions related to the transactions. Attribute information of the dimensions are contained in logical dimension tables.
- You can provide more detailed settings for a logical fact table based on the modeled business process. For example, you can add fields to the central table in the logical fact table model and configure the filter condition for the primary key.

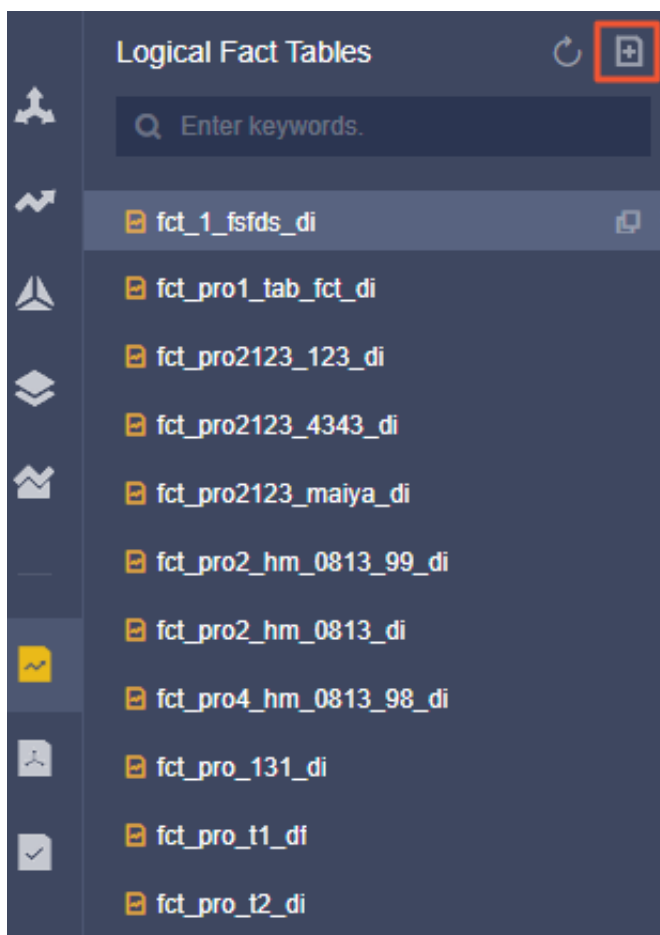
#### Procedure

1. [Log on to the Dataphin console](#). Click R&D in the top navigation bar to go to the R&D page.

2. In the left-side navigation pane, choose Standards and Modeling > Logical Tables > Logical Fact Tables, as shown in the following figure.

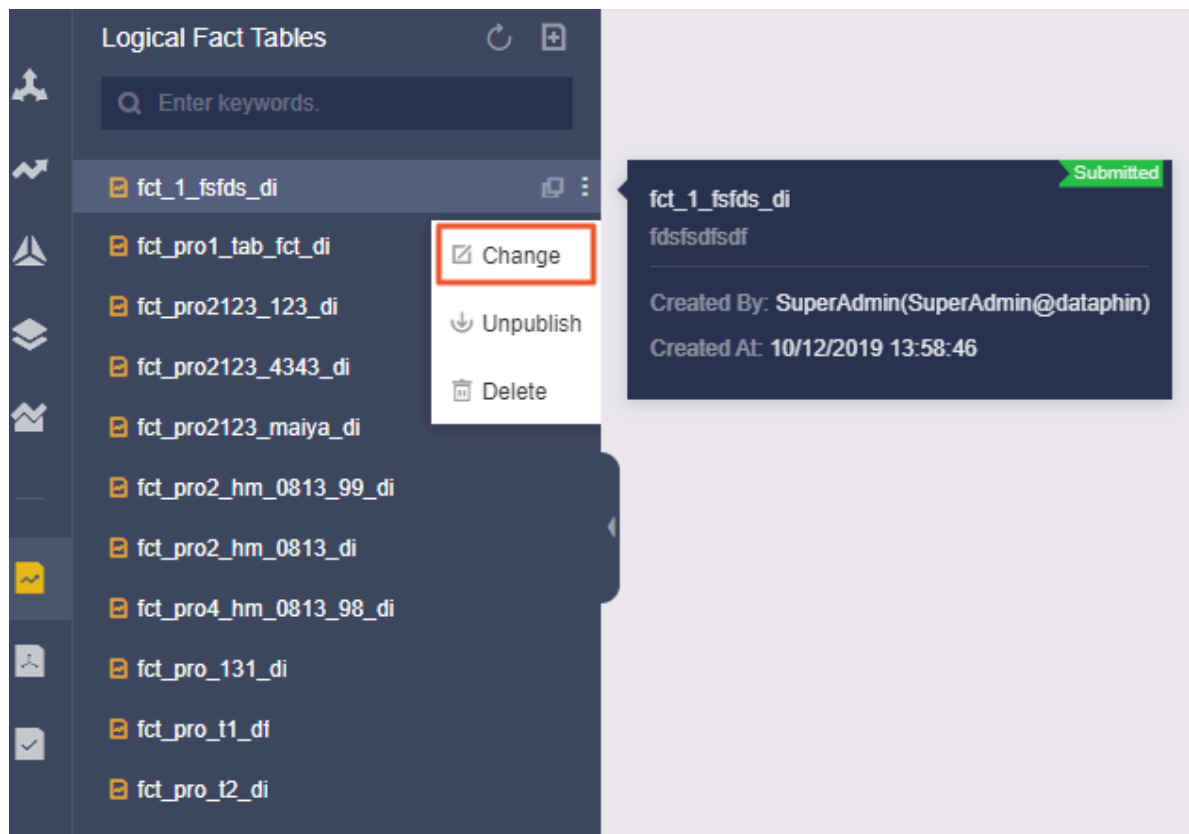


3. Click the Create File icon to create a logical fact table.





4. After creating a logical fact table, you can edit it as required.



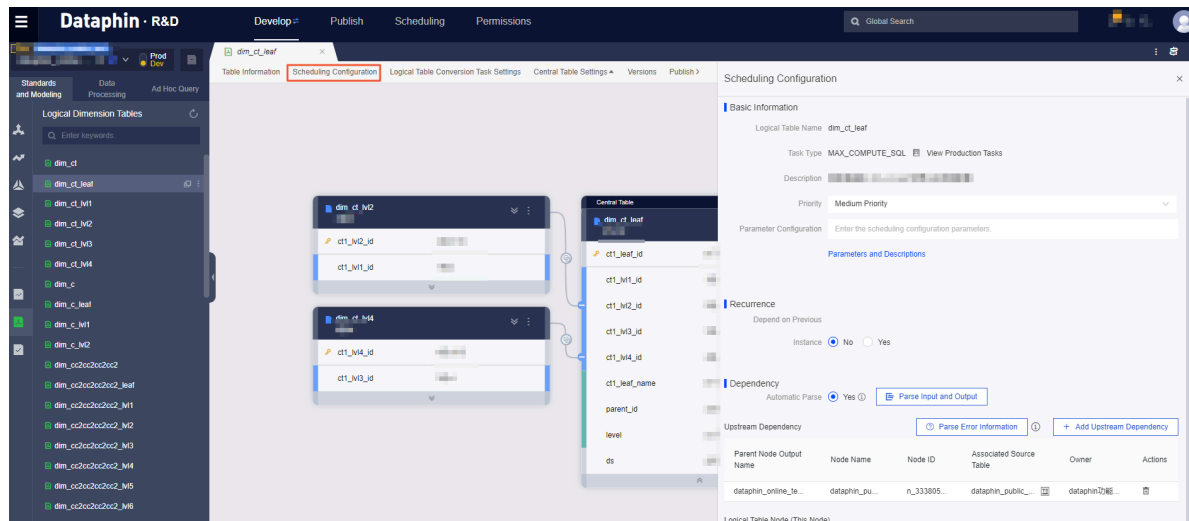
You have completed the standard creation of a data model.

### 9.7.6 Logical tables: Scheduling configuration

The scheduling configuration method for logical dimension tables is the same as that for logical fact tables. This topic describes how to view and configure the scheduling policy of a logical dimension table.

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.
3. On the Develop tab of the R&D page, click the Standards and Modeling tab, and then click Logical Dimension Tables on the left-side navigation submenu.
4. In the left-side logical dimension table list, click a logical dimension table. On the table configuration tab that appears, click Scheduling Configuration in the

top navigation bar. In the Scheduling Configuration dialog box that appears, you can view and modify the scheduling policy of the logical dimension table.



The Scheduling Configuration dialog box contains the following parameters:


- **Basic Information**

The Basic Information section displays the logical table name, task type, description, and parameter settings. In the Parameter Configuration field, you can specify multiple parameters by separating them with semicolons(;).

- **Recurrence**

- You can specify whether the operation on the logical table is dependent on the previous instance. If you set Depend on Previous Instance to Yes, the operation on the logical table is performed after the operation of the previous instance is completed.
- After you set Depend on Previous Instance to Yes, you can select node dependency in the current logical table or field dependency in the custom node. You can add multiple dependencies. The following table describes the two dependency methods.

Dependency	Description
Node dependency in the current logical table	By default, all fields in the logical table are listed for you to select. You can also enter a keyword to search for all fields that contain the keyword.

Dependency	Description
Field dependency in the custom node	<p>You can select a node ID from the corresponding drop-down list or enter a keyword to search for a node ID. All fields in the node are listed for you to select.</p> <div> <b>Note:</b> If you move the pointer over a dependency, the Delete icon appears for you to delete the dependency.</div>

- **Dependency**

The Dependency section consists of Automatic Parse, Upstream Dependency, and Logical Table Node (This Node). The following table describes relevant parameters.

Parameter	Description
Automatic Parse	Automatically parses the upstream dependency and logical table output based on the code content. If you click Parse Input and Output, Dataphin automatically parses the upstream dependency and logical table output.
Upstream Dependency	Displays the parent nodes on which the nodes of the current logical table depend. The information about each parent node includes the node output name, node name, node ID, associated source table, and owner of the node.
Logical Table Node (This Node)	Displays all output nodes of the current logical table. The information about each node includes the node output name, node name, node ID, and owner of the node.

### 9.7.7 Data standardization: Atomic metrics and business filters

This topic describes how to view atomic metrics and business filters existing in the Dataphin system and how to open the creation pages for them.

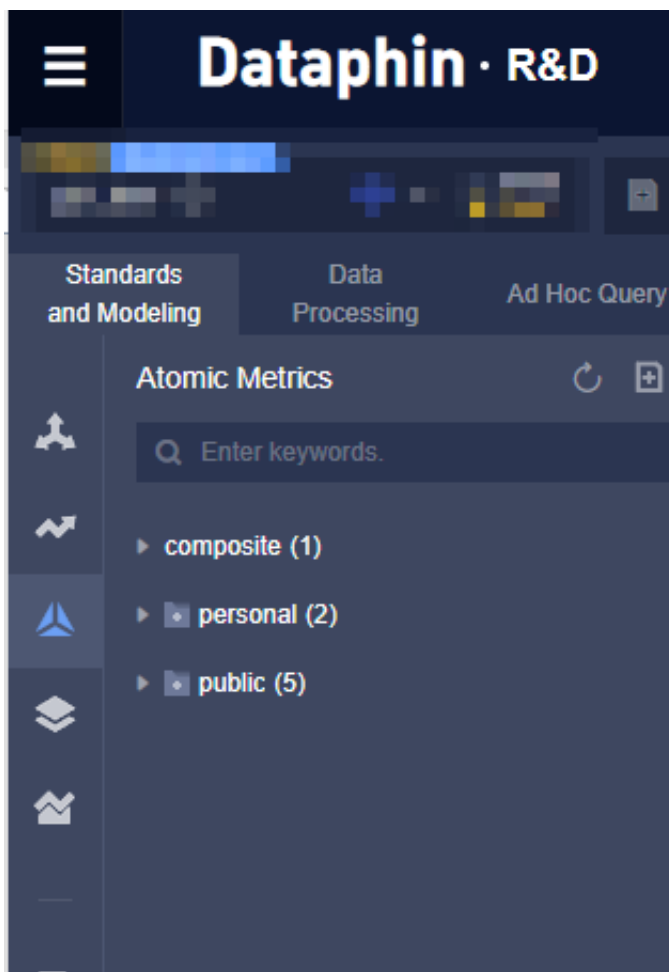
#### Context

An atomic metric and business filter are the computing logic and limitation commonly used in business. An atomic metric and business filter are expressions formulated based on fields in a logical table. These are reusable common data elements extracted to calculate aggregated data. Creating atomic metrics and business filters is to extract frequently used atomic computing logic and limitations for reuse.

## Procedure

1. *Log on to the Dataphin console.*
2. On the Dataphin homepage, click R&D in the top navigation bar to go to the R&D page.
3. Choose Develop > Standards and Modeling > Atomic Metrics to view the existing atomic metrics, as shown in *Figure 9-7: Atomic metric list.*

Figure 9-7: Atomic metric list

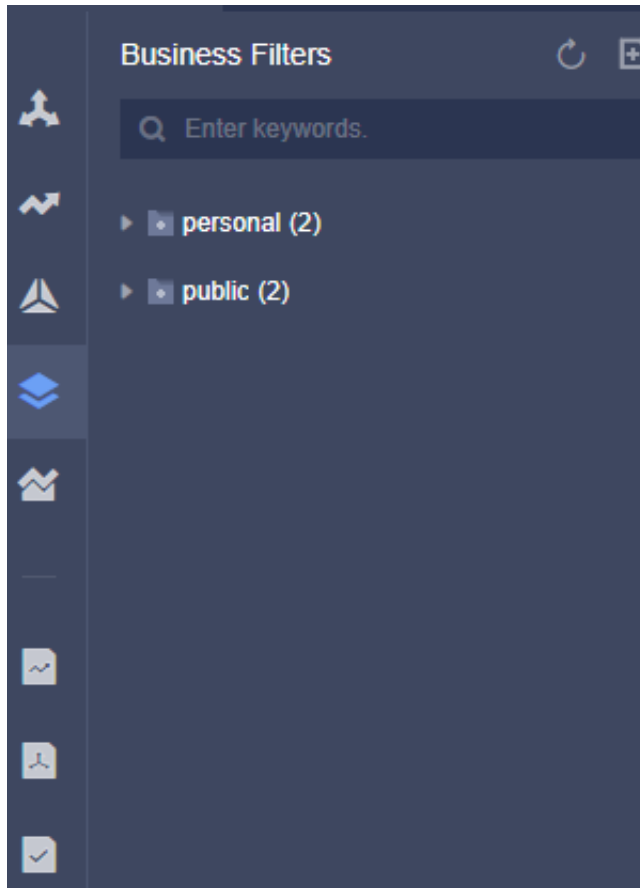


### Note:

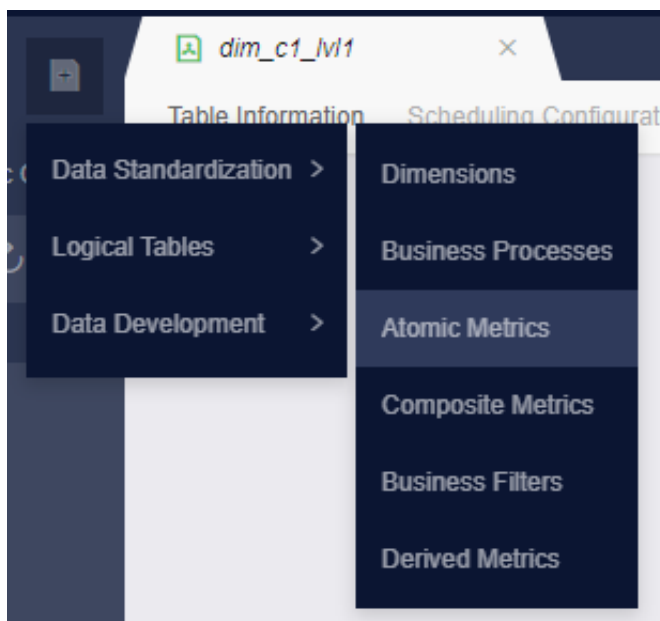
You can also click Data Standardization on the homepage to go to the Develop page. In the left-side navigation pane, click the Atomic Metrics submenu to view the existing atomic metrics. You can move the pointer over an icon to expand its corresponding submenu.

4. Choose **Develop > Standards and Modeling > Business Filters** to view existing business filters, as shown in *Figure 9-8: Business filter list*.

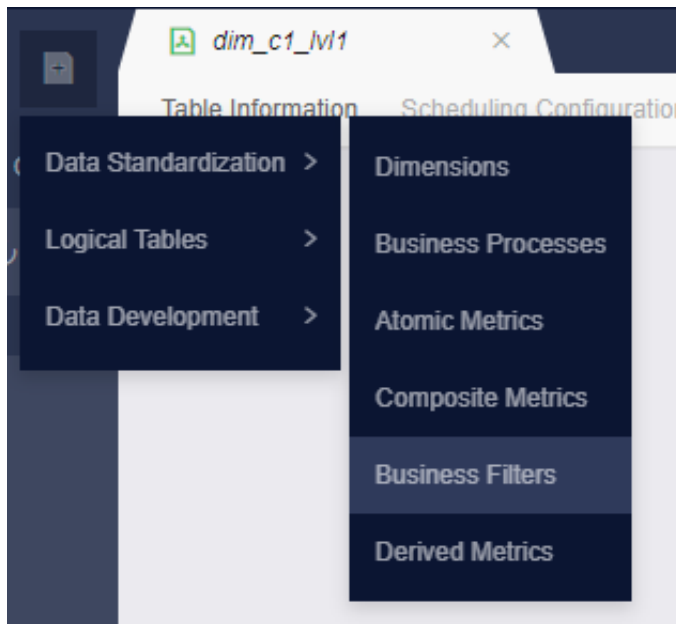
Figure 9-8: Business filter list



5. Click the **Create** icon next to the project name. Choose **Data Standardization > Atomic Metrics** to open the atomic metric creation page.



6. Click the Create icon next to the project name. Choose **Data Standardization > Business Filters** to open the business filter creation page.

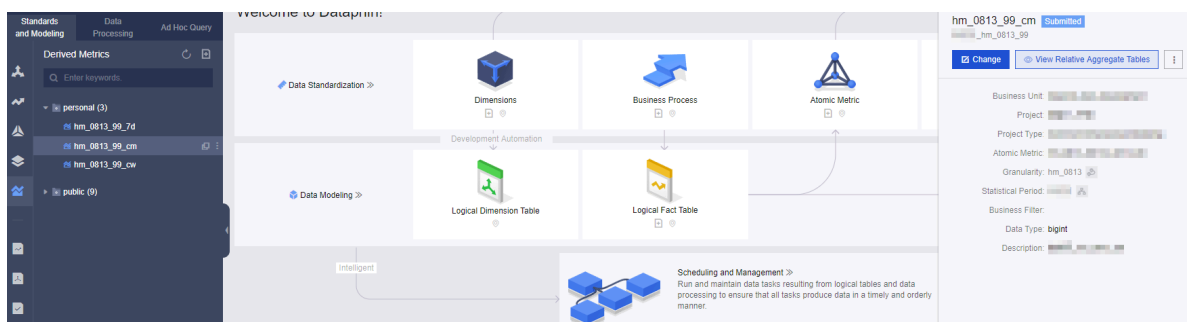


### 9.7.8 Data standardization: Derived metrics

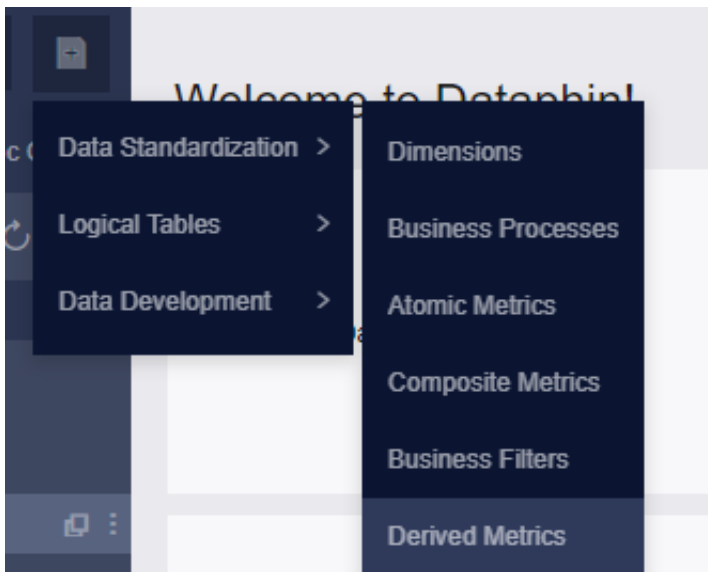
A derived metric is a commonly used statistical metric. It is used to aggregate the data of an object group in a specific range during a time period. Therefore, a derived metric is defined by the time period (statistical period), statistical object (statistic granularity), range (business filter), and calculation method (atomic metric). When creating a derived metric, you need to select the preceding elements and specify a name and display name for the derived metric.

#### Procedure

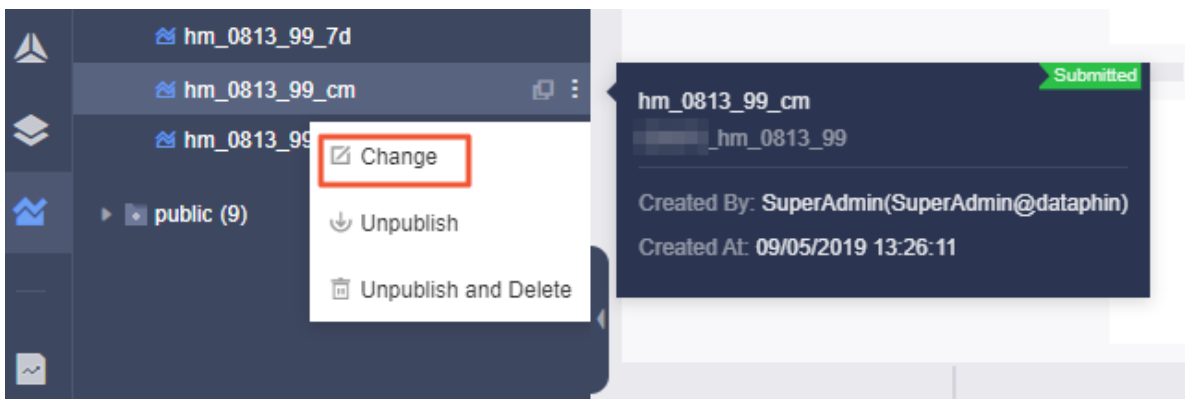
1. [Log on to the Dataphin console](#). Click R&D in the top navigation bar to go to the R&D page.
2. In the left-side navigation pane, choose **Standards and Modeling > Data Standardization > Derived Metrics**, as shown in the following figure.



3. Click the plus sign icon next to the project name. From the drop-down menu, choose Data Standardization > Derived Metrics. On the page that appears, you can create a derived metric.



4. In the left-side navigation pane, click the More icon next to the derived metric you want to edit. From the drop-down menu, select Change. In the dialog box that appears, you can edit the derived metric.



## 9.8 Data distillation

### 9.8.1 Instructions for data distillation

Based on data accumulation by data modeling, the master data in the entire system, that is, the core objects throughout all isolated business systems are identified and

associated. Data silos are interconnected, and the highly valuable tags that can be directly used are extracted.

## Overview

You can configure parameters on a GUI to extract and identify the mappings between various types of IDs, such as a shopping membership ID, video viewer ID, shopping device MAC address, and video viewing device ID. Then the behavioral data (such as online shopping and video viewing) of people in various behavioral domains can be collected. Based on algorithmic models, preference tags such as natural attributes (gender), social attributes (occupation), interest attributes (brands) of people can be further extracted. This achieves the identification of mappings between target object-related IDs, standardized and structured aggregation of all behaviors of target objects, and quick creation of target object-related tag attributes. In this way, you can quickly construct enterprise user data assets and use them in data application products, in order to implement marketing and advertising strategies.

The data distillation module completes data correlation and in-depth data mining intelligently and generate code and tasks to extract value from the data of target objects. This module accumulates data assets to make a data distillation center. Compared with traditional tag production, the data distillation module enjoys advantages in terms of data standardization, controlled production of high-quality tags, in-depth extraction of value from data, algorithmic transparency, customizable and upgradable tag production, use of business experience for tag production, and lower technical readiness overheads for tag production.

Currently, the data distillation module gives priority to the OneID system that takes consumers as target objects. This module consists of two parts: Behavior Engine and Tag Engine.

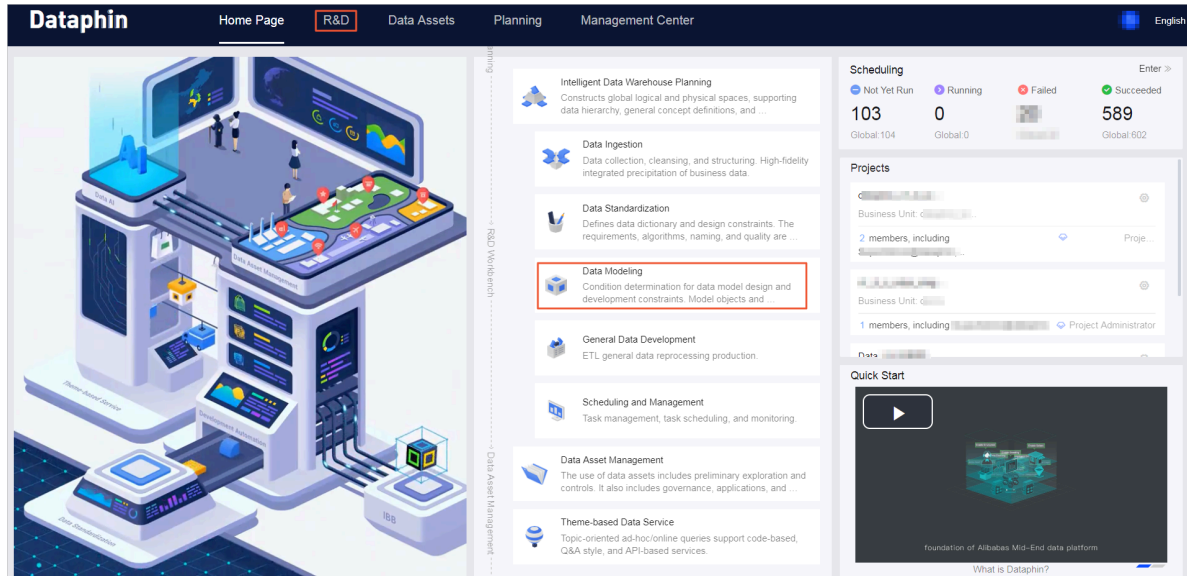
Go to the Distilling page

**Follow these steps:**

1. [Log on to the Dataphin console.](#)



2. Click R&D in the top navigation bar, or click Data Modeling on the home page to open the R&D page, as shown in the following figure:



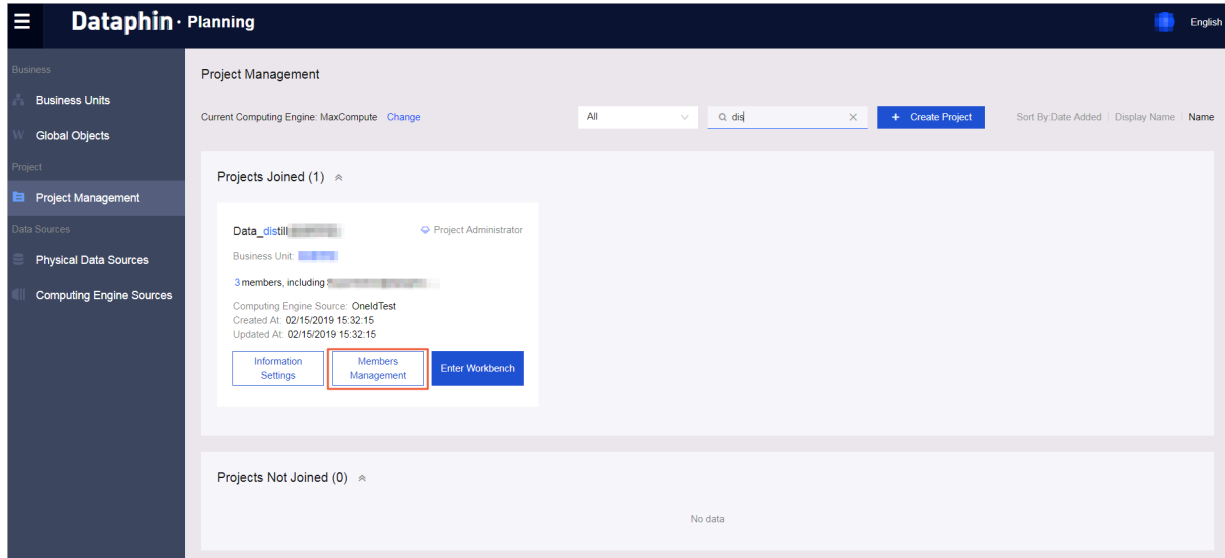
3. Go to the R&D page. Click Distilling in the top navigation bar to go to the Distillingpage.

Initialize your data distillation workspace

**Before you start data distillation, configure the required computing engine source to initialize your data distillation workspace. The effect of initialization is equivalent to the project creation performed before you use the R&D workbench . The super administrator performs the initialization configuration, such as selecting a business unit and binding a computing engine source, when entering the Distilling page for the first time. Only the super administrator can perform this operation. After the operation is completed, the system project for data distillation**

is generated. Other members of Dataphin can perform distillation development on the Distilling page by becoming a member of this project.

Figure 9-9: Member management

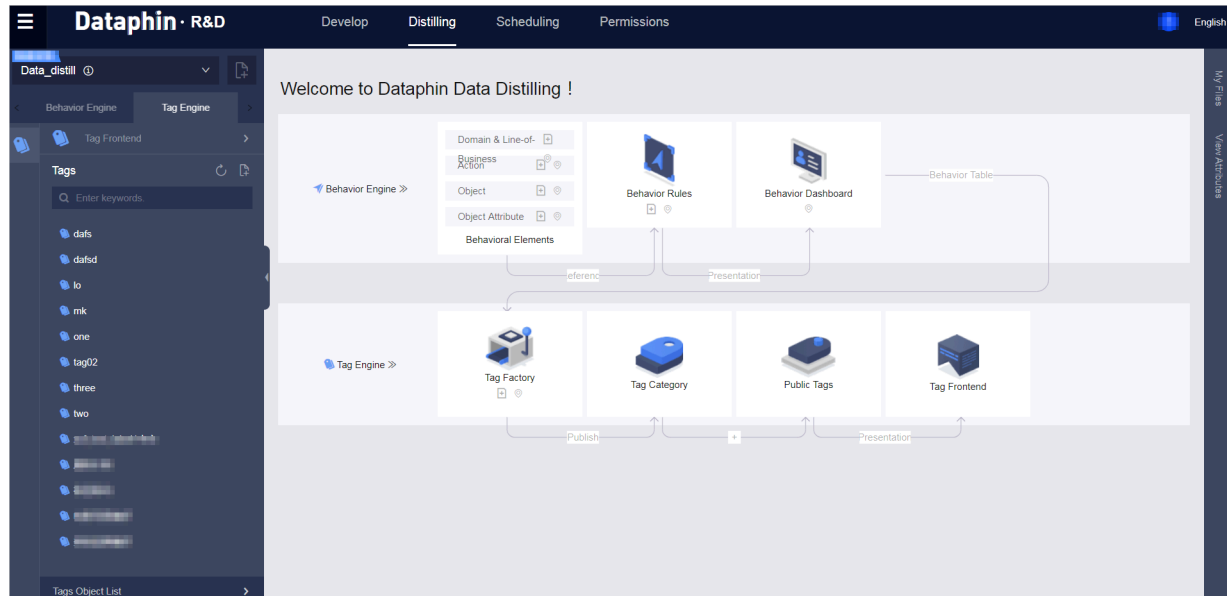


## Distilling page introduction

On the Distilling page, an operation guide is provided for you to learn about a workflow that shows the relationship between behavioral elements (including domains & lines of business, actions, objects, and object attributes) behavior rules, behavior dashboard, tag factory, and other modules of Tag Engine.

You can also view the publishing status of an object by hovering over the object in the left-side navigation pane.

Figure 9-10: Distilling page



## 9.8.2 Behavior Engine

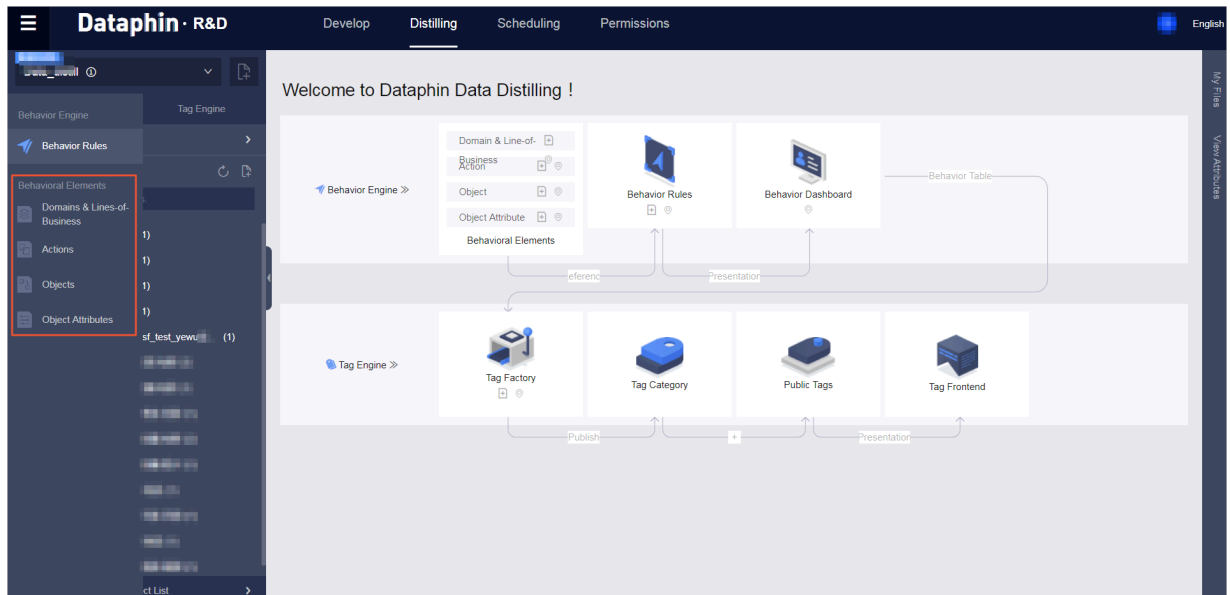
The Behavior Engine allows you to configure detailed behavior and displays statistical behavioral data. Behavior of identified IDs can be recorded. This is similar to creating a profile by summarizing a person's basic information and recording behavioral information or additional information based on each business activity in which that person is involved. For example, a person watched a celebrity's Chanel commercial three times in a specific community on the Double 11, and used a coupon to buy a Chanel bag endorsed by a celebrity in a specific shopping mall on Christmas. This can be recorded as behavioral information.

### 9.8.2.1 Define behavioral elements

Dataphin allows you to manage behavioral elements. You can define standard behavior categories and behavioral elements, for example, watching videos in a specific platform (entertainment behavior) and buying goods in a specific commercial district (shopping behavior).

You can manage different types of domains and lines-of-business, actions, objects, and object attributes to structure detailed behavioral data. For example, you can define and manage the e-commerce behavioral domain, domestic and international

lines-of-business, payment actions, commodity objects, and object attributes such as category, brand, price, size, and color.



## Domains and Lines-of-business

**You can manage behavioral domains and the lines-of-business in each behavioral domain.**

Figure 9-11: Create a domain

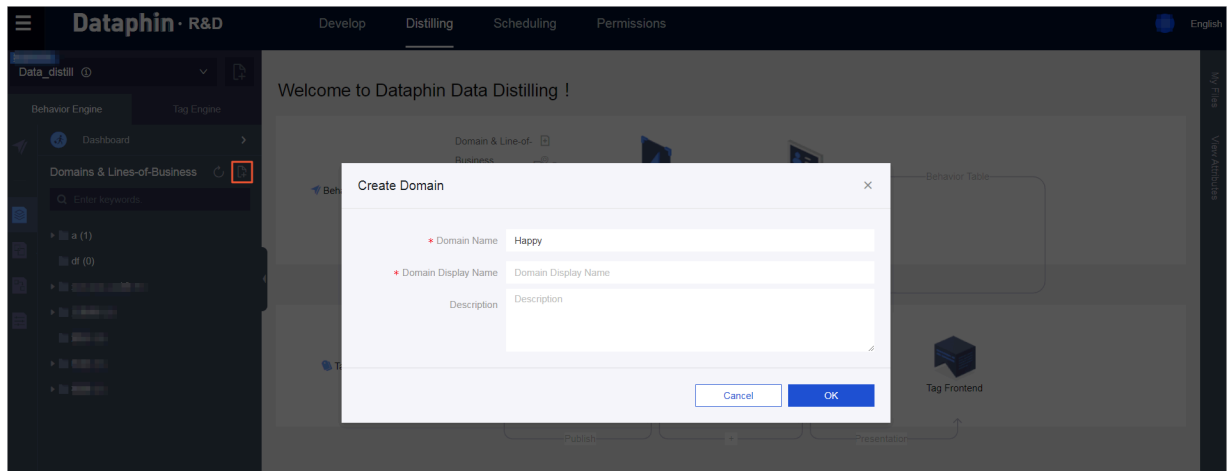
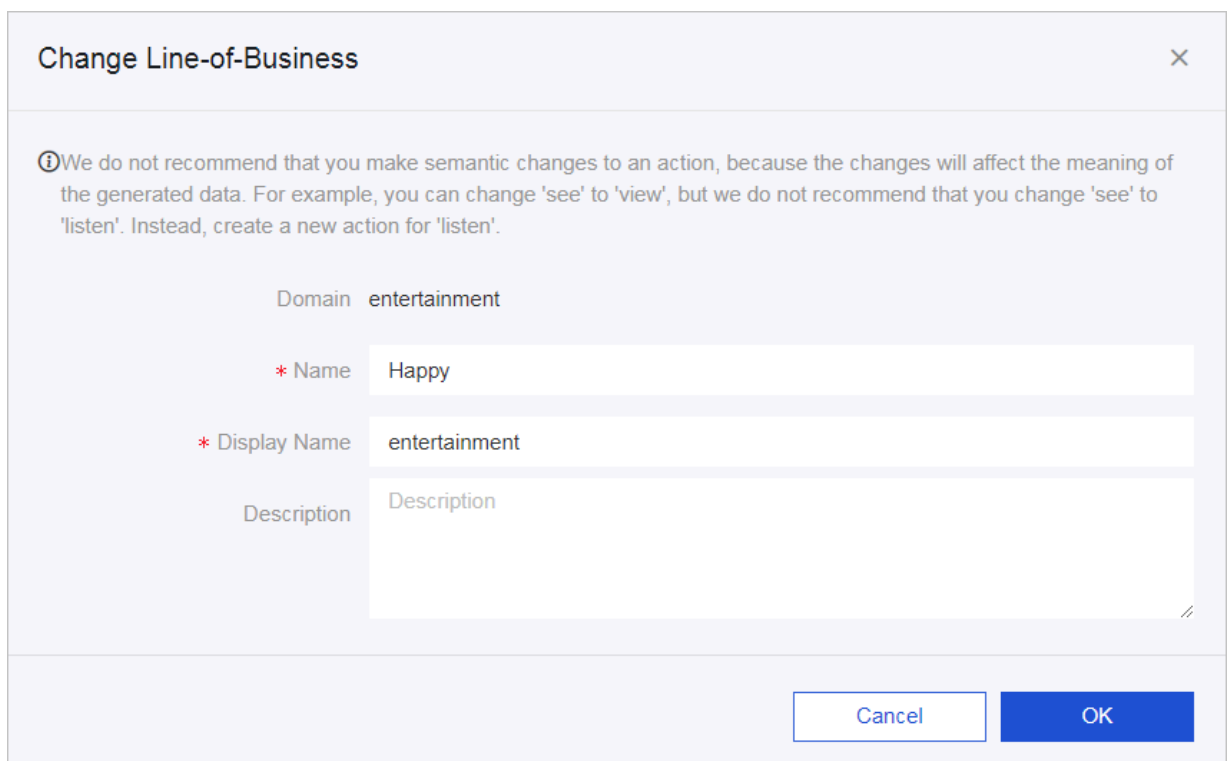


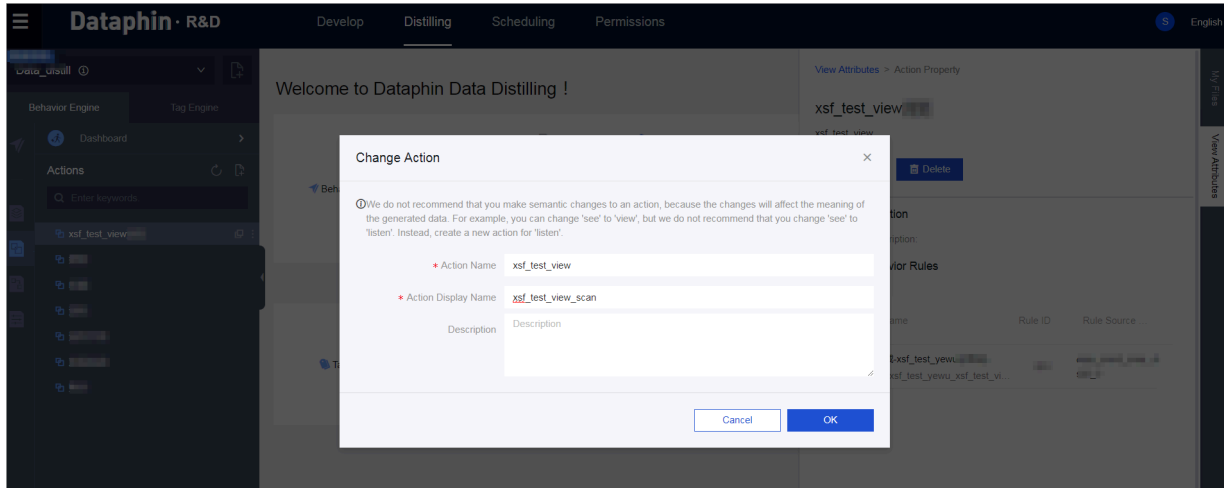
Figure 9-12: Edit a line-of-business



## Actions

**You can manage actions, such as viewing, commenting, adding to favorites, placing orders, and paying.**

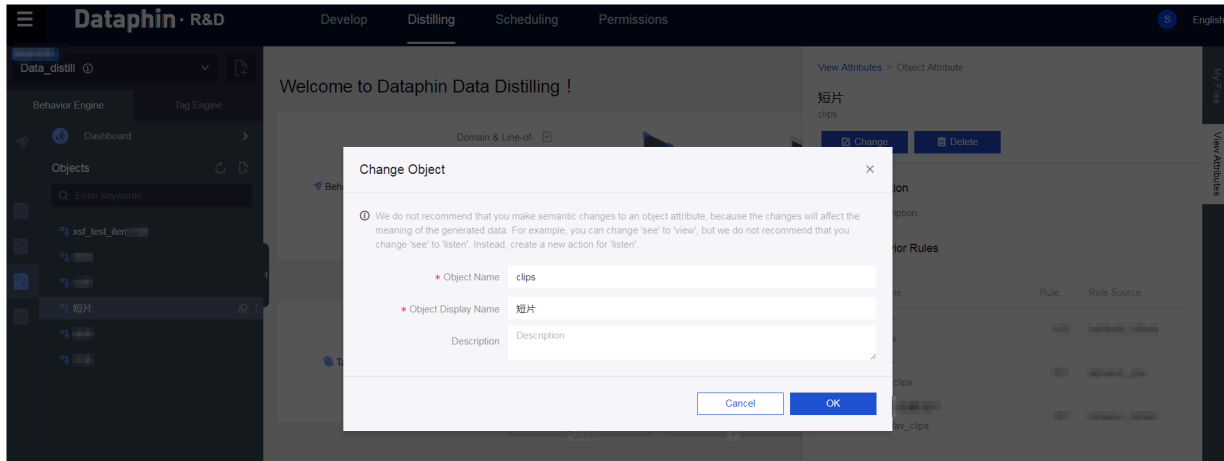
Figure 9-13: Edit an action



## Objects

**You can manage objects, such as products, goods, videos, music, and projects.**

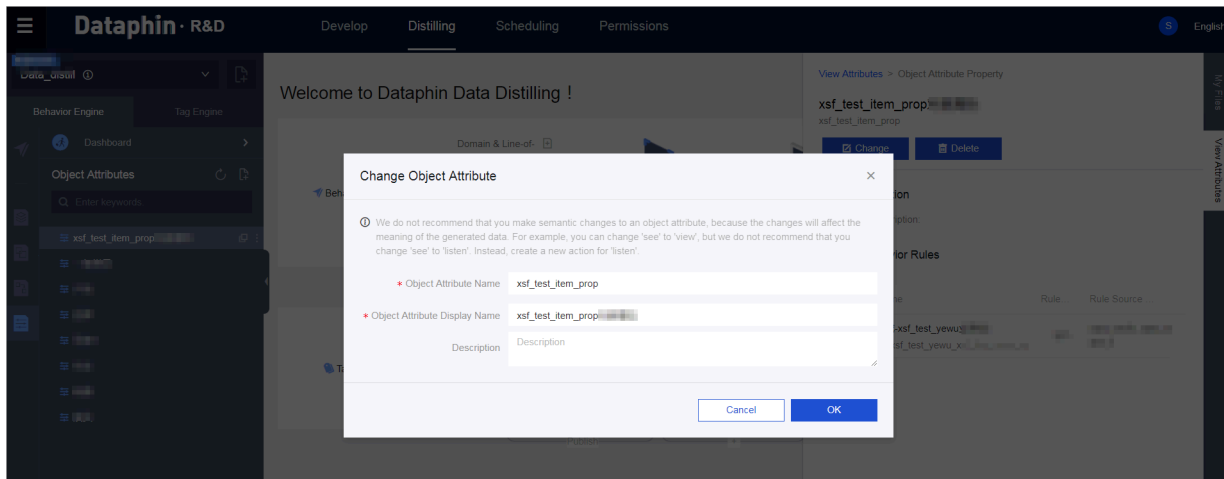
Figure 9-14: Edit an object



## Object attributes

**You can manage object attributes, such as video style, genre, and celebrity.**

Figure 9-15: Edit an object attribute



### 9.8.2.2 Define and design behavior rules

**Dataphin allows you to manage behavior rules. Based on the defined behavioral elements, you can further configure the corresponding data source, and data processing and cleansing rules.**

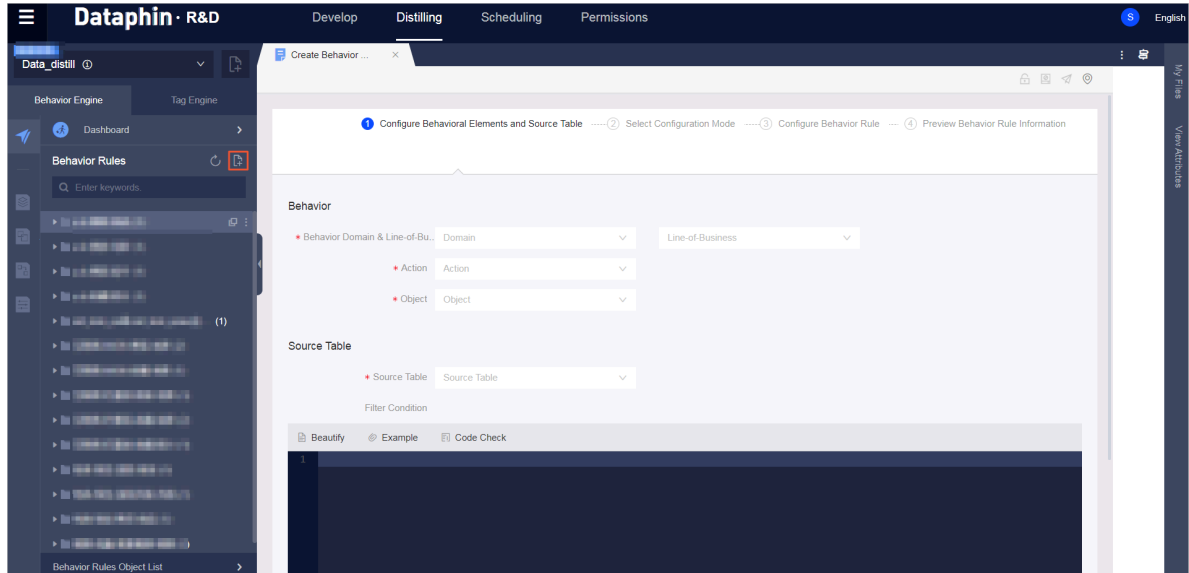
**You need to configure the data source for the behavior that is based on a behavioral domain, a line-of-business, an action, and an object. This matches the standardized and structured behavior with the actual data.**

**Follow these steps to create a behavior rule:**

- 1. Select a behavioral domain, a line of business, an action, an object, and a source table. Pay attention to the configuration of the source table and the filter condition, because you must ensure that the raw data meets the behavioral**

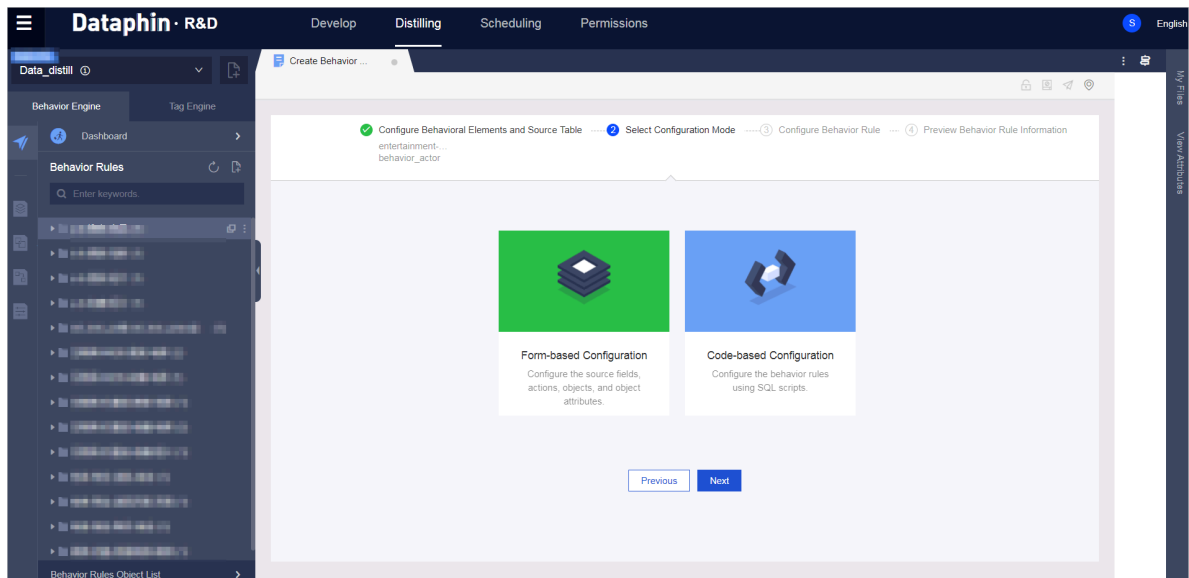
data requirements. The filter condition is the specific cleansing rule (WHERE condition) for the source table.

Figure 9-16: Configure behavioral elements and the source table



2. You can configure a behavior rule in two modes (form-based and code-based). Select a configuration mode.

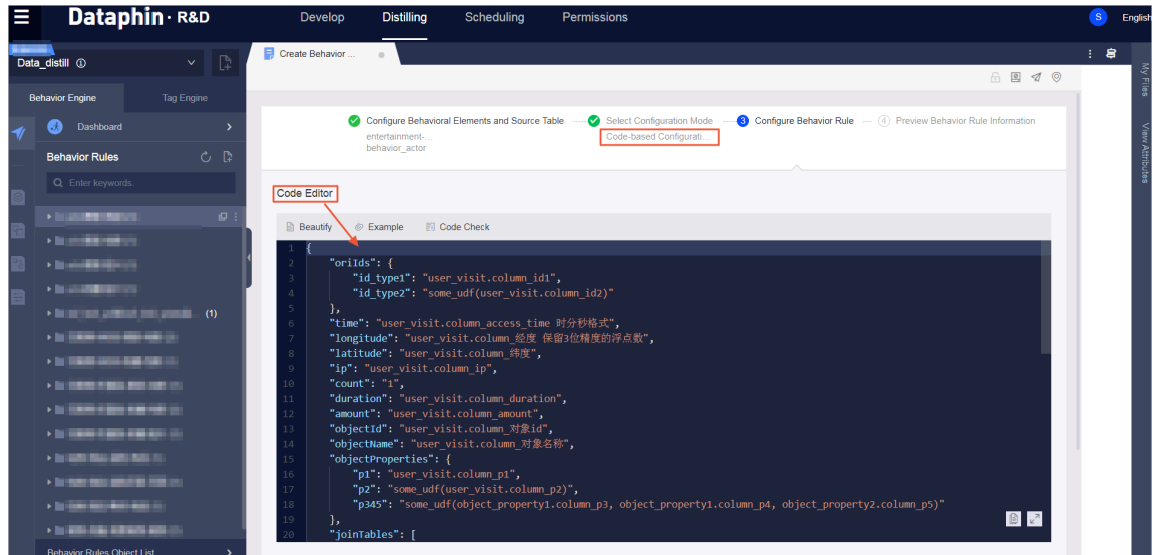
Figure 9-17: Select a configuration mode



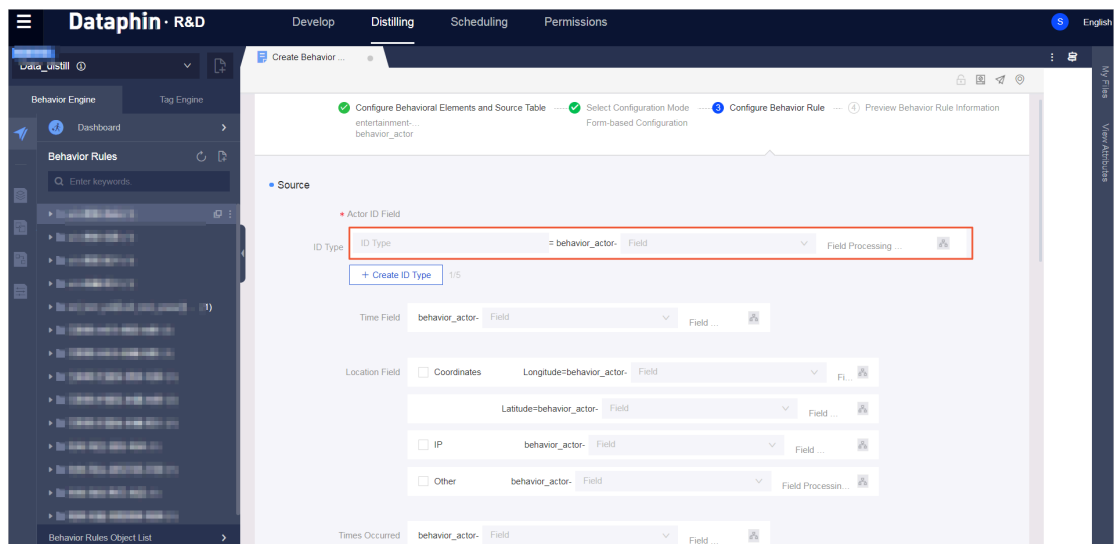


3. The code-based configuration mode requires you to write SQL code. Specify necessary table and field parameters in the SQL script.

Figure 9-18: Code-based configuration mode



- For form-based configuration mode, configure the following items:
  - The actor ID type (manually input) and the rules for identifying the actor ID field based on the source table.



- The rules for identifying the time when and location where the behavior occurs based on the source table.

• Source

\* Actor ID Field

ID Type ID Type = behavior\_actor- Field Field Processing ...

+ Create ID Type 1/5

Time Field behavior\_actor- Field Field ...

Location Field ☐ Coordinates Longitude=behavior\_actor- Field Field ...

Latitude=behavior\_actor- Field Field ...

☐ IP behavior\_actor- Field Field ...

☐ Other behavior\_actor- Field Field Processing...

Times Occurred behavior\_actor- Field Field ...

Duration behavior\_actor- Field Field ...

- The rules for identifying the object and object ID based on the source table.

• Objects & Object Attributes

\* Object Field behavior\_actor- Field Field ...

Object ID Field behavior\_actor- Field Field ...

\* Object Attribute Sources

Object Attribute Object Attribute = Source Table/Configure Object Attribute Association- Table Field

+ Create Object Attribute

Configure Object Attribute Associ...

Select an associated table. Load Associated Table Filter ...

Source Table

- The data source for the object attribute. You can select an associated attribute source table.

Object Attribute Sources

Object Attribute Object Attribute = Source Table/Configure Object Attribute Association- Table Field

[+ Create Object Attribute](#)

Configure Object Attribute Associ...

Select an associated table. Load Associated Table Filter ...

Source Table behavior\_actor Association Type Association Type Configure Object Attribute Ass...

Configure Object Attribute Association

Source Table behavior\_actor Field Association Condition = Field Associated with Object Attribute undefined Field

[+ Add Object Attribute Association](#) [Refresh](#)

When configuring the identification rules, you can enter fields from the source table or a SQL expression (field processing condition) based on fields from the source table.



#### Note:

- During behavioral data calculation, a globally unique ID is generated by joining the ID type and the ID field together. The globally unique ID is used to uniquely identify the actor during tagging. We do not recommend that you enter a number as the ID type. We recommend that you define a unified and fixed ID type, such as 'mobile phone number' or 'MAC address', to ensure the consistency of your defined ID information within the Dataphin system.
- When multiple IDs point to the same person, the relationship between the IDs cannot be entered into the system in the form of an inverted index. Therefore, we recommend that you process the ID data in advance and define a unique ID that identifies the person and enter this unique ID into the system. Then, all behavioral data of this person can be recorded under the defined unique ID.

4. • **Publish the behavior rule if the preceding configuration is correct, the rule information can be successfully previewed, and the rule information meets your requirements.**

Figure 9-19: Preview behavior rule information

Configure Object Attribute Associ...

Select an associated table. Load Associated Table Filter ...

Source Table behavior\_actor Association Type Association Type

Configure Object Attribute Ass...

Configure Object Attribute Association

Source Table behavior\_actor Field Association Condition = Field Associated with Object Attribute undefined

+ Add Object Attribute Association Refresh

Previous Publish Preview Behavior Rule Information

Figure 9-20: Publish a behavior rule

Develop Distilling Scheduling Permissions

Create Behavior ...

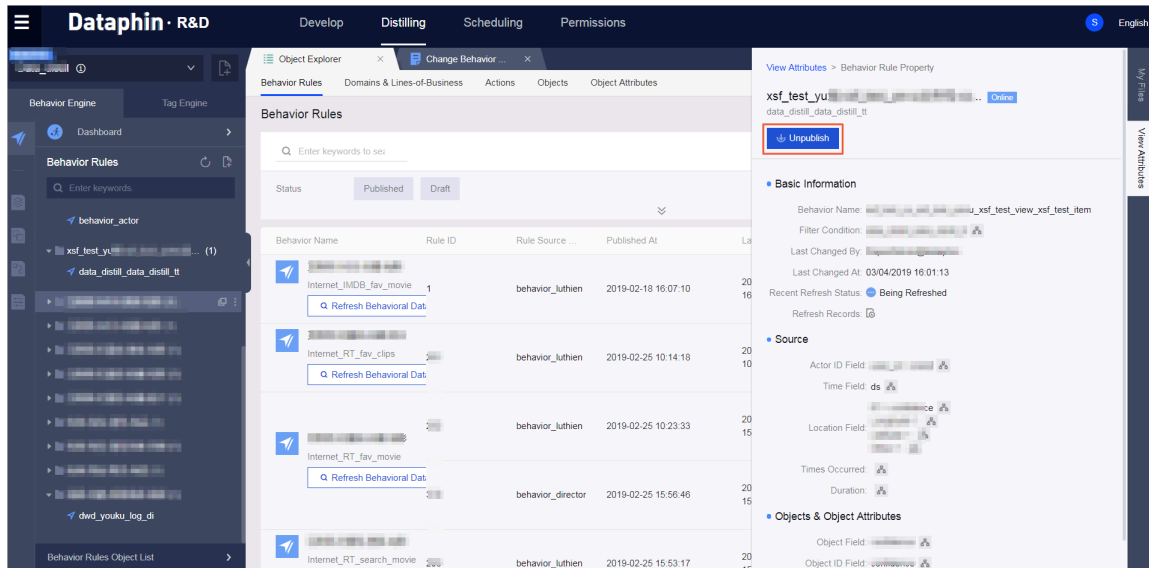
Configure Behavioral Elements and Source Table Select Configuration Mode Configure Behavior Rule Preview Behavior Rule Information

entertainment-... behavior\_actor Form-based Configuration Configured

Actor ID	Object Attributes	Location	Frequencies	Time	Duration (Seconds)
A: [redacted]	[redacted]	[redacted]	1	W	1

Previous Publish

- **If you want to modify a published behavior rule, make sure that no tags are being produced using this rule and unpublish the rule to edit it.**



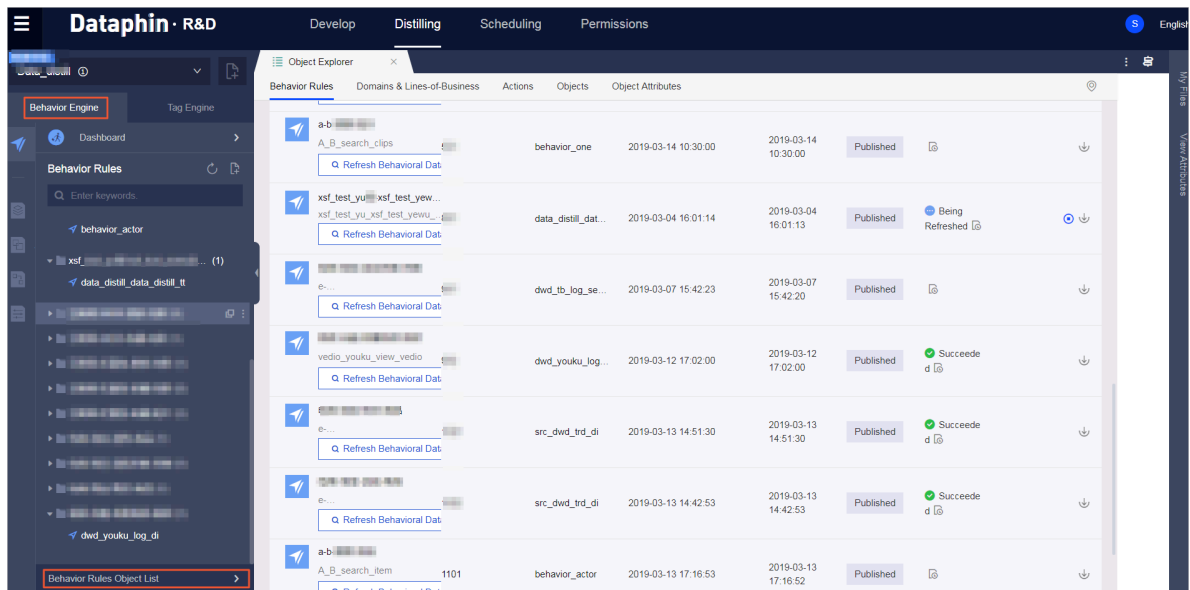
### 9.8.2.3 View behaviors

All objects created in the Behavior Engine are listed in the console. You can view and manage the objects in the list. You can also verify samples of the behavioral data that is generated based on each behavior rule.

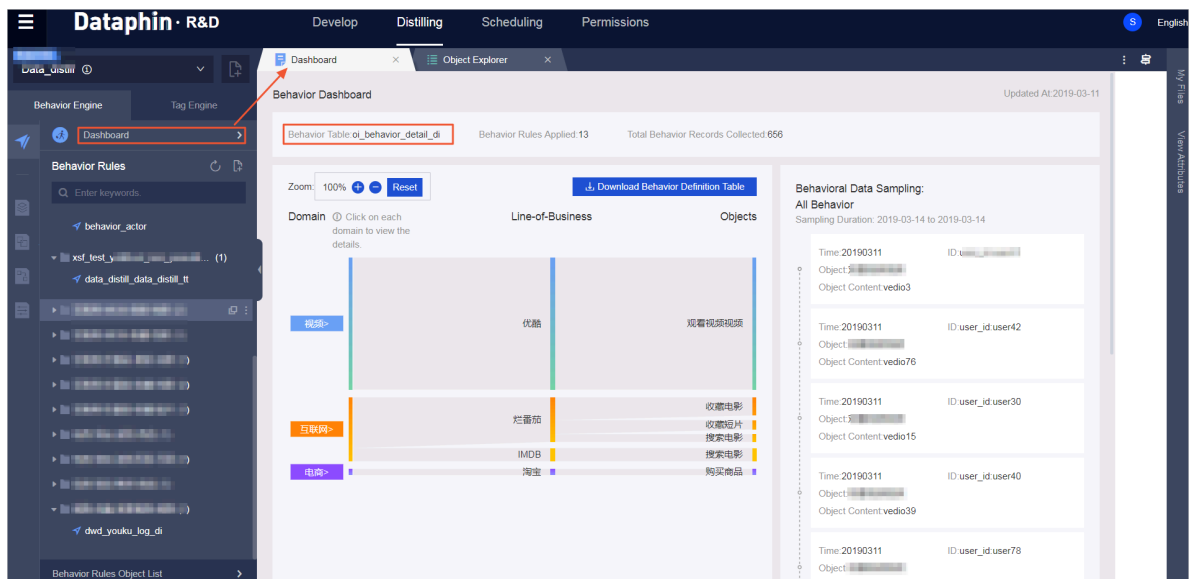
You can define standard behavior categories, behavioral elements, and their data sources through the definition and design of behavioral elements and behavior rules. For example, you can define behavior rules for the behavior of watching videos in a specific platform (entertainment behavior) and buying goods in a specific commercial district (shopping behavior). After all behavior rules are configured, the behavior rules (defined based on a domain, a line of business, an action, an object, and a source table) and rule IDs are automatically generated and categorized based on the behavioral elements. All behavioral data of the target objects are aggregated into a wide table to form a specific behavioral data report for reference.

Follow these steps to view behavior-related information:

## 1. View and manage your defined behavioral elements and behavior rules in the list.



## 2. A behavior table is a behavior wide table generated based on the behavioral data. It aggregates behavioral data and can be queried and used in the same way as other physical tables. The behavior dashboard allows you to easily view behavioral data, including the results of recent behavioral data sampling, the number of behavior records, and the behavior under specific actor IDs.



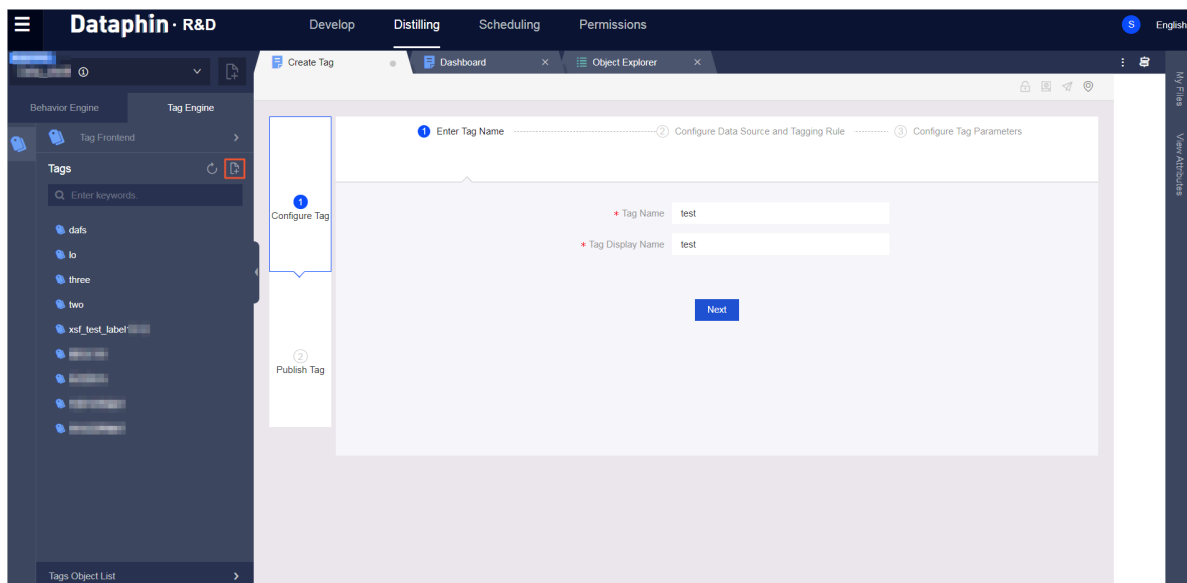
## 9.8.3 Tag Engine

The Tag Engine provides a graphical configuration console that allows for automatic tag data production and that allows you to manage (such as modify and unpublish) the tagging tasks under configuration or production.

### 9.8.3.1 Define tags

A graphical configuration console is provided for the configuration and creation of tags inferred from user behaviors. The tags can reflect the user preferences computed based on user behaviors and behavior weights, such as preferences for clothing styles and video types. In the console, you can set the tag name, data sources, and tagging rules, select the time decay mode, assign weights to behaviors, run tagging tests, view tag reports, and publish tags to run tagging tasks.

1. Create or edit tag data, including basic information such as the tag name, source behavioral data, and tagging rules.



- **Select the source behavioral data:** Select a behavior defined in the Behavior Engine. Select a time period based on the behavioral data freshness required for tagging, for example, last 30 days or last 90 days. In addition, define tag values based on the behavioral data source attributes and target object information.
- **The following tagging methods are supported:**
  - **Direct tagging:** For example, the gender tag female can be directly assigned to the actors who have viewed women's clothes in the last 30 days. You can use an object attribute as a tag value. For example, you can set an attribute

of the skincare products favorited in the last 30 days as a user preference tag value.

- **Mapping with object attribute:** You can map specific object attributes to a value and set this value as the tag value. For example, the moisturizing and nourishing attributes of the skincare products favorited in the last 30 days can be mapped to the preference for hydration. In addition, the distribution of object attributes is provided to help you determine the tagging method.

Change Tag: thr... Create Tag Dashboard Object Explorer

1 Enter Tag Name 2 Configure Data Source and Tagging Rule 3 Configure Tag Parameters

1 Configure Tag

2 Publish Tag

\* Data Source \* Filter Behavior a-b- \* Time Period 30

\* Tagging Rule \* Tagging Method Mapping with Obj... \* Tag Value chandi

+ Add

Mapping with Objec... Direct Tagging

Previous Next

Tag Value

2 Mapping Method

Field Distribution 产地

① Data on vertical bar chart is the pre-run data for the selected behavior, regardless of the specific filter conditions. The top 20 attributes with the highest percentage are shown in the chart.

Mapping Method ☒ Applied as Tag Value ☐ Set Mapping Rules ① When multiple rules are applied to the same behavior record, the first

+ Add

Cancel OK



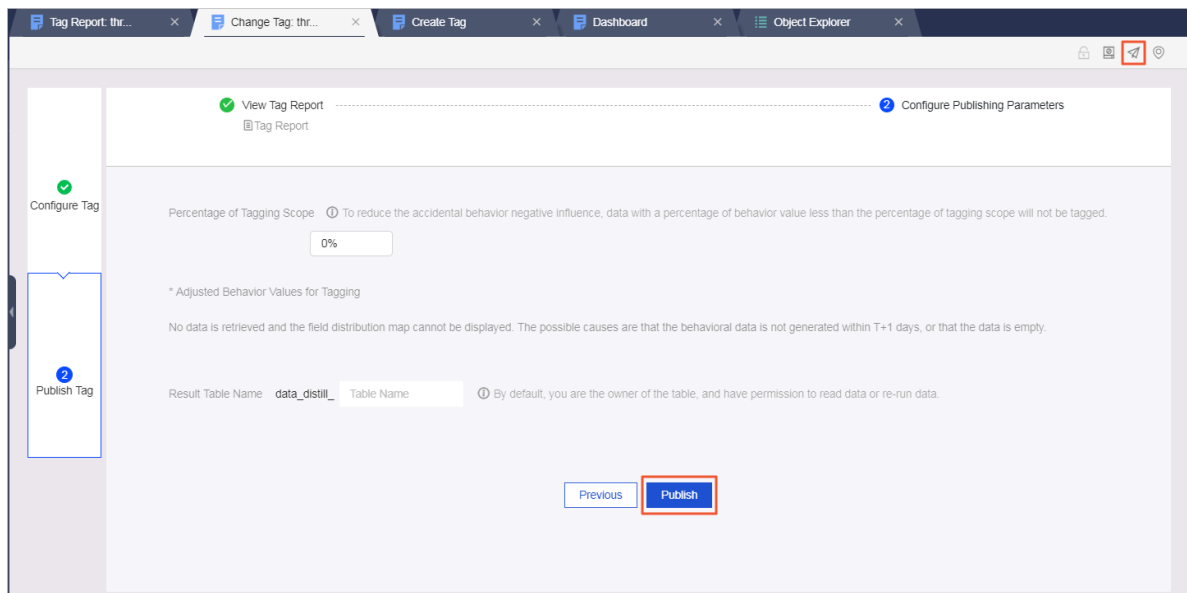
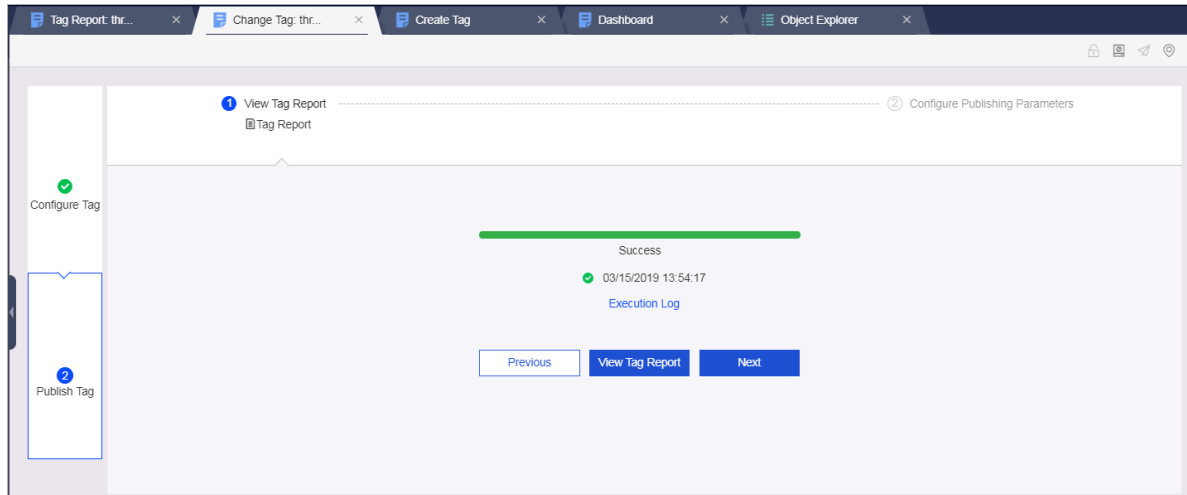
## 2. Configure the algorithm parameters of the tag, including the number of returned tag values.

- **Time decay mode:** You can configure the decay mode for each time segment (for example 7 days) of the time period during which the occurring behavior is studied (for example 90 days). In this example, there are 13 segments in total. A decay mode indicates how much the behavior that occurs during different time segments affects the final tag value. For example, if the behaviors of favoriting skincare products that occurred in and before the last 7 days do not affect summarizing a customer's skincare product preference, select the No Decay mode.
- **Set behavior weights:** The weights are used for normalization. Specify a weight for each type of behavior based on your business experience. Assign a higher weight to the behavior that affects the tagging result more. The system can also filter out the impact of incidental behaviors on the tagging result.

The screenshot displays the 'Configure Tag Parameters' step in the Dataphin interface. The interface is divided into three main sections: 'Returned Tag Value', 'Time Decay Mode', and 'Behavior Weight Ratio'. The 'Returned Tag Value' section has a dropdown menu set to '1'. The 'Time Decay Mode' section has a dropdown menu set to '7 Days'. The 'Time Decay Curve' section has three options: 'No Decay', 'Linear Decay', and 'Exponential Decay'. The 'Behavior Weight Ratio' section has a table with two rows, 'a-b' and 'c-d', and two columns, 'Weights', both set to '0.5'. The interface also includes a 'Previous' button and a 'Run' button at the bottom.

## 3. Run a tagging test and publish the tag. If the preceding configuration is correct, run a tagging test. You can view the tag report after the test is completed. If the tagging result meets your expectations, you can configure the tag publishing

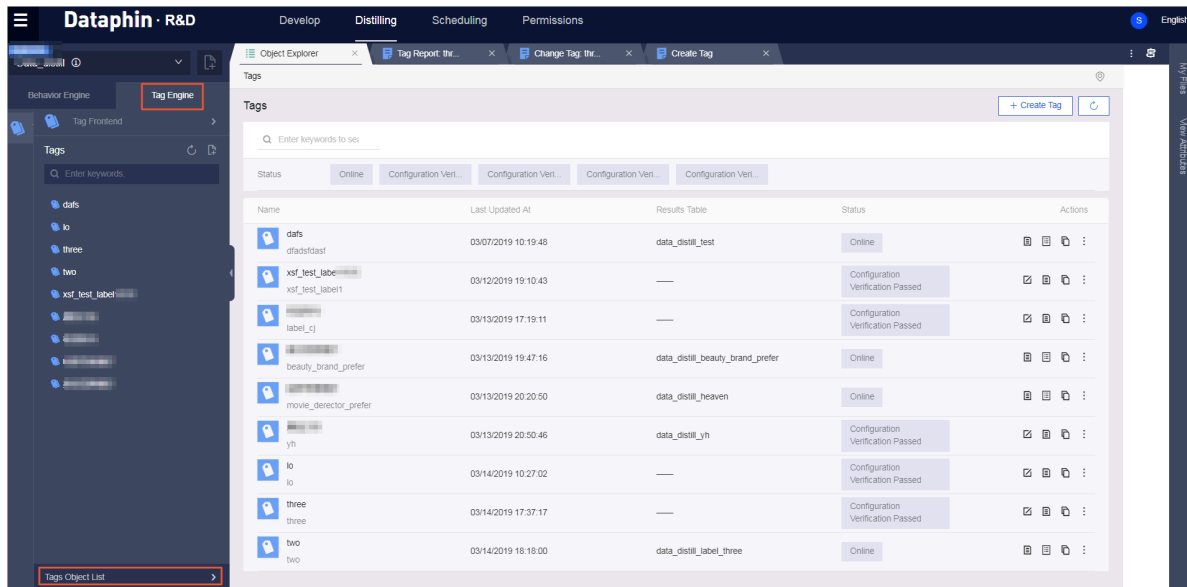
parameters and publish the tag to start recurring tagging tasks. If you want to modify a published tag, unpublish the rule to edit it.



### 9.8.3.2 View tags

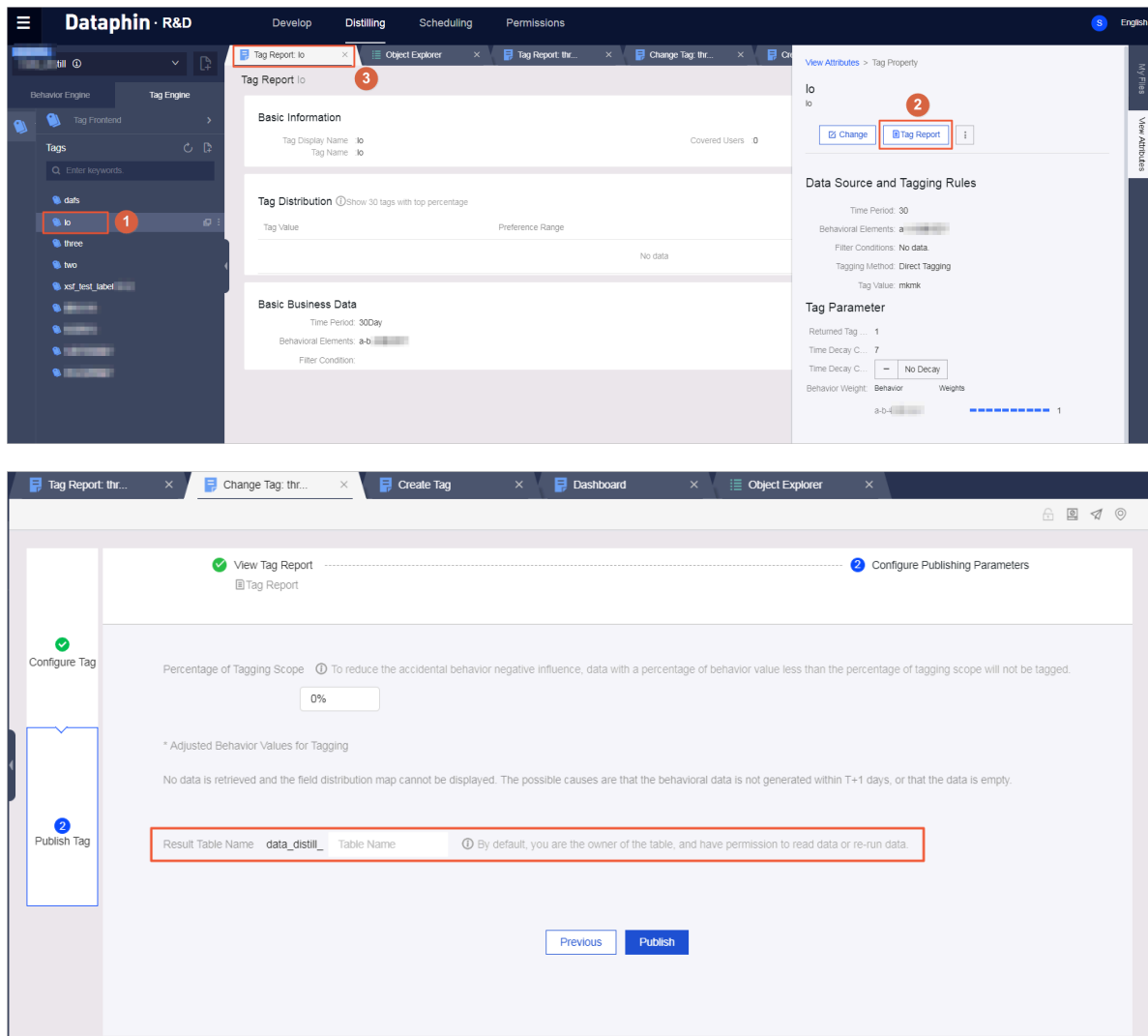
All tags defined in the Tag Engine are listed in the console. You can view and manage the tags in the list, including the tags being configured and published tags. You can also view the tag report for each tag.

#### 1. View and manage the defined tags in the list.



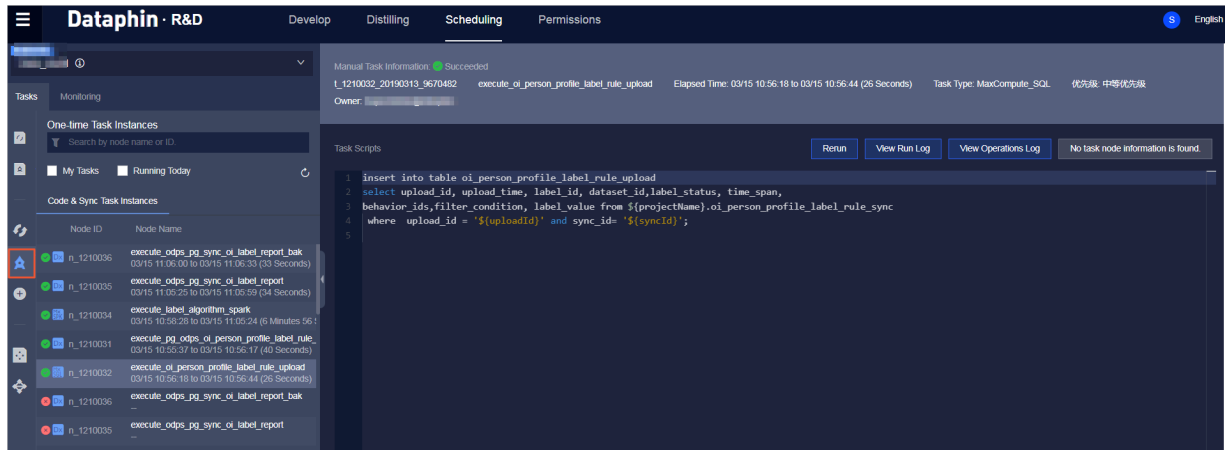
#### 2. View the detailed data reports of the defined tags for which a tagging test is successfully executed. The reports can help you learn about the tagging result. You can obtain the physical table with the result table name specified during

the configuration of publishing parameters, and query the tagging result in the table.



## 9.8.4 Manage data distillation tasks

After behavior rules and tags are published, the corresponding tasks are automatically generated. Members of the data distillation project can go to the scheduling center and manage tasks in the data distillation project. (You can find behavioral data production tasks or task instances on the One-time Tasks or One-time Task Instances page. Other tasks or task instances are listed on the Recurring Tasks or Recurring Task Instances page.)



Follow these steps to obtain a task:

- You can obtain a behavioral data production task using the refresh log of the relevant behavior rule. Click a task ID starting with t\_ to go to the corresponding operational log page. Find the node ID in the log and search for this ID to locate

the task on the Scheduling > One-time Tasks/One-time Task Instances page.  
Then, you can view and manage the task.

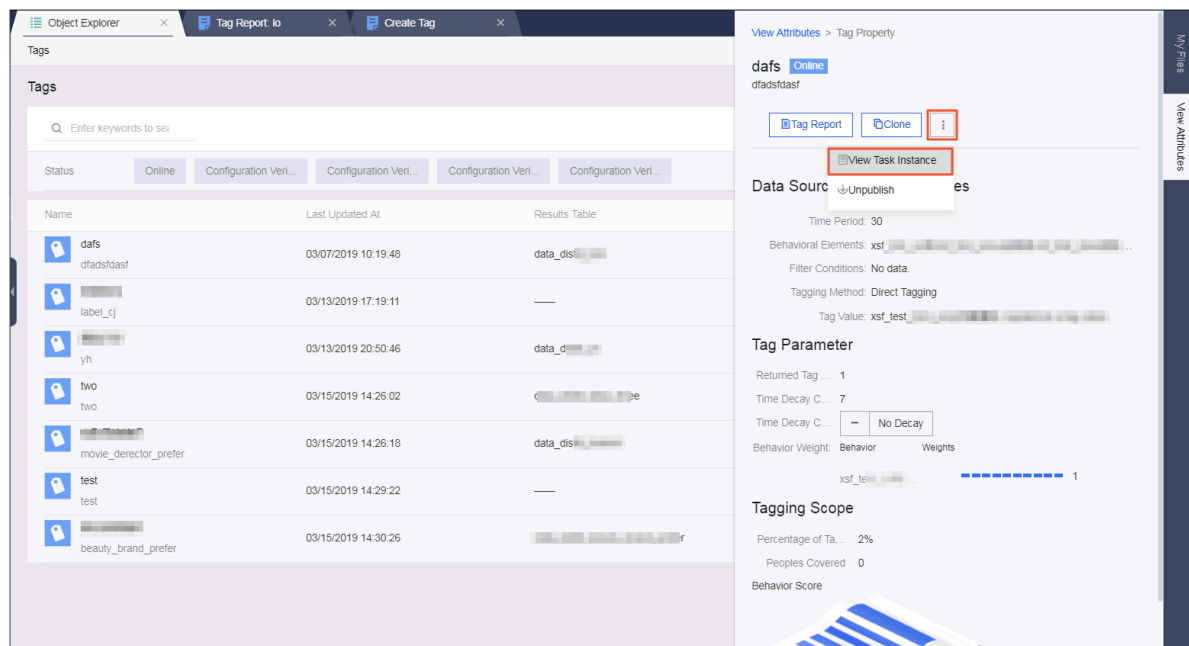
The screenshot shows the Dataphin interface with the 'Refresh Behavioral Data' task list. A modal window titled 'Refresh Records' is open, displaying a table of refresh operations. A dropdown menu is also visible, listing task IDs.

Operation Time	Operated By	Refresh Time	Status
2019-03-12 17:02:26		2019-03-12 17:02:26	Succeeded
2019-03-11		2019-03-11	Succeeded

The screenshot shows the Dataphin 'Operational Log' page. The log displays the execution details of a task, including timestamps, task ID, task type, and task parameters.

```
1 2019-03-12 17:17:40 ----- voldemort task initiating -----
2 2019-03-12 17:17:40 Node id: n_1210030
3 2019-03-12 17:17:40 Task id: t_1210030_20190301_8920425
4 2019-03-12 17:17:40 Task Type: NORMAL
5 2019-03-12 17:17:40 Taskrun id: tr_1210030_20190301_8860509
6 2019-03-12 17:17:40 Taskrun priority is MIDDLE
7 2019-03-12 17:17:40 Taskrun was due to execute at 2019-03-12 17:17:35
8 2019-03-12 17:17:40 Taskrun delay time is: 13138ms
9 2019-03-12 17:17:40 Current Taskrun has been dispatched to agent:
10 2019-03-12 17:17:40 Begin to execute DATAX task.
11 2019-03-12 17:17:40 Current task status: RUNNING
12 2019-03-12 17:17:40 -----
13 2019-03-12 17:17:40 List of task parameters:
14 2019-03-12 17:17:40 bizdate=20190311
15 2019-03-12 17:17:40 nodeid=n_1210030
16 2019-03-12 17:17:40 end_date=20190311
17 2019-03-12 17:17:40 yesterday=20190228
18 2019-03-12 17:17:40 datarefinedsid=
19 2019-03-12 17:17:40 begin_date=20190301
20 2019-03-12 17:17:40 tenantid=
21 2019-03-12 17:17:40 behavior_id=
22 2019-03-12 17:17:40 lastsevenday=20190222
23 2019-03-12 17:17:40 projectname=
24 2019-03-12 17:17:40 metasourceid=
25 2019-03-12 17:17:40 taskid=t_1210030_20190301_8920425
26 2019-03-12 17:17:40 -----
27 2019-03-12 17:17:40 -----
28
29 DataX (DATA-OPEN-SOURCE-3.0), From Alibaba !
30 Copyright (C) 2010-2017, Alibaba Group. All Rights Reserved.
31
32
33 2019-03-12 17:17:40.253 [main] INFO VMInfo - VMInfo# operatingSystem class => sun.management.OperatingSystemImpl
34 2019-03-12 17:17:40.260 [main] INFO Engine - the machine info =>
```

- A tagging task is a tag instance. Click View Instance on the right of a tag name to go to the corresponding task page. On this page, you can view and manage the task.



## 9.9 Scheduling center

### 9.9.1 Tasks

A task is an object that is published after you submit a code script and can automatically run at specified intervals or be manually triggered. You can configure the scheduling policy for a task. A task consists of one or more nodes.

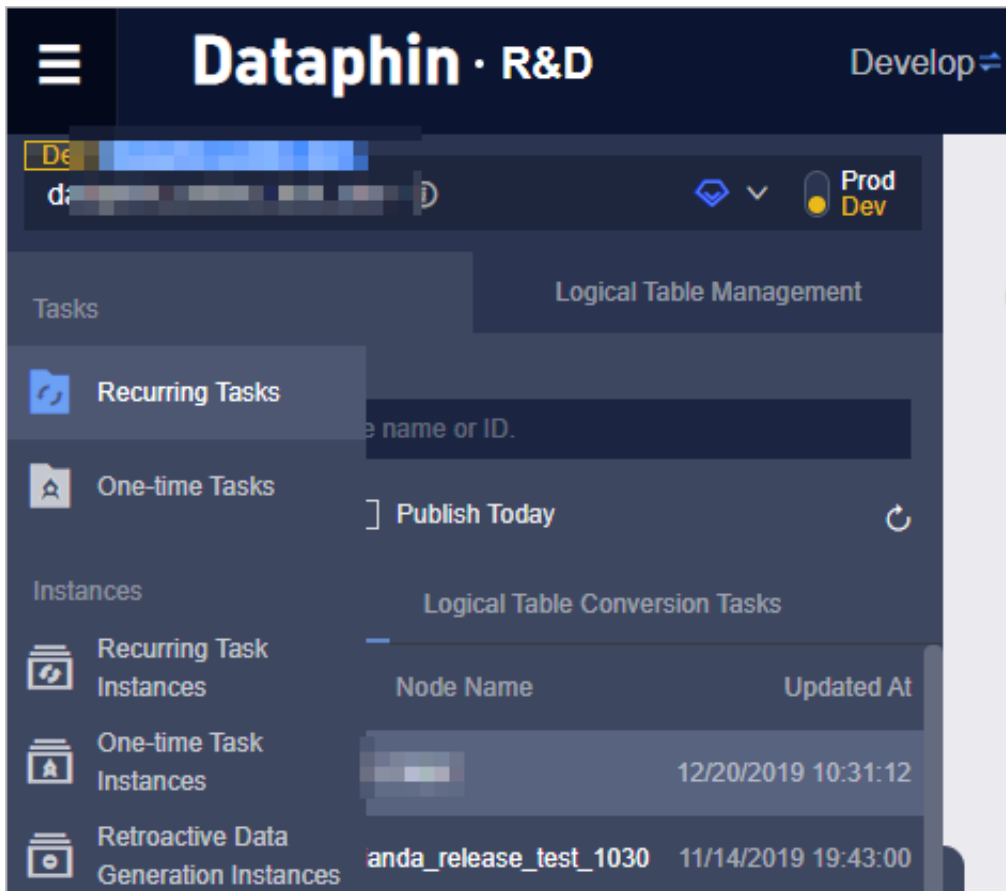
#### Context

After you build a data model or configure data distilling rules, Dataphin automatically generates and schedules tasks to produce data.

#### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click Scheduling and Management to go to the scheduling center.

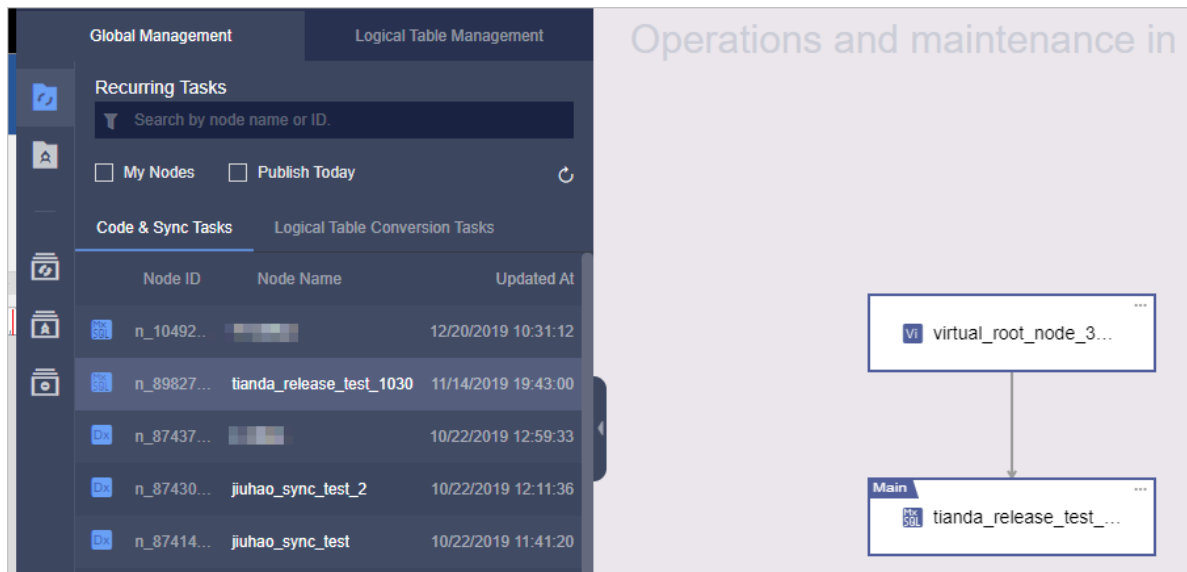
3. On the Scheduling tab, click **Recurring Tasks** or **One-time Tasks** on the left-side navigation submenu.



- **Recurring Tasks:** You can configure the recurrence for recurring nodes. The scheduling system of Dataphin runs recurring nodes at specified intervals. Corresponding recurring instances are generated each time recurring nodes are run.
  - **One-time Tasks:** The scheduling system of Dataphin does not automatically trigger one-time nodes. You can manually run such nodes as required. Corresponding one-time instances are generated each time one-time nodes are run.
4. On the Scheduling tab, select a project to view the nodes generated in the project. Click a node in the left-side node list. The node dependency directed acyclic graph (DAG) appears on the right. Right-click a node in the DAG and select a



required operation. For example, you can view the node script or edit node information.



## 9.9.2 Instances

A task node is scheduled to run by following its recurrence pattern. Every time a task node runs, an instance is generated. You can also right-click a node and select **Generate Retroactive Data** to create a task instance that generates retroactive data of specific dates. An instance is a runtime occurrence of a task and it has a specific running status.

### Context

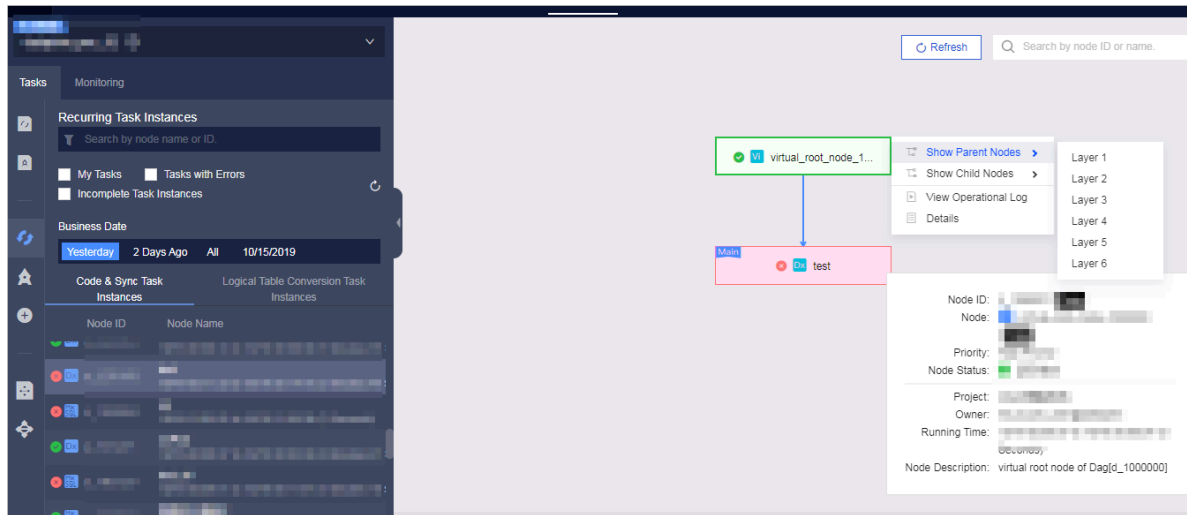
After you build a data model or configure data distilling rules, the system automatically generates and schedules tasks to produce data.

### Procedure

1. [Log on to the Dataphin console.](#)
2. On the Dataphin homepage, click **Scheduling and Management**.

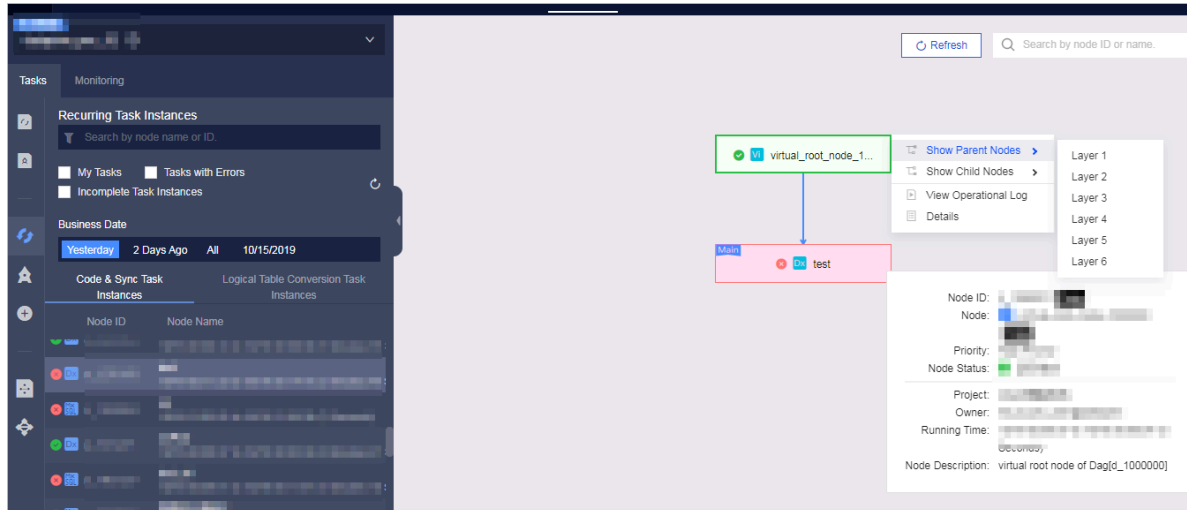
3. On the Scheduling page, select a project to view the instances generated in the project. You can select an instance to view its running logic graph and status, as shown in *Figure 9-21: Instance*.

Figure 9-21: Instance



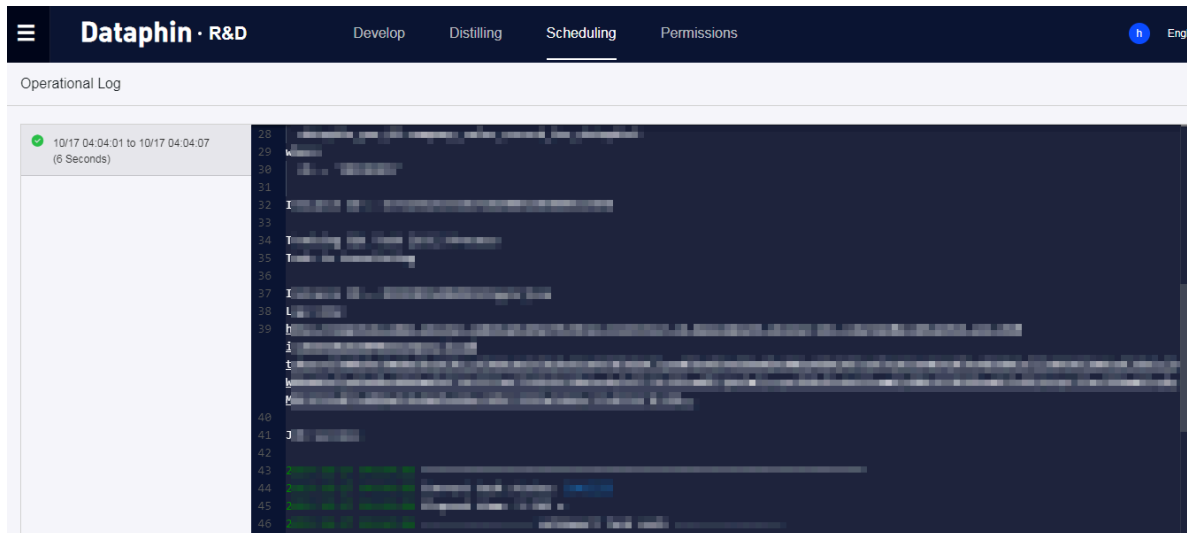
4. Select an instance and right-click it to manage the instance, as shown in [Figure 9-22: Managing an instance](#).

Figure 9-22: Managing an instance



You can open the operational log of an instance to view the code generated for the corresponding data model or data distilling task and task status, as shown in [Figure 9-23: Operational log](#).

Figure 9-23: Operational log



### 9.9.3 Logical table tasks

The Logical Table Tasks page displays the internal task relationship of each logical table.

#### Procedure

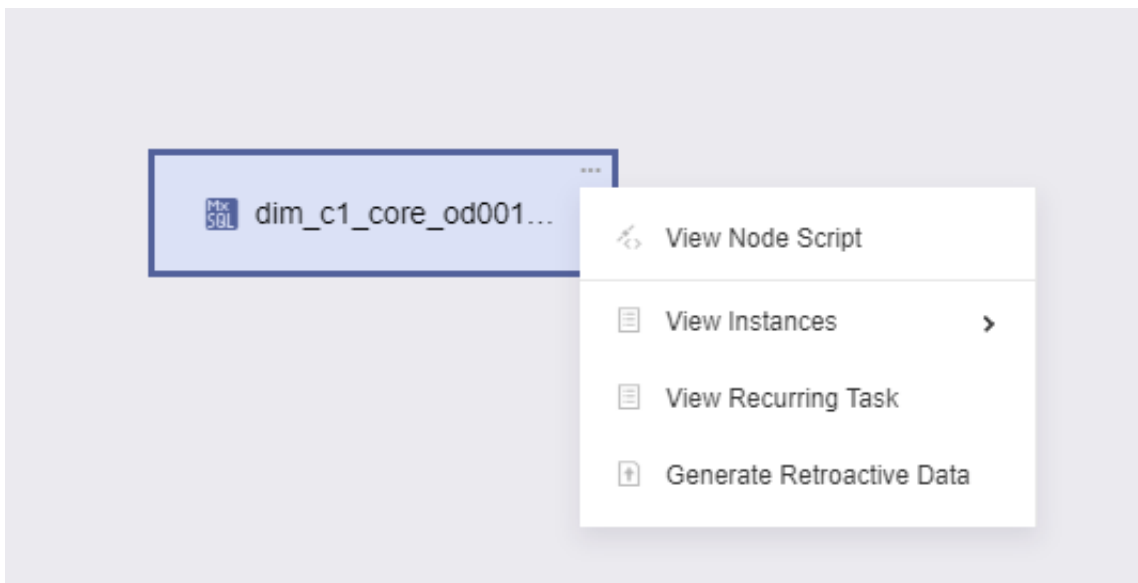
1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click **Scheduling and Management**.
3. On the Scheduling page, click the **Logical Table Tasks** submenu. You can move the pointer over an icon to expand its corresponding submenu. Click a logical table to view its internal task relationship.
4. In the task relationship graph, right-click a node to display the operations supported for the node, as shown in [Logical table](#).

**Note:**

You can perform the following operations on a node if you have the required permissions:

- View the node script
- View its task instances
- View the corresponding recurring task
- Generate retroactive data

Figure 9-24: Logical table



### 9.9.4 Logical table task instances

On the Logical Table Task Instances page, the left-side navigation pane is used to search for and display logical tables and the instances of their conversion task nodes. The search results are displayed in two layers. The first layer displays the searched logical table, and the second layer displays the task node instances of the

logical table. By default, the node type (displayed as an icon), node ID, node name, and instance run time are displayed. All logical tables are listed in alphabetical order. By default, the right-side workspace displays all task node instances of a logical table and status (such as running, succeeded, and failed). In the lower-left corner of the workspace, a thumbnail for the task node instances is displayed.

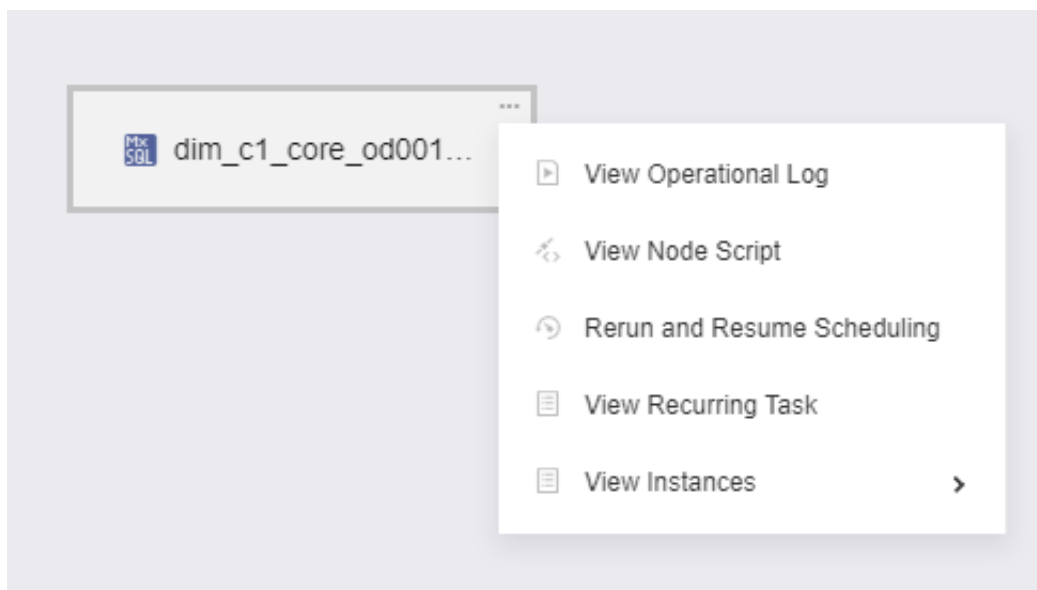
## Context

The Logical Table Task Instances page displays the task node instances of a logical table and status.

## Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click Scheduling and Management.
3. On the Scheduling page, click Logical Table Task Instances. On the page that appears, you can view the task node instances of a logical table and status.
4. In the right-side workspace, right-click an instance to display the operations supported for the instance. The operations include view the operational log, view the node script, rerun and resume scheduling, view the corresponding recurring task, and view instances, as shown in [Logical table](#).

Figure 9-25: Logical table



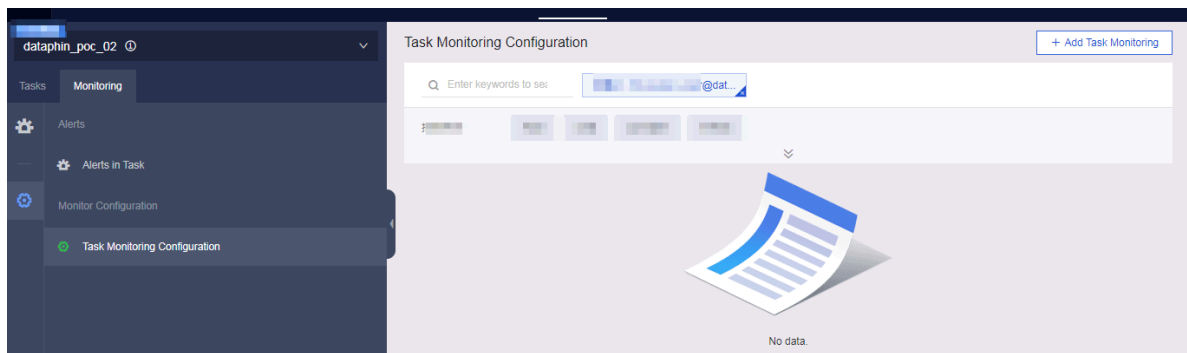
## 9.10 Monitoring and alerting

### 9.10.1 Task monitoring settings

#### 9.10.1.1 Create task monitoring settings

You can configure task monitoring to monitor the running status of published tasks. This helps you know the tasks status in real time.

1. Choose **R&D > Scheduling > Monitoring > Task Monitoring Configuration** to go to the Task Monitoring Configuration page.



2. Click **Add Task Monitoring** in the upper-right corner. A dialog box appears, as shown in the following figure.

The 'Add Task Monitoring' dialog box contains the following fields and options:

- Select Task:** A dropdown menu with the text 'Select Task'.
- Alert Cause:** Three radio buttons: 'Error', 'Complete', and 'Incomplete'. To the right is a time picker set to '00:00'.
- Timeout:** A radio button labeled 'Timeout' followed by a numeric input '0' and the unit 'Minutes'.
- Recipient:** Two radio buttons: 'Owner' (selected) and 'Custom Recipients'. To the right is a 'Select Recipients' button.
- Notification Method:** Two checkboxes: 'Email' and 'SMS'.

At the bottom right are 'Cancel' and 'OK' buttons.

### 3. Configure the fields in the dialog box.

- **Select Task:** You can select one or more tasks from the drop-down list for monitoring.
- **Alert Cause:** You can select one of the following four options, including Error, Complete, Incomplete (until a certain time point), and Timeout (of specified minutes).
- **Recipient:** You can set the task owner as the alert recipient. Alternatively, you can select a maximum of three recipients from the drop-down list as custom recipients.
- **Notification Method:** Only emails and SMS are supported.

### 4. Click OK.



#### Note:

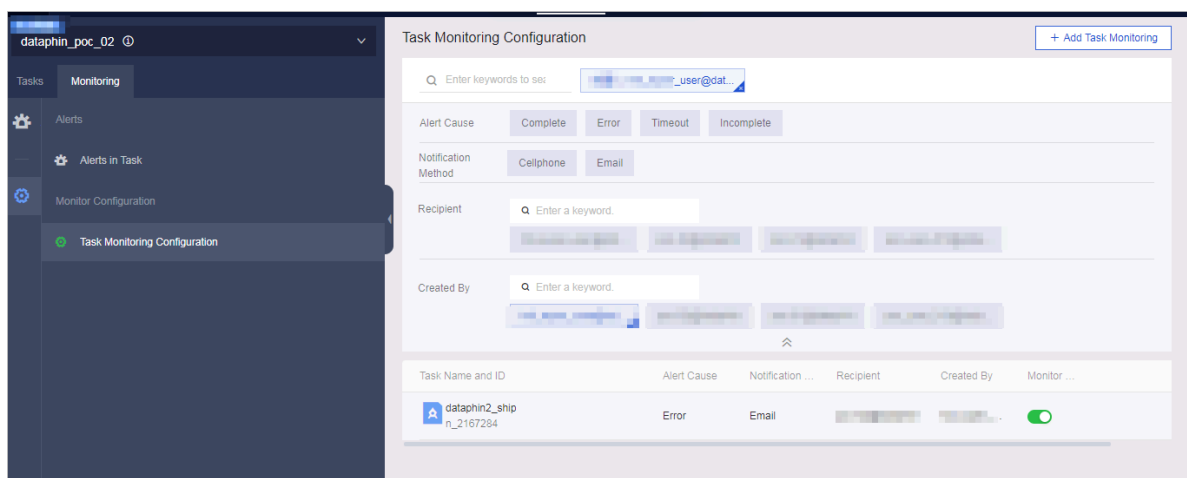
The monitoring service is in the beta phase. Configuring contact information and pushing alert notifications are not supported.

## 9.10.1.2 Manage task monitoring settings

This topic describes how to search for and view existing task monitoring settings, enable and disable task monitoring, and modify and delete task monitoring settings.

Search for and view existing task monitoring settings

1. Choose **R&D > Scheduling > Monitoring > Task Monitoring Configuration** to go to the Task Monitoring Configuration page.



## 2. The preceding figure shows the Task Monitoring Configuration page.

- In the search box at the top of the workspace, enter keywords to search for task monitoring settings. Click the drop-down arrow to expand the filter settings. You can filter the search results by selecting the alert cause, notification method, recipient, and monitoring setting creator.
- The search result is shown below the filter settings. The information displayed includes task name and ID, alert cause, notification method, recipient, and monitoring setting creator.

### Enable and disable task monitoring

In the search result section, you can enable or disable monitoring for a specific task by clicking Monitor Switch.

### Modify and delete task monitoring settings

In the search result section, click Change in the Actions column. In the dialog box that appears, you can modify the alert cause, recipient, and notification method for a task.

Add Task Monitoring ×

\* Select Task:

\* Alert Cause: ☐ Error ☐ Complete ☐ Incomplete  ⌚

☐ Timeout  Minutes

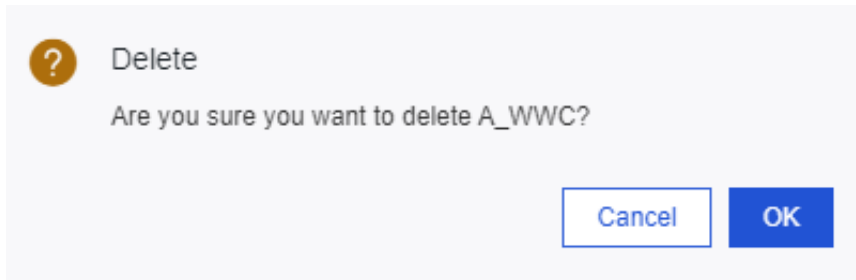
\* Recipient: ☒ Owner ☐ Custom Recipients

\* Notification Method: ☐ Email ☐ SMS

Cancel OK

Click Delete in the Actions column. A confirmation message appears, confirming whether you want to delete the task monitoring configuration.



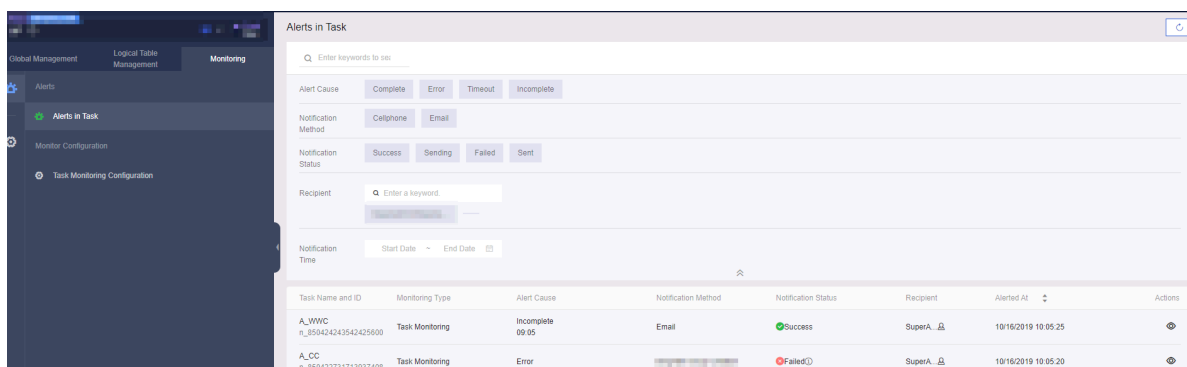


### 9.10.2 Alert records

When a monitored task triggers an alert, the system pushes an alert notification through the notification channels you specify in your alert rule. An alert record appears on the Alerts in Task page. You can view historical alerts on this page.

Search for and view alert records

1. Choose R&D > Scheduling > Monitoring > Alerts in Task, as shown in the following figure.



2. In the search box at the top of the workspace, enter keywords to search for alert records. You can also expand the filter settings under the search box to filter results. You can filter the search results by selecting filter conditions such as the alert cause, notification method, and notification status.



#### Note:

In the Sending Status column, the Sent status indicates that an alert notification has been sent but no delivery receipt is returned. The Success status indicates that an alert notification has been delivered to the recipient.

3. The alert record section displays task name and ID, monitoring type, alert cause, notification method, notification status, recipient, and alert notification time.

By default, alert records are displayed in descending order of notification time (from latest to earliest).

The screenshot shows the 'Alerts in Task' interface. At the top, there is a search bar with the placeholder 'Enter keywords to search'. Below the search bar are filter buttons: 'Complete', 'Error', 'Timeout', and 'Incomplete'. The main table has columns: 'Task Name', 'Monitoring Type', 'Alert Cause', 'Notification Method', 'Sending Status', 'Recipient', 'Alerted At', and 'Alert Details'. The table contains several rows of alert records. One row is highlighted, and an 'Alert Details' modal is open over it. The modal displays the following information: Task ID: n\_3143427, Alerted At: 10/16/2019 16:00:12, Alert Information (a blurred screenshot), and Created By: poc-02@dataphin.



#### Note:

If you have not configured the contact information for the notification recipient or the information is inaccurate, the Failed notification status is displayed. To view the cause of the failure, you can move the pointer over the information icon next to the notification status.

View alert details

In the alert record section, click **Alert Details** in the **Actions** column to view the alert details of a task.

Limits

The following describes the maximum number of notifications that can be sent by using each type of notification channel:

- **SMS:** A maximum of 100 messages for each tenant per day.
- **Emails:** A maximum of 2,000 emails for each tenant per day.
- **Voice messages:** A maximum of 100 voice messages for each tenant per day. You are billed for the length of each voice message. Voice messages that are shorter than one minute is charged for one minute.

**Note:**

Currently, the monitoring service is in the beta phase. Configuring contact information and pushing alert notifications are not supported.

## 9.11 Data assets

### 9.11.1 Overview

After data is collected, integrated, and processed during data development, you can manage the data on the Data Assets page in a systematic way.

Dataphin allows you to take inventory of and assess the data assets across your enterprise based on the standards and methodology of enterprise data asset management, including:

- Automatically extracts and analyzes metadata, and creates an overview of data assets. This allows managers to understand the value of data assets.
- Provides an end-to-end inventory check and analysis of computing, storage, security, and applications during data production. This helps you detect problems, propose and implement governance optimization solutions, reduce costs, and improve efficiency for data. You can go to the Data Assets page to view the data modeling results and data table details.

### 9.11.2 Asset overview

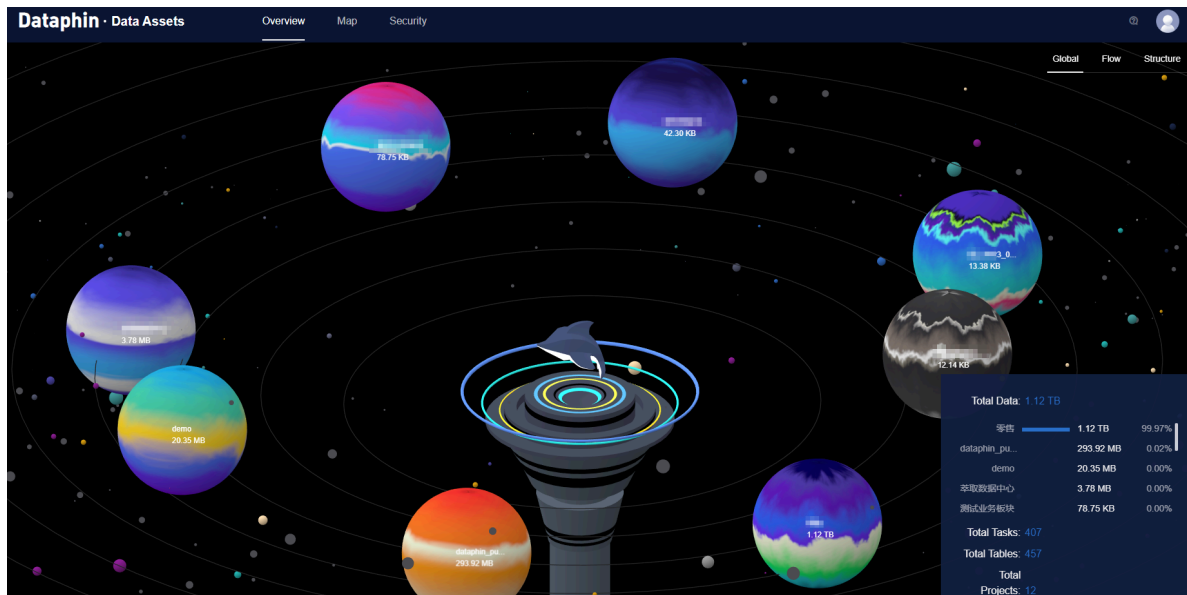
#### 9.11.2.1 Global mode

The Global mode displays business units with a large data size in the form of planets and their respective data sizes.

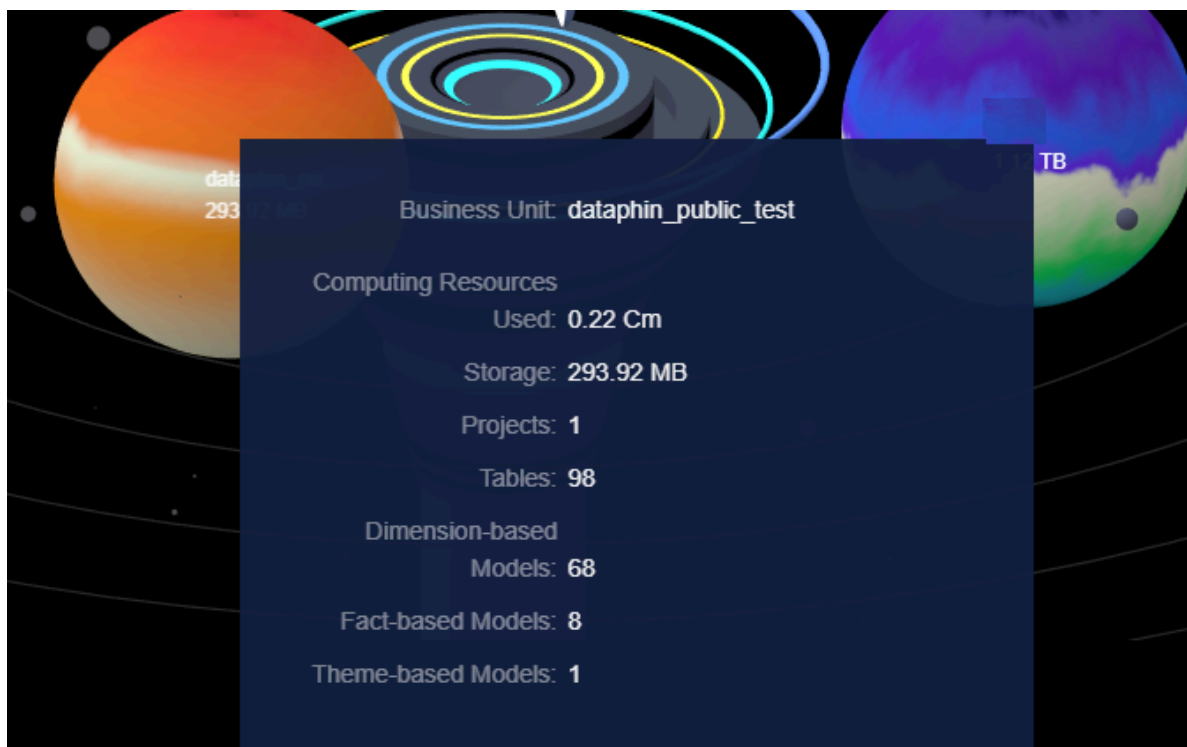
#### Procedure

1. [Log on to the Dataphin console.](#)
2. On the Dataphin homepage, click Data Assets in the top navigation bar to go to the Data Assets page.
3. On the Overview tab, click Global in the upper-right corner. In the lower-right corner, this page displays the total number of tasks, tables, and projects in

Dataphin. It also displays the rankings of business units with a large data size, including the business unit name, data size, and proportion to the total data size.



4. Move the pointer over a planet to view the information about the business unit, including the used computing resources, storage size, number of projects, and number of tables.

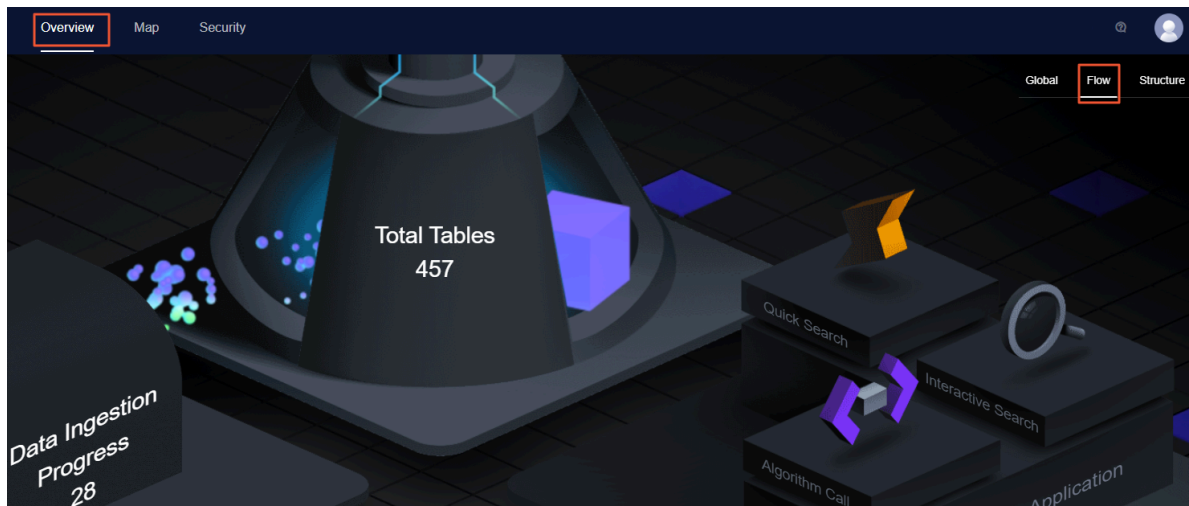


### 9.11.2.2 Flow mode

The Flow mode displays the entire process of data ingestion, integration, and output. This reveals the underlying potential capabilities of Data Mid-End.

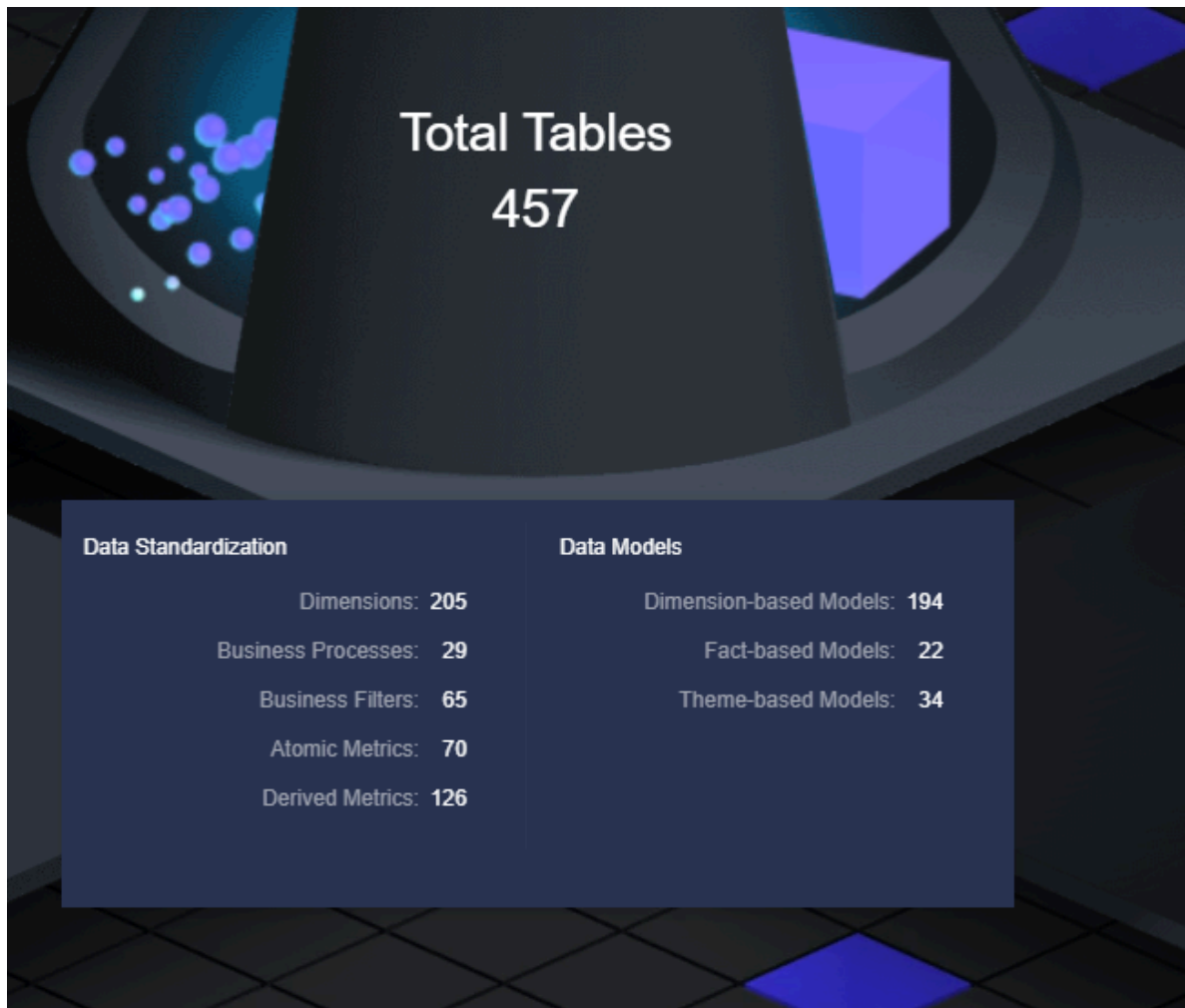
#### Procedure

1. [Log on to the Dataphin console.](#)
2. On the Dataphin homepage, click Data Assets in the top navigation bar to go to the Data Assets page.
3. On the Overview tab, click Flow in the upper-right corner. This page displays the data ingestion progress, the total number of tables, and data applications such as data query in a visual view.



4. Move the pointer over Data Ingestion Progress to view the number of data sources after deduplication, five data sources with the largest data input, and five data sources with the least data input.

5. Move the pointer over Total Tables to view the statistics about tables from the perspectives of data standardization and data models.



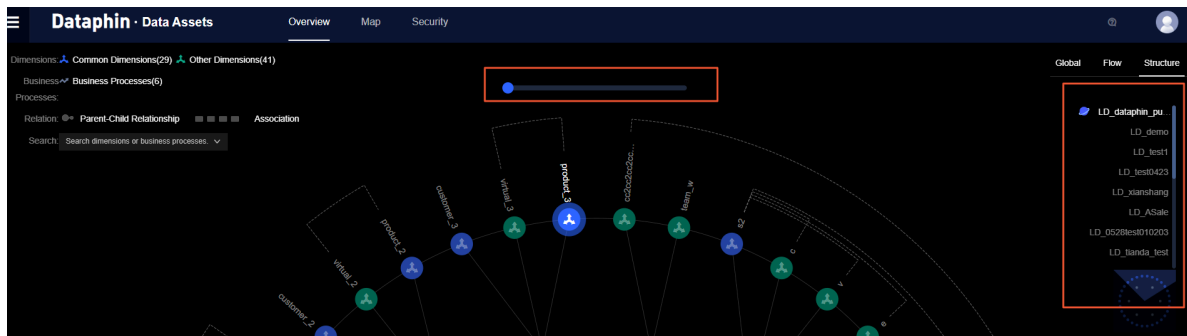
### 9.11.2.3 Structure mode

The Structure mode displays components in different shapes to represent business entities and uses lines of different styles to represent relationships between these entities. This mode clearly shows the structure of the data for a business unit.

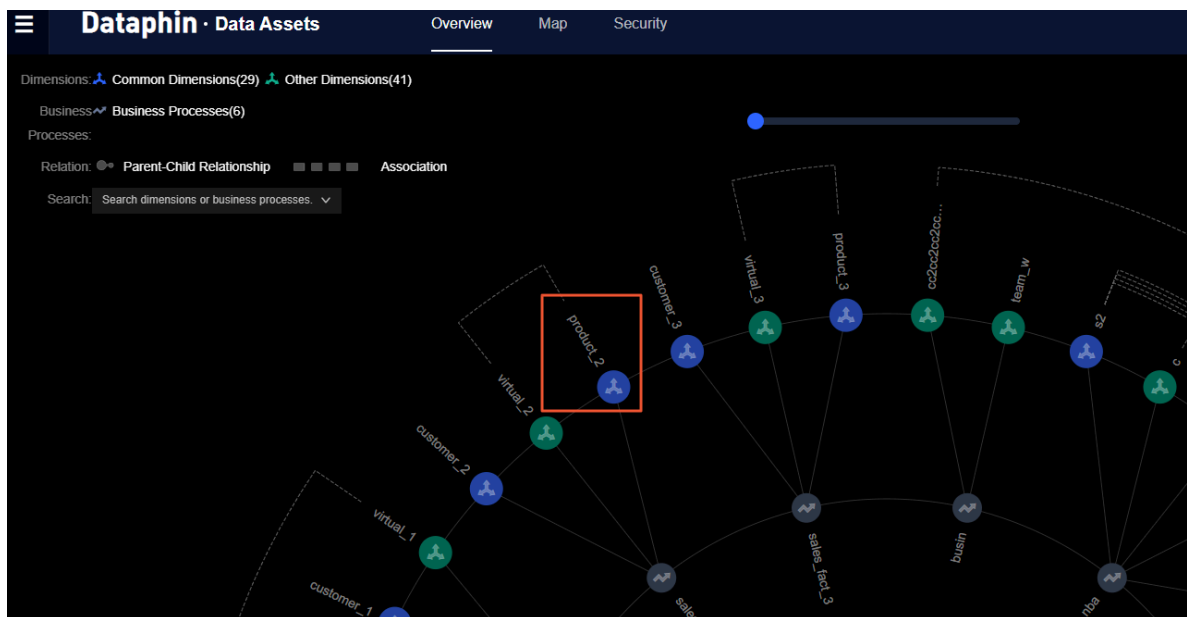
#### Procedure

1. [Log on to the Dataphin console.](#)
2. On the Dataphin homepage, click Data Assets in the top navigation bar to go to the Data Assets page.
3. On the Overview tab, click Structure in the upper-right corner. On the page that appears, click a business unit in the upper-right corner to display all the dimensions, business processes, and their relationships under this business

unit. You can drag the top progress bar to rotate and display other dimensions and business processes of the selected business unit.



4. Click a dimension, for example, the stock dimension. The dimensions and business processes related to this dimension are highlighted. Click a business process. The associated dimensions are highlighted. You can also search for a dimension or business process in the search box in the upper-left corner to view its associated objects.



### 9.11.3 Map

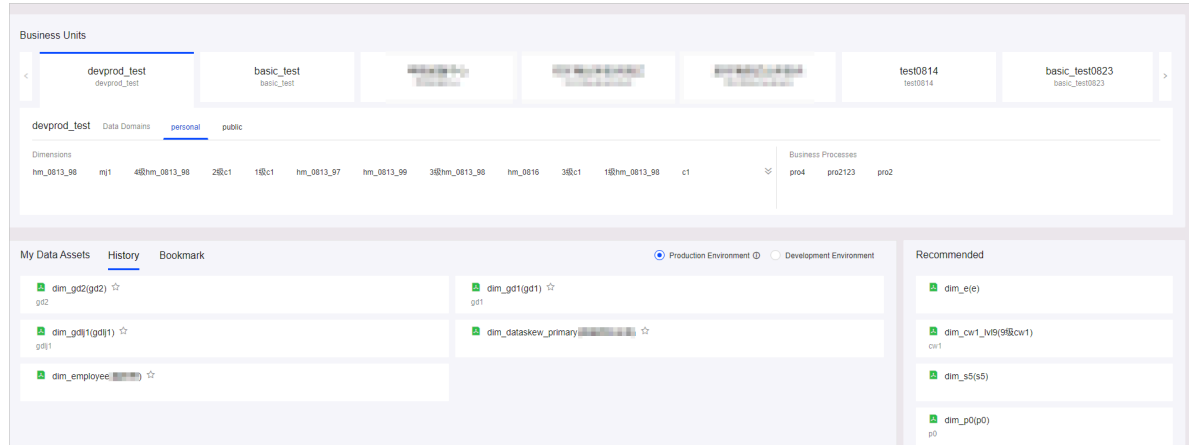
Dataphin can provide a data catalog for you to inventory the data assets that are created through standardized data modeling. The data catalog enables you to quickly find required data.

#### Procedure

1. Go to the Data Assets page.
2. Click Map.

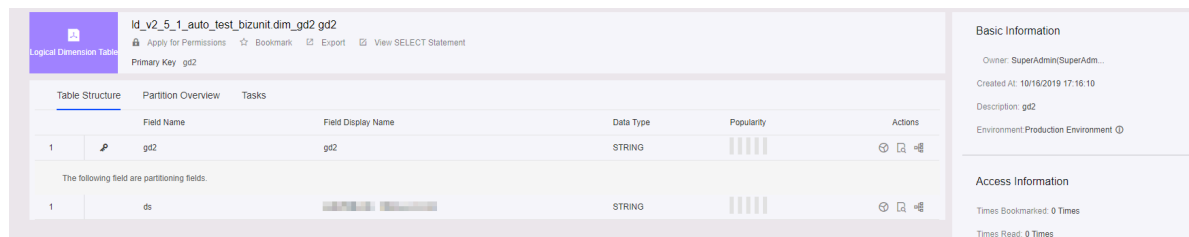
3. You can search or select data categories for specific tables, as shown in [Figure 9-26: Table](#).

Figure 9-26: Table



4. Click a table name to view the table details, including the table structure and metadata, as shown in [Figure 9-27: Metadata](#).

Figure 9-27: Metadata



## 9.11.4 Administration

All data resources are created in projects. To ensure secure use of data, you must apply for permission to use specific data. In your application, you must specify the fields you want to use. You can only use the specified data after your application is approved.

### Procedure

1. [Log on to the Dataphin console](#). On the Dataphin homepage, click Data Assets in the top navigation bar to go to the Data Assets page.
2. Click Map in the top navigation bar to go to the asset map homepage.



### 3. Enter keywords in the search bar to search for the table you want to use.

**Dataphin · Data Assets** Overview Map Administration

Permissions  
Permissions  
Permission Request  
My Permissions

Permission Request

Type: Logical Table

Business Unit: All

Permissions For: Select a request content.

Permission Type: Query

Account Type: Personal Account System Account

Validity Period: 10/16/2019 ~ 11/15/2019

Quick Select Days: 30 90 180 365

Reason:

Submit

### 4. In the list of search results, click the required table to go to the details page of the table.

### 5. On the table details page, click Apply for Permissions.

**Dataphin · Data Assets** Overview Map Administration

Permissions  
Permissions  
Permission Request  
My Permissions

Permissions

Submitted Requests My Approvals

Type: All Status: All Submitted At: Start Date ~ End Date

Search by request permission.

Request ID	Permissions For	Type	Permission Type	Permission Ownership	Validity Period	Reason	Submitted At	Approval Status	Approver	Actions
	demo.dim_...	Physical ...	Query	System Account	Permanent		10/12/2019 09:36:47	Approved	hdl_supe_r_user...	<a href="#">Details</a>
	demo.ods_...	Physical ...	Query	System Account	Permanent		10/12/2019 09:33:42	Approved	hdl_supe_r_user...	<a href="#">Details</a>
	demo.dim_...	Physical ...	Query	System Account	Permanent	sdd	10/11/2019 19:15:07	Approved	hdl_supe_r_user...	<a href="#">Details</a>

6. On the Permission Requests page that appears, specify the following information to submit a permission request.

The screenshot shows the 'Permission Request' form in the Dataphin interface. The form is titled 'Permission Request' and is part of the 'Data Assets' section. It includes the following fields and options:

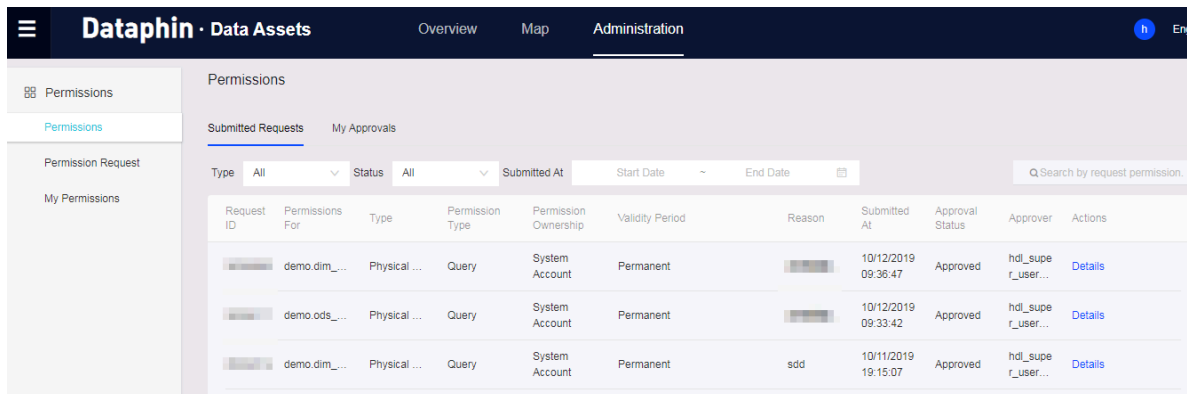
- Type:** Logical Table (dropdown menu)
- Business Unit:** All (dropdown menu)
- Permissions For:** Select a request content. (text input with a search icon)
- Permission Type:** Query (radio button selected)
- Account Type:** Personal Account (radio button selected), System Account (radio button)
- Validity Period:** 10/16/2019 ~ 11/15/2019 (date range selector)
- Quick Select Days:** 30, 90, 180, 365 (checkboxes)
- Reason:** (text input)
- Submit:** (button)

Field	Description
Type	Dataphin displays the type of table you want to use by default . The supported types include logical table, physical table, and data source.
Business Unit	Dataphin displays the business unit to which the requested table belongs by default.
Permissions For	Dataphin can display the metadata of table fields. When you are applying for permissions, you must follow the principle of least privilege and only select the fields you need to access .
Permission Type	The supported permission type is the query permission.
Account Type	<p>When you are applying for permissions, you need to select the type of account to which the requested permission will be granted. You can select either of the following two account types.</p> <ul style="list-style-type: none"><li>• <b>Personal account:</b> This type of account belongs to a specific user and can be used to perform operations such as development and query.</li><li>• <b>System account:</b> This type of account belongs to a specific project and is administered by the super administrator or project administrator. This type of account can be used to run one-time and recurring tasks.</li></ul>
Validity Period	Dataphin provides multiple options for permission validity period. You can customize a date range or select 30 days, 90 days, 180 days, or 365 days as the validity period.

Field	Description
Reason	You can describe the purposes for which you intend to use the requested permissions. The approver can determine whether to grant you the permissions based on the description.

7. Click Submit.

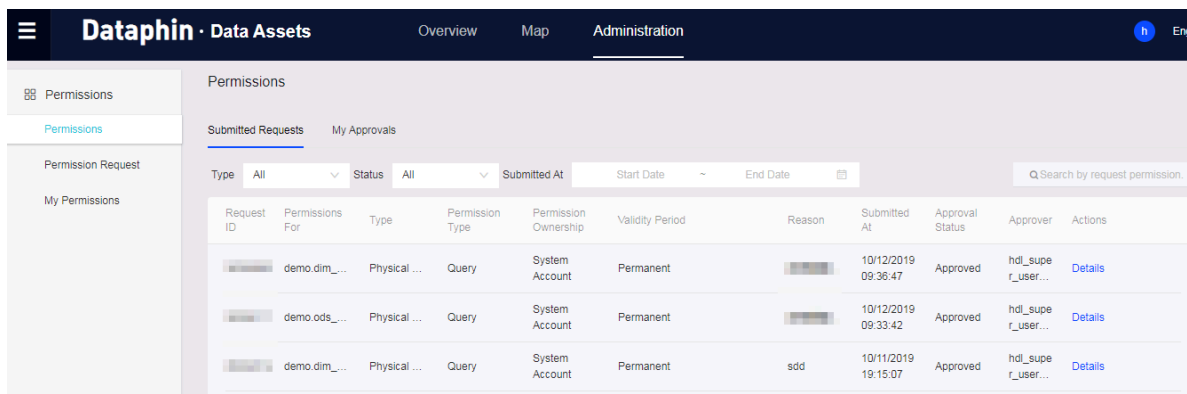
8. You can view the submitted permission request on the Permissions > Submitted Requests tab.



Request ID	Permissions For	Type	Permission Type	Permission Ownership	Validity Period	Reason	Submitted At	Approval Status	Approver	Actions
demo.dim_...	Physical ...	Query	System Account	Permanent			10/12/2019 09:36:47	Approved	hdl_supe_r_user...	<a href="#">Details</a>
demo.ods_...	Physical ...	Query	System Account	Permanent			10/12/2019 09:33:42	Approved	hdl_supe_r_user...	<a href="#">Details</a>
demo.dim_...	Physical ...	Query	System Account	Permanent		sdd	10/11/2019 19:15:07	Approved	hdl_supe_r_user...	<a href="#">Details</a>

Action	Description
Details	You can click Details for a request that is in the Reviewing status to view the request details.
Cancel	You can click Cancel for a request that is in the Reviewing status to cancel the request.

9. The permission request approver can view your request and other requests that require approval on the Permissions > My Approvals tab. The approver can approve or reject a request. After your request is approved, you can access the requested data resources.



Request ID	Permissions For	Type	Permission Type	Permission Ownership	Validity Period	Reason	Submitted At	Approval Status	Approver	Actions
demo.dim_...	Physical ...	Query	System Account	Permanent			10/12/2019 09:36:47	Approved	hdl_supe_r_user...	<a href="#">Details</a>
demo.ods_...	Physical ...	Query	System Account	Permanent			10/12/2019 09:33:42	Approved	hdl_supe_r_user...	<a href="#">Details</a>
demo.dim_...	Physical ...	Query	System Account	Permanent		sdd	10/11/2019 19:15:07	Approved	hdl_supe_r_user...	<a href="#">Details</a>

## 9.12 Theme-based data service

### 9.12.1 Ad hoc query

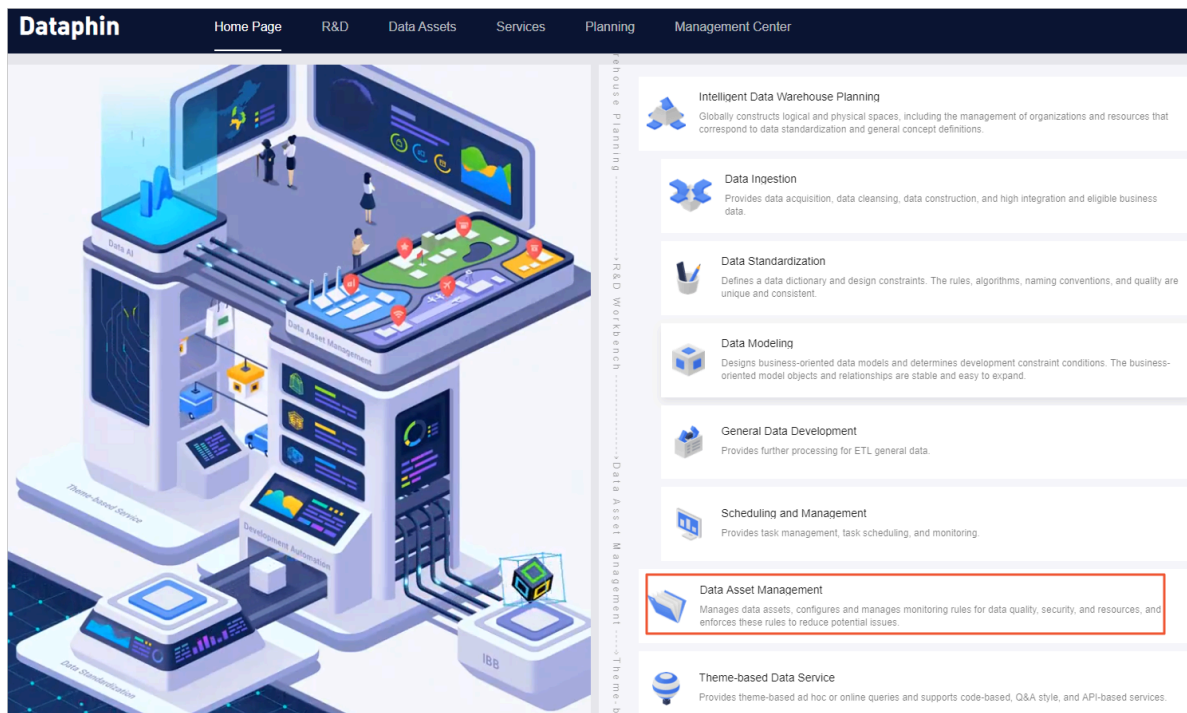
Dataphin provides the ad hoc query feature for you to query data.

#### Context

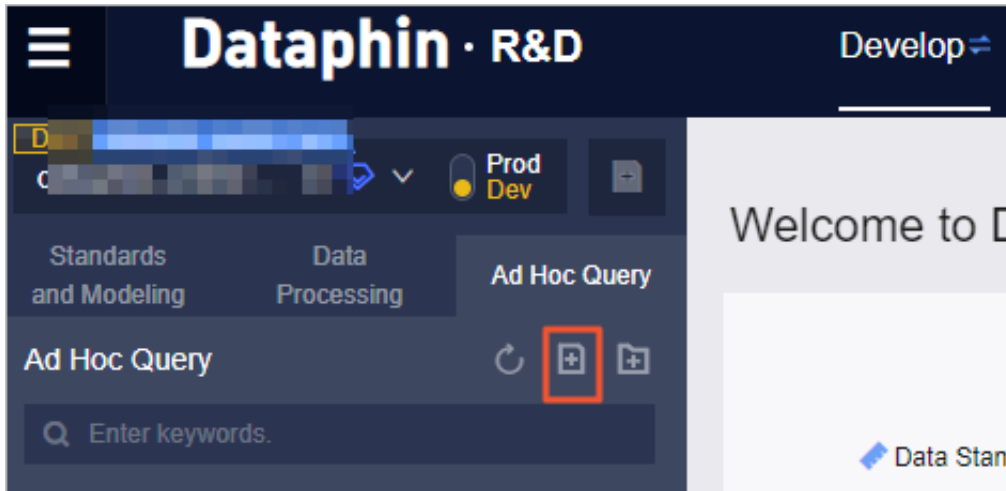
If your computing engine type is Hadoop, the Hive SQL syntax is supported in ad hoc queries. If your computing engine type is MaxCompute, the MaxCompute SQL syntax is supported in ad hoc queries. Dataphin can automatically identify the SQL syntax based on your computing engine configuration.

#### Procedure

1. [Log on to the Dataphin console](#).
2. On the Dataphin homepage, click Theme-based Data Service to go to the Ad Hoc Query tab.



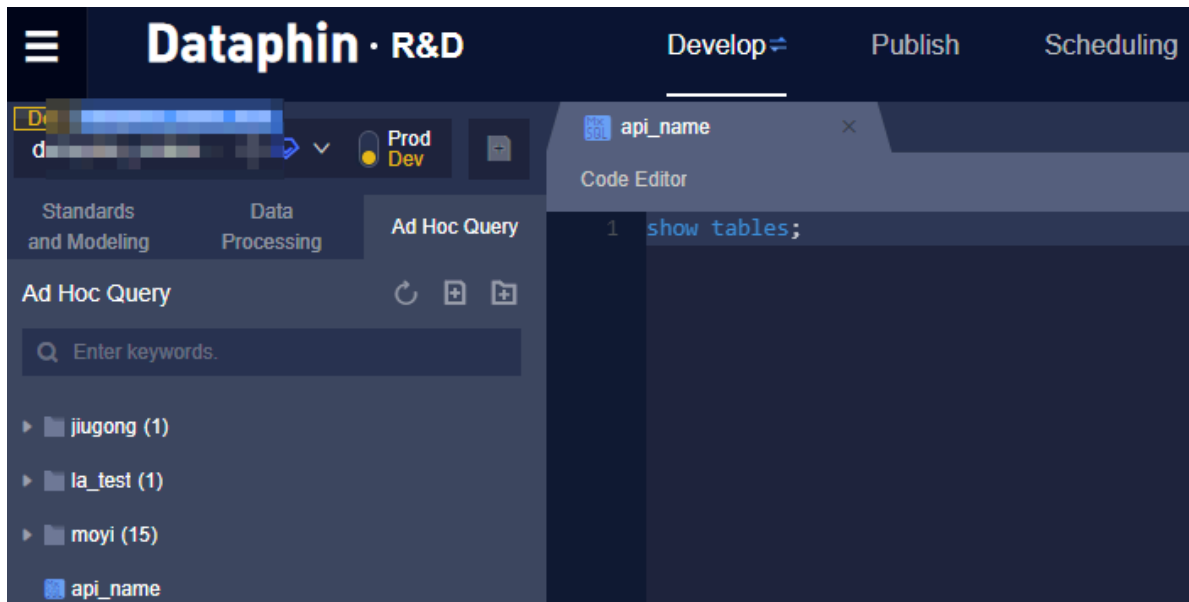
3. On the Ad Hoc Query tab of the Develop tab, click the Create File icon in the left-side navigation pane.



4. In the Create File dialog box that appears, enter the task name and description, select a directory, and then click OK.

The 'Create Item' dialog box is shown with a close button (X) in the top right corner. It contains three input fields: a text field for 'Name' with a red asterisk and the placeholder 'Enter a task name.', a text area for 'Description' with the placeholder 'Enter a task description.', and a dropdown menu for 'Select Directory' currently showing 'Temporary Code'. At the bottom right, there are two buttons: 'Cancel' and 'OK'.

5. In the left-side ad hoc query task list, click the ad hoc query task that you created in preceding steps to go to the Code Editor tab.



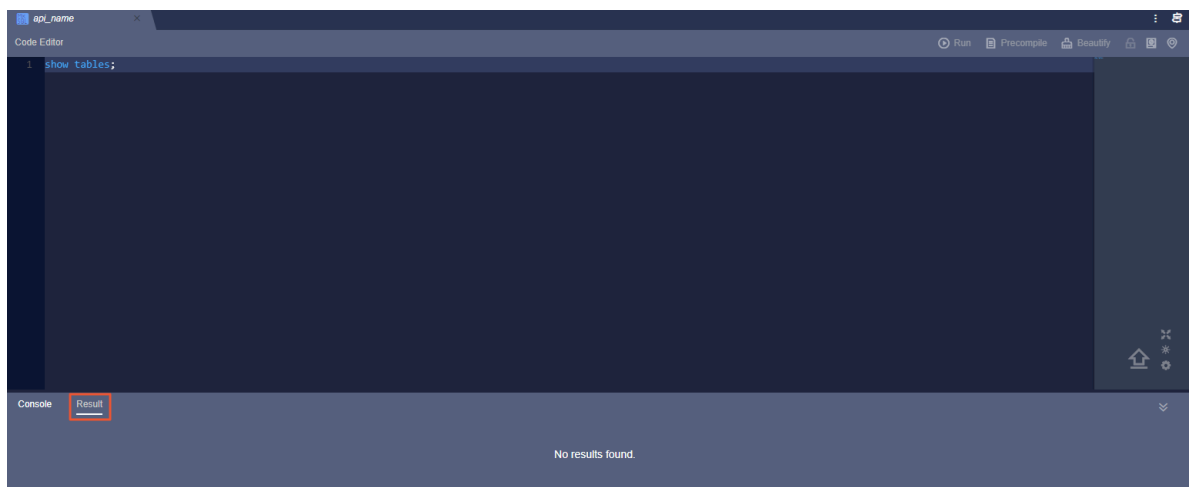
6. Write SQL statements, and click the Save icon and then Run in the upper-right corner.



**Note:**

When you write SQL statements, reference table names in the following format:  
Project name. Table name, **for example**, demo.dim\_qwe.

7. After SQL statements are run, view the results on the Result tab.



## 10 Elasticsearch

---

### 10.1 What is Elasticsearch?

Elasticsearch is a distributed search and data analytics service based on Lucene. It provides a distributed multi-tenant search engine that supports full text queries. This engine is based on a RESTful Web interface. Elasticsearch is developed based on Java. It is released as an open source product that complies with the Apache license terms and conditions. Elasticsearch is a mainstream search engine for enterprises. Elasticsearch is designed to serve cloud computing for real-time search. It is stable, reliable, fast, and easy to install and use.

Apsara Stack Elasticsearch provides two open source versions: Elasticsearch V5.5.3 and Elasticsearch V6.3.2. Apsara Stack Elasticsearch is designed to serve users in data search, data analytics, and other scenarios. Based on open source Elasticsearch, Apsara Stack Elasticsearch also supports enterprise-class permission management.

The default plug-ins provided by Apsara Stack Elasticsearch include but are not limited to the following:

- **IK analyzer:** an open source and lightweight Chinese analysis kit based on Java. The IK analyzer plug-in is very popular in open source communities for Chinese tokenization.
- **Smart Chinese analysis plug-in:** the default Lucene Chinese tokenizer.
- **ICU analysis plug-in:** a Lucene ICU tokenizer. ICU is a set of stable, tested, powerful, and easy to use libraries, providing Unicode and globalization support for applications.
- **Japanese (Kuromoji) analysis plug-in:** a Japanese tokenizer.
- **Stempel (Polish) analysis plug-in:** a French tokenizer.
- **Mapper attachments type plug-in:** an attachment-type plug-in which can parse files of different types into strings based on the Tika library.

## 10.2 Planning and preparation

### 10.2.1 Data types

This topic describes the major types of data that is stored in Apsara Stack Elasticsearch. This can help you estimate the required storage space and manage storage space properly.

Small storage space results in high disk utilization of Elasticsearch cluster logs.

Apsara Stack Elasticsearch mainly stores the following types of data:

- User data that has been pushed to Elasticsearch.
- Elasticsearch replicas. You can specify the number of replicas for each index, but must make sure that each index has a minimum of one replica.



**Notice:**

When an Apsara Stack Elasticsearch instance experiences a spike in disk usage, such as a spike higher than 80%, the health status of the Elasticsearch instance changes to yellow or red. In this situation, the Elasticsearch instance cannot be restarted. Before you restart the Elasticsearch instance, make sure that the health status of the instance is green.

### 10.2.2 Connect to Elasticsearch

This topic describes the methods used to connect to Apsara Stack Elasticsearch:

- Purchase an Elastic Compute Service (ECS) instance deployed in the same VPC network and region as your Elasticsearch instance. Then use the ECS instance to connect to the internal network endpoint of the Elasticsearch instance.
- Connect to the public network endpoint of the Elasticsearch instance.
- Connect to the Elasticsearch instance from the Kibana console of the Elasticsearch instance.

Use an ECS instance deployed in the same VPC network

1. For more information about how to create an ECS instance and an Elasticsearch instance, see [Create an ECS instance](#).
2. Log on to the ECS instance through SSH, and then install the curl tool.



**Note:**



For more information about using other methods to log on to an ECS instance, see *ECS* in the Apsara Stack documentation.

3. Add curl to the environment variable of the ECS instance, and use curl to connect to your Elasticsearch instance from the ECS instance.
4. Run the curl command to connect to the internal network endpoint of the Elasticsearch instance.

```
curl http://<HOST>:<PORT>
```



**Note:**

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance.

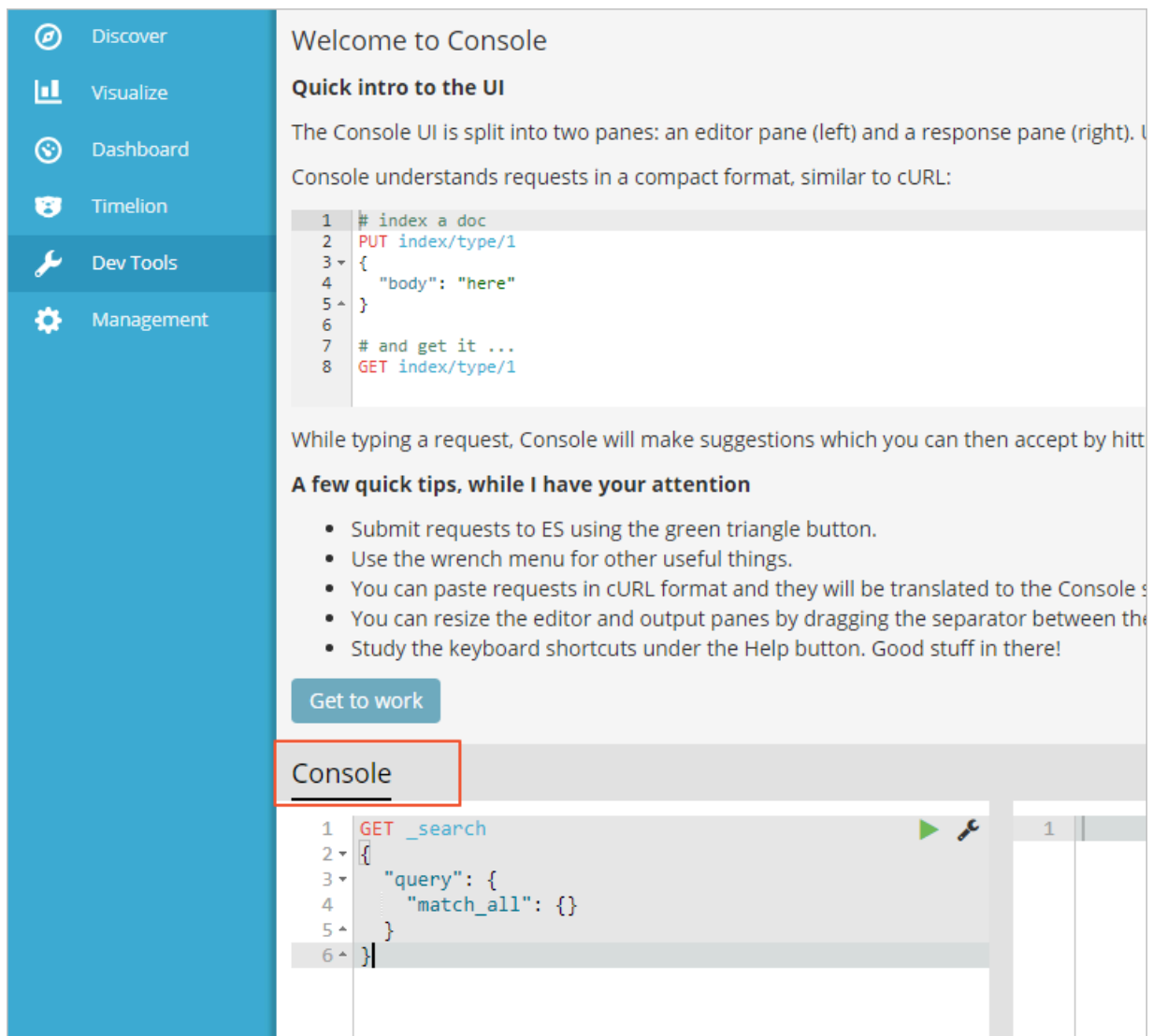
```
[root@iZ7t... ~]# curl http://es-cn-d...elasticsea
rch.aliyun-inc.com:9200
{
 "name" : "k7d3r4E",
 "cluster_name" : "es-cn-d...",
 "cluster_uuid" : "Fmo...",
 "version" : {
 "number" : "5.5.3",
 "build_hash" : "9305a5e",
 "build_date" : "2017-09-07T15:56:59.599Z",
 "build_snapshot" : false,
 "lucene_version" : "6.6.0"
 },
 "tagline" : "You Know, for Search"
}
```

Connect to public network (Apsara Stack internal network) endpoint of the Elasticsearch instance

1. Enable public network access.
2. Configure the public network whitelist.
  - Run the curl command to query the IP address of the client.
  - Add the IP address to the public network whitelist to allow the client to connect to Elasticsearch.

Use the Kibana console to connect to Elasticsearch

Log on to the Kibana console of the Apsara Stack Elasticsearch instance. In the left-side navigation pane, choose Dev Tools. You can then send queries to the Elasticsearch instance from the Console tab.



## Terms

- VPC
  - **Virtual Private Cloud (VPC) is an isolated network environment built on Apsara Stack. VPC networks are logically isolated from each other.**
  - **VPC networks are dedicated to their Apsara Stack tenants. You have full control over your own VPC networks. For example, you can customize the IP range, routing table, and gateway. You can also use Apsara Stack resources, such as ECS, ApsaraDB for RDS, and Server Load Balancer (SLB) instances, in your own VPC network.**
- ECS

**Elastic Compute Service (ECS) is a basic cloud computing service of Apsara Stack. ECS is high performance and easy to use. You can purchase as many ECS instances as needed to meet your workload requirements and no hardware**

devices are required. When you use ECS instances, you can expand disks or increase the bandwidth on demand as your workloads scale out. If you no longer need an ECS instance, you can release it immediately to save costs.

- Elasticsearch

Elasticsearch is a Lucene-based data search and analysis tool that provides distributed services. Elasticsearch is an open source product that complies with the Apache open standards. It is a mainstream enterprise-class search engine.

## 10.3 Quick start

This topic shows you how to create an Elasticsearch instance based on ECS. It covers creating a VPC, creating a security group, creating an ECS instance, creating an Elasticsearch instance, and connecting to an Elasticsearch instance.

Create an Elasticsearch instance based on ECS

After you create an Elasticsearch instance, a VPC, and an ECS instance (in the same region as the Elasticsearch instance), the ECS instance can be used as the client. Then you can deploy a user program or run the curl command.



### Note:

If your Elasticsearch instance and ECS instance share the same VPC and region but reside in different zones, you must create a VSwitch in the zone where the ECS instance resides to ensure that the ECS instance can connect to your Elasticsearch instance.

### 10.3.1 Create a VPC

1. Log on to the Virtual Private Cloud console. Click the VPC tab. On the VPC tab page, click Create.

Virtual Private Cloud (VPC)

VPCs | NAT Gateways | Router Interfaces

Department: All Region: All Regions Name: Search Refresh Create

## 2. Configure the required parameters.

**Create VPC**

1 Create VPC 2 Create VSwitch

\* Name

Description

\* Region

\* Department

\* Shared with Subdepartments ☐ Yes ☒ No ?

\* CIDR Block

① You cannot change the CIDR block of a VPC once the VPC is created.

OK Cancel

## 3. Click OK. The VPC is created, as shown in the following figure.

**Create VPC**

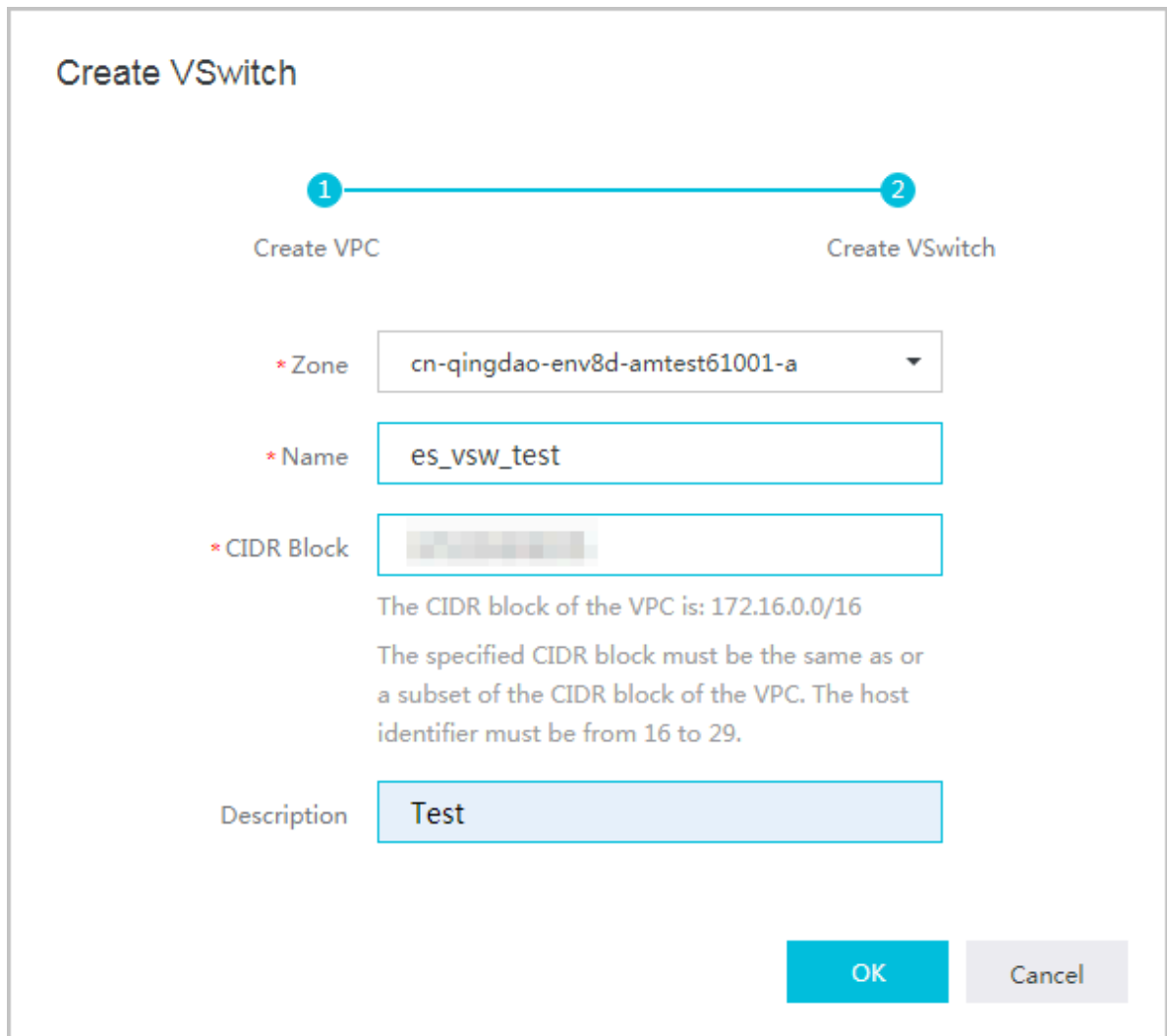
1 Create VPC 2 Create VSwitch

VPC ID

You can continue to manage VSwitches. [Manage VSwitches](#)

Next Cancel

4. Click Next to create a VSwitch.



The 'Create VSwitch' dialog box shows a progress bar with two steps: '1 Create VPC' and '2 Create VSwitch'. The '2 Create VSwitch' step is active. The form contains the following fields:

- \* Zone:** A dropdown menu showing 'cn-qingdao-env8d-amtest61001-a'.
- \* Name:** A text input field containing 'es\_vsw\_test'.
- \* CIDR Block:** A text input field containing a blurred value. Below it, a message states: 'The CIDR block of the VPC is: 172.16.0.0/16. The specified CIDR block must be the same as or a subset of the CIDR block of the VPC. The host identifier must be from 16 to 29.'
- Description:** A text input field containing 'Test'.

At the bottom right, there are two buttons: 'OK' (highlighted in blue) and 'Cancel' (greyed out).

5. Configure the required parameters and click OK. The VSwitch is created, as shown in the following figure.



The 'Create VSwitch' dialog box shows the 'VSwitch ID' field with the value 'vsw-q8c' followed by a blurred suffix. At the bottom right, there are two buttons: 'Continue' (highlighted in blue) and 'Cancel' (greyed out).

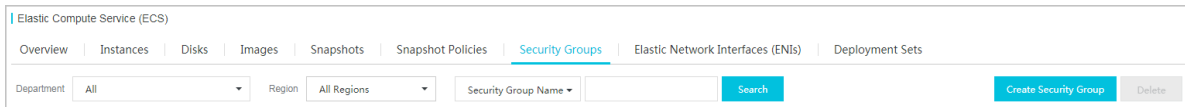


**Note:**

For more information about how to create a VPC, see *VPC User Guide* .

## 10.3.2 Create a security group

1. Log on to the Elastic Compute Service console. Click the Security Groups tab. On the Security Groups tab page, click Create Security Group.



2. Configure the required parameters and click OK.




The screenshot shows the 'Create Security Group' form. It has a title 'Create Security Group' with a refresh icon. The form contains several fields: 'Region' (with a location pin icon and a dropdown menu showing 'cn-qingdao-env8d-d01'), 'Basic Settings' section with 'Department' (dropdown showing 'asr\_test'), 'Project' (dropdown showing 'asr\_test' with a close icon), 'Network Type' (radio button selected for 'VPC'), 'Security Group Name' (text input field containing 'SGTest'), and 'Description' (text area). At the bottom, there are three buttons: 'Submit' (with a list icon), 'OK' (in blue), and 'Cancel' (in grey).

Table 10-1: Parameter description

Parameter	Description
Region	Same as that of the VPC.
Department	Same as that of the VPC.

Parameter	Description
Network Type	VPC.
VPC	Select the new VPC as stated in <a href="#">Create a VPC</a> .

3. The new security group is displayed on the Security Groups tab page. Click the management icon in the Actions column corresponding to the new security group and choose View Details from the shortcut menu.

<input type="checkbox"/>	Security Group ID/Name	Department	Project	Region	ECS Instances	Network Type	VPC ID/Name	Description	Created At	Actions
<input type="checkbox"/>	sg-q8...	asr_test	asr_test	cn-qingdao-env8...	0	VPC	vpc-q8...	--	Jun 3, 2019, 11:01:05 GMT+8	
<input type="checkbox"/>	sg-q8...	rdm_m	rdm_m	cn-qingdao-env8...	1	VPC	vpc-q8...	rdm_m_sg	May 28, 2019, 1...	
<input type="checkbox"/>	sg-q8...	develep	develop_test	cn-qingdao-env8...	1	VPC	vpc-q8...	--	May 23, 2019, 1...	

View Details

Change

Delete

4. On the Security Group Details page, click the Security Group Rules tab. On the Security Group Rules tab page, click Add Security Group Rule. In the dialog box that appears, configure the required parameters and click OK.

Create Security Group Rule

\* Authorization Policy

Allow

\* Rule Direction

Inbound

\* Protocol Type

All

\* Port Range

-1/-1

The value range is 1 to 65,535, with the start and end separated by a slash (/), for example "1/200".

\* Priority

1

\* Authorization Type

IP Address Range Access

\* Authorized IP Addresses

0.0.0.0/0

Description

OK

Cancel

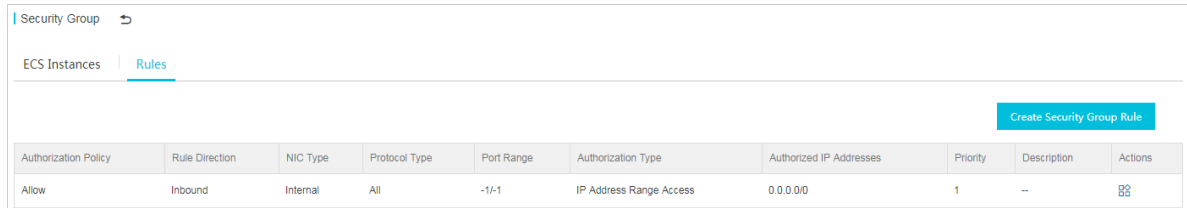
Table 10-2: Parameter description

Parameter	Description
Authorization Type	Select IP Address Range Access.



Parameter	Description
Authorized IP Addresses	Select the same CIDR block as the VPC.

The new security group rule is added, as shown in the following figure.



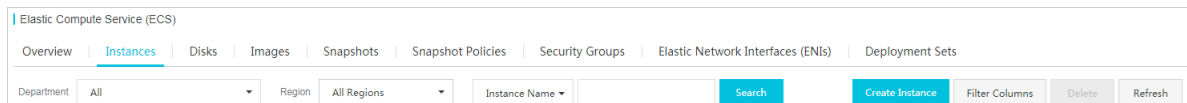
The screenshot shows the 'Security Group' page in the AWS console, specifically the 'Rules' tab. A table lists the security group rules. A new rule has been added with the following details:

Authorization Policy	Rule Direction	NIC Type	Protocol Type	Port Range	Authorization Type	Authorized IP Addresses	Priority	Description	Actions
Allow	Inbound	Internal	All	-1/-1	IP Address Range Access	0.0.0.0/0	1	--	<a href="#">Edit</a>

Buttons for 'ECS Instances' and 'Rules' are visible at the top. A 'Create Security Group Rule' button is also present.

### 10.3.3 Create an ECS instance

1. Log on to the Elastic Compute Service console. Click the Instances tab. On the Instances tab page, click Create Instance.



The screenshot shows the 'Elastic Compute Service (ECS)' console. The 'Instances' tab is selected. The page includes filters for Department (All), Region (All Regions), and Instance Name. Buttons for Search, Create Instance, Filter Columns, Delete, and Refresh are visible.

## 2. On the Create Cloud Server (ECS) page, configure the required parameters and click Create.

Create Instance ↗

📍 Region \* Region

\* Zone

---

⚙️ Basic Settings \* Department  [Create Department >](#)

\* Project  [Create Project >](#)

Select a department first.

---

🌐 Network \* Network Type

Select a department first. Select a department first.

VPC Name:  
VPC ID:  
Department: All  
VPC CIDR Block:

VSwitch Name:  
VSwitch ID:  
VSwitch CIDR Block:

Configure Private IP Address

Table 10-3: Parameter description

Parameter	Description
Network Type	VPC. Select an existing VPC from the first drop-down list box and an existing VSwitch from the second drop-down list box.
Security Group	Select an existing security group.



### Note:

For more information about other parameters, see *ECS User Guide*.

## 3. The instance is being created after you click Create.

Overview

Instances

Disks

Images

Snapshots

Snapshot Policies

Security Groups

Elastic Network Interfaces (ENIs)

Deployment Sets

Department

All

Region

All Regions

Instance Name

Search

Create Instance

Filter Columns

Delete

Refresh

<input type="checkbox"/>	Instance ID	Instance Name	Department	Project	Region	OS	Network Type	IP Address	Monitoring	Configuration Details	Status	Created At	Actions
<input checked="" type="checkbox"/>	i-q8...	rdr_ecs					VPC	(Private)	Monitoring	CPU: 1CoresMemory: 2GB Data Disk: 0GB	Running	May 28, 2019, 14:48:00	

### 10.3.4 Log on to the Elasticsearch console

This topic describes how to log on to the Elasticsearch console.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

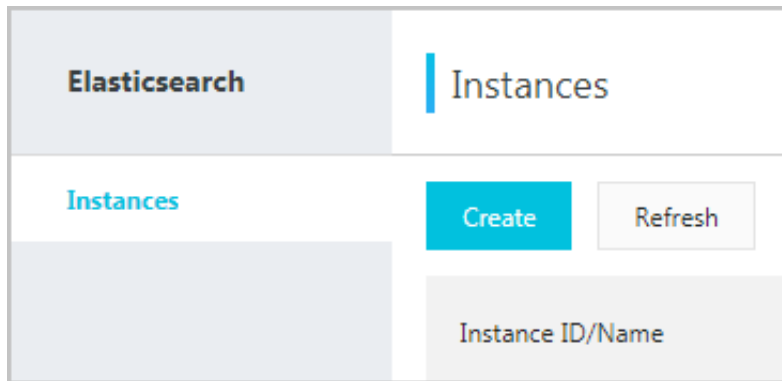
#### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username `super`. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. In the left-side navigation pane, select Big Data > Elasticsearch to open the portal of Elasticsearch.
6. Select a region and department, and click ELASTICSEARCH to log on to the Elasticsearch console.

### 10.3.5 Create an Elasticsearch instance

This topic describes how to create an instance in the Elasticsearch console. The Elasticsearch instance must be created in the same VPC network and zone as the ECS instance from which you connect to the Elasticsearch instance.

1. Log on to the [Elasticsearch console](#).
2. Click Create.



3. Specify the region, instance, storage, and password.

#### Region

Region	* Region	<input type="text" value="cn-qingdao-env8d-d01"/>
	* Zone	<input type="text" value="Select"/>



**Notice:**

**An Apsara Stack Elasticsearch instance can be deployed across multiple regions. You may find deployment options available from the drop-down Zones list. Select an appropriate deployment.**

## Instance

\* Version: 5.5.3, 6.3.2

\* Network Type: VPC

\* VPC: Select

\* VSwitch: Select

\* Specification Family: Local SATA, Local SSD, Cloud Disk

**Instance**

\* Instance Specification: Select

\* Count:   
A minimum of two nodes are required. A cluster that contains only two nodes may have the split-brain syndrome.

Dedicated Master Node: ☐

Client Node: ☐

Warm Node: ☐

- **Version:** Elasticsearch V5.5.3 and V6.3.2 are supported.
- **Network Type:** VPC networks are supported.
- **VPC:** select the same VPC network as your ECS instance.
- **VSwitch:** select the same VSwitch as your ECS instance.
- **Specification Family and Instance Type:** Elasticsearch supports the following specification families and instance types

Specification family	Instance type
Local SATA	16-core 64 GB and 8-core 32 GB
Local SSD	16-core 64 GB and 8-core 32 GB

Specification family	Instance type
Standard disk	16-core 64 GB, 8-core 32 GB, and 4-core 16 GB

- **Nodes:** specify the number of nodes. A minimum of two nodes are required. A cluster that contains only two nodes may have the split-brain syndrome.
- **Dedicated Master Node:** add dedicated master nodes to the Elasticsearch instance. We recommend that you select this option to improve the stability of the instance. The supported dedicated master node types are 4-core 16 GB, 8-core 32 GB, and 16-core 64 GB. Standard SSDs are supported for storage.
- **Client Node:** For CPU-intensive workloads, we recommend that you purchase client nodes to offset the CPU loads from the data nodes so that you can improve the performance and stability of your workloads. For example, you can use client nodes to offset the loads if too many aggregation operations are performed. The supported client node types are 4-core 16 GB, 8-core 32 GB, and 16-core 64 GB. Ultra disks are supported for storage.
- **Warm Node:** If your workloads involve the following index types at the same time, we recommend that you select this option to store warm and hot data separately. This improves the computing performance and service stability.
  - Frequently queried or written indexes
  - Infrequently queried or written indexes, typically indexes of records.

Hot-warm architecture

If you select Warm Node on the Elasticsearch instance buy page or Configuration Upgrade page, `-Enode.attr.box_type` is added to the node startup configuration.

Node type	Startup parameter
Data node	<code>-Enode.attr.box_type=hot</code>

Node type	Startup parameter
Warm node	-Enode.attr.box_type=warm

The supported warm node types are 4-core 16 GB, 8-core 32 GB, and 16-core 64 GB. Ultra disks are supported for storage.

## Storage

Storage

\* Note

When you configure the storage settings for data nodes, you must specify the disk type and storage space. The total storage space assigned to an Elasticsearch instance equals the storage space per data node multiplied by the number of data nodes.

We recommend that you determine the storage space required for storing indexes, shards, replicas, and reserved resources before configuring the storage settings. The storage settings are not applied to dedicated master nodes.

\* Disk Type

Select

\* Storage Space per Data Node

Unit: Gigabytes. An SSD disk supports up to 2,048 GiB of storage space.

You can expand an ultra disk to a maximum of 2,048 GiB. The largest ultra disk that you can purchase is 5,120 GiB. Ultra disks larger than 2,048 GiB include 2,560 GiB, 3,072 GiB, 3,584 GiB, 4,096 GiB, 4,608 GiB, and 5,120 GiB.

Standard SSDs and ultra disks are supported. A standard SSD can provide up to 2 TiB of storage space. Use standard SSDs in online data analytics and searches that require high throughput and fast response. An ultra disk can provide up to 5 TiB of storage space. Ultra disks are cost-effective. We recommend that you use ultra disks in scenarios where volumetric data is logged or analyzed. Ultra disks larger than 2.5 TiB use the disk array and RAID 0 technologies to provide services. You cannot expand these disks.

## Password

Password

\* Username

elastic

This password is used to log on to the Elasticsearch instance and Kibana console.

\* Password

\*\*\*\*\*

\*\*\*\*\*

The password must be 8 to 32 characters in length and can contain English letters, numbers, and special characters. Special characters include ! @ # \$ % ^ & \* ( ) \_ + - =

### 4. Click Create to create an Elasticsearch instance.

You can then find the instance in the instance list. When the instance status displays Active, the instance is successfully created.

**Note:**

You do not have to create an ECS instance and an Elasticsearch instance in sequence. However, you must make sure that both instances are connected to the same VPC network and zone.

### 10.3.6 Connect to an Elasticsearch instance

This topic shows you how to connect to an Elasticsearch instance through an ECS instance.

1. Log on to the ECS instance through SSH, and then install the curl tool.

**Note:**

For more information about using other methods to log on to an ECS instance, see *ECS* in the Apsara Stack documentation.

2. Add curl to the environment variable of the ECS instance, and use curl to connect to your Elasticsearch instance from the ECS instance.
3. Run the curl command to connect to the internal network endpoint of the Elasticsearch instance.

```
curl http://<HOST>:<PORT>
```

**Note:**

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance.

```
[root@iZ7t... ~]# curl http://es-cn-d...elasticsearch.aliyun-inc.com:9200
{
 "name" : "k7d3r4E",
 "cluster_name" : "es-cn-d...",
 "cluster_uuid" : "Fmo...",
 "version" : {
 "number" : "5.5.3",
 "build_hash" : "9305a5e",
 "build_date" : "2017-09-07T15:56:59.599Z",
 "build_snapshot" : false,
 "lucene_version" : "6.6.0"
 },
 "tagline" : "You Know, for Search"
}
```



## 10.4 Instance management

Elasticsearch instance management covers Kibana console, restart instances, and refresh.

### 10.4.1 Kibana console

Apsara Stack Elasticsearch provides the Kibana console for you to scale your businesses. The Kibana console has been seamlessly integrated into Elasticsearch, allowing you to view the status of your Elasticsearch instances and manage these instances.

1. Log on to the Elasticsearch console.
2. Click the `instance ID` to go to the instance details page.
3. Click **Kibana Console** in the Basic Information area to log on to the Kibana console.

### 10.4.2 Restart an instance

The instance restart feature allows you to perform the restart or forced restart operation on your Elasticsearch cluster. Select an appropriate restart method based on your business scenario.

1. Log on to the Elasticsearch console.
2. Click an Elasticsearch instance ID or click **Manage** in the Actions column corresponding to an Elasticsearch instance to go to the Basic Information page.
3. Click **Restart Instance** at the upper-right corner. A dialog box is displayed.
4. Select a restart method and click **OK**.
  - **Restart:** The instance continues providing the highly efficient and highly available service (including at least one replica) during restart. This restart method takes a longer time compared with force restart.
  - **Force Restart:** may cause the Elasticsearch cluster to provide unstable service during the process. However, this restart method is faster than the previous restart method.



**Note:**

Ensure that the health status of your Elasticsearch instance is green. The CPU utilization and memory usage of the Elasticsearch instance surge during the

restart process. This may affect the stability of your service for a short period of time.



**Notice:**

When an Elasticsearch instance has a high disk usage, such as 85% or higher, the health status of the instance may change to yellow or red. In this case, the restart operation is disabled, and you can only perform forced restart.

- We recommend that you do not perform instance operations (such as node scaling, disk scaling, restart, password change, and configuration modification ) when the health status of your Elasticsearch instance is yellow or red. Perform these operations when the health status of your instance is green.
- If changing the configuration of an unhealthy instance that contains two or more nodes causes the instance to remain in the Initializing status, submit a ticket to resolve this issue.
- If performing the update, restart, scaling, or password reset operation on an Elasticsearch instance that contains only one node causes the service to become unavailable during the execution of the operation, create another Elasticsearch instance and migrate your service to the new instance.

### 10.4.3 Refresh

If part of information in the console (such as the status of a newly created Elasticsearch instance) is not refreshed in time, the console may fail to display the information. In this case, you can manually refresh the Elasticsearch instance status on the page.

1. Log on to the Elasticsearch console.
2. Click an Elasticsearch instance ID or click Manage in the Actions column corresponding to an Elasticsearch instance to go to the Basic Information page.
3. Click Refresh at the upper-right corner to refresh the Elasticsearch instance status.

## 10.4.4 Basic information

The screenshot displays the 'Basic Information' page for an Elasticsearch instance. On the left is a sidebar with navigation links: 'Basic Information' (selected), 'Cluster Configuration', 'Plug-ins', 'Security', and 'Snapshots'. The main content area is divided into two sections: 'Basic Information' and 'Configuration'. The 'Basic Information' section includes fields for Instance ID, Name (with an 'Edit' link), Elasticsearch Version (6.3.2), Regions, VPC, Internal Network Address, Kibana Console Connection URL, Created At (May 20, 2019, 20:59:32), Status (Initializing), Billing Method (Pay-As-You-Go), Zone, VSwitch, and Internal Network Port. An 'Upgrade' button is located at the bottom right of this section. The 'Configuration' section shows Data Node Type (elasticsearch.sn2ne.2xlarge(8Cores 32G)), Data Nodes (2), Disk Type (SSD Cloud Disk), and Storage Space (20 GiB).

Table 10-4: Parameters

Parameter	Description
Upgrade	For more information, see <a href="#">Cluster upgrade</a> .
Name	The name of the instance. By default, the name of an Elasticsearch instance is the same as its ID. You can specify a name for an instance. You can also enter an instance name into the search box on the instances page to search for the instance.
Dedicated Master Node	Apsara Stack Elasticsearch dedicated master nodes. Dedicated master nodes are used to improve the stability of the instance. If you have purchased dedicated master nodes, this parameter displays Enabled on the basic information page.
Internal Network Address	You can use an internal network address to access an Elasticsearch instance from an ECS instance that is connected to the VPC network as the Elasticsearch instance. Supported ports include port 9200 for HTTP and port 9300 for TCP.
Kibana console	This parameter shows the address that is used to log on to the Kibana console.

Parameter	Description
Other parameters	For other parameters that are not described in this table, reference their parameter names.

## 10.4.5 Elasticsearch cluster configurations

Elasticsearch cluster configurations include system configurations, language analysis configurations, and YML configurations.

### 10.4.5.1 Word splitting

The synonym settings in the word splitting configuration is mainly applied to the Elasticsearch synonym dictionary. After you configure a synonym filter, new indexes are tokenized according to the latest synonym dictionary.

#### Description

When you configure the synonym settings, you can define a synonym in each row of the UTF-8 encoded `.txt` file.



#### Note:

- After you upload and submit a synonym dictionary file, Elasticsearch does not need to restart the instance to update the dictionary. However, it takes a period of time for the new configuration to take effect.
- If you want to use the synonym dictionary to tokenize indexes that are created before the uploaded synonym dictionary file takes effect, then you must recreate the indexes and configure the synonym settings.

#### Procedure

1. Upload and save a synonym dictionary file in the Apsara Stack Elasticsearch console. Make sure that the uploaded file takes effect.
2. When you create an index and configure the `settings`, you need to specify the `"synonyms_path": "analysis/your_dict_name.txt"` path. Add a mapping for this index to configure synonyms for the specified field.
3. Verify the synonyms and upload a file for testing.

For more information, see [Configure synonyms](#).

## 10.4.5.2 Configure synonyms

### Description



#### Note:

- After you upload a synonym dictionary file to an Apsara Stack Elasticsearch instance, you do not need to restart the nodes in the instance. The system will send the synonym dictionary file to all nodes. Depending on the number of nodes, this process may be time-consuming.
- For example, the index 'index-aliyun' is using the synonym dictionary file 'aliyun.txt'. You have uploaded a new synonym dictionary file to overwrite the existing dictionary file. However, the index 'index-aliyun' cannot automatically load the updated dictionary file. If you want the index to load the updated dictionary file, disable the index and then re-enable the index. We recommend that you rebuild the index after you update the dictionary file as a best practice. Otherwise, this may cause an issue that only the newly created data is using the updated dictionary file.

You can use a filter to configure synonyms. The sample code is as follows:

```
PUT /test_index
{
 "settings": {
 "index": {
 "analysis": {
 "analyzer": {
 "synonym": {
 "tokenizer": "whitespace",
 "filter": ["synonym"]
 },
 "filter": {
 "synonym": {
 "type": "synonym",
 "synonyms_path": "analysis/synonym.txt"
 }
 }
 }
 }
 }
 }
}
```

- **filter:** configure a synonym token filter that contains the `analysis/synonym.txt` path. This path is relative to the location of config.

- **tokenizer:** the tokenizer that tokenizes synonyms. It is set to `whitespace` by default. Additional settings:
  - **ignore\_case:** the default value is `false`.
  - **expand:** the default value is `true`.

**Two synonym formats are supported: Solr and WordNet.**

- **Solr synonyms**

**The following is a sample format of the file:**

```
Blank lines and lines starting with pound are comments.
Explicit mappings match any token sequence on the LHS of "=>"
and replace with all alternatives on the RHS. These types of
mappings
ignore the expand parameter in the schema.
Examples:
i-pod, i pod => ipod,
sea biscuit, sea biscit => seabiscuit
Equivalent synonyms may be separated with commas and give
no explicit mapping. In this case the mapping behavior will
be taken from the expand parameter in the schema. This allows
the same synonym file to be used in different synonym handling
strategies.
Examples:
ipod, i-pod, i pod
foozball , foosball
universe , cosmos
lol, laughing out loud
If expand==true, "ipod, i-pod, i pod" is equivalent
to the explicit mapping:
ipod, i-pod, i pod => ipod, i-pod, i pod
If expand==false, "ipod, i-pod, i pod" is equivalent
to the explicit mapping:
ipod, i-pod, i pod => ipod
Multiple synonym mapping entries are merged.
foo => foo bar
foo => baz
is equivalent to
foo => foo bar, baz
```

**You can also directly define synonyms for the token filter in the configuration file. You must use `synonyms` instead of `synonyms_path`. Example:**

```
PUT /test_index
{
 "settings": {
 "index": {
 "analysis": {
 "filter": {
 "synonym": {
 "type": "synonym",
 "synonyms": [
 "i-pod, i pod => ipod",
 "begin, start"
]
 }
 }
 }
 }
 }
}
```

```

 }
 }
}

```

We recommend that you use `synonyms_path` to define large synonym sets in the file. Using `synonyms` to define large synonym sets will increase the size of your cluster.

- **WordNet synonyms**

Synonyms based on the WordNet format can be declared by using the following format:

```

PUT /test_index
{
 "settings": {
 "index": {
 "analysis": {
 "filter": {
 "synonym": {
 "type": "synonym",
 "format": "wordnet",
 "synonyms": [
 "s(100000001,1,'abstain',v,1,0).",
 "s(100000001,2,'refrain',v,1,0).",
 "s(100000001,3,'desist',v,1,0)."
]
 }
 }
 }
 }
 }
}

```

You can also use `synonyms_path` to define WordNet synonyms in a file.

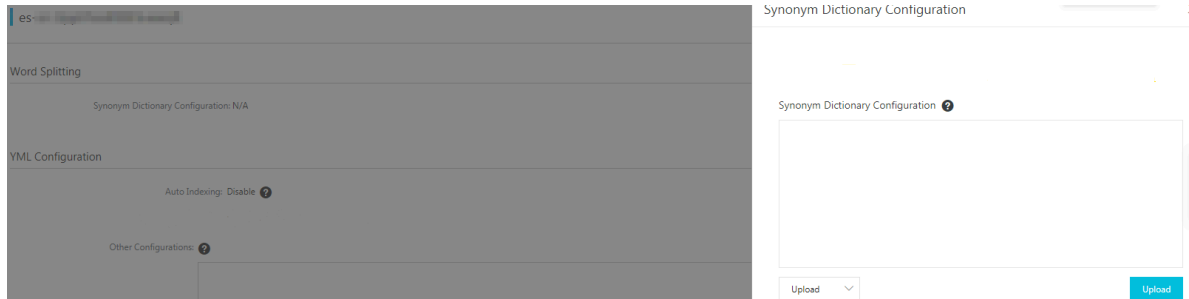
Example 1:

#### Upload a synonym dictionary file

1. Log on to the Apsara Stack Elasticsearch console.
2. Click Create in the upper-left corner to create an Apsara Stack Elasticsearch instance.
3. Click the instance to go to the configuration page.

4. In the left-side navigation pane, select **Cluster Configuration**, and then click **Synonym Dictionary Configuration**.

Figure 10-1: Configure synonyms



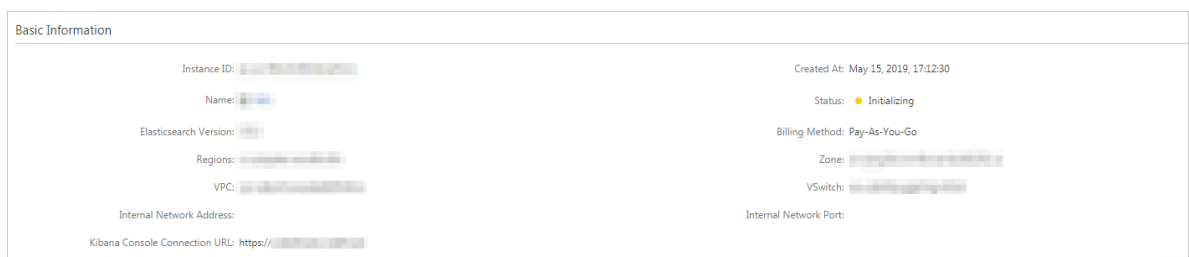
5. Click **Upload**, select the synonym dictionary file that you want to upload, and click **Save**. In this example, the TXT file that is generated as described in the preceding sections is uploaded.

After the Elasticsearch instance is activated and its status changes to **Active**, you can then use the synonym dictionary. In this example, file `aliyun_synonyms.txt` is uploaded for testing. The file contains: `begin, start`

## Configure and test the synonym dictionary

1. Click **Kibana Console** in the **Basic Information** area to log on to the Kibana console.

Figure 10-2: Kibana console



2. In the left-side navigation pane, click **Dev Tool**.
3. Run the following command in the Console to create indexes:

```
PUT aliyun-index-test
{
 "index": {
 "analysis": {
 "analyzer": {
 "by_smart": {
 "type": "custom",
 "tokenizer": "ik_smart",
 "filter": ["by_tfr", "by_sfr"],
```



```

 "char_filter": ["by_cfr"]
 },
 "by_max_word": {
 "type": "custom",
 "tokenizer": "ik_max_word",
 "filter": ["by_tfr", "by_sfr"],
 "char_filter": ["by_cfr"]
 }
 },
 "filter": {
 "by_tfr": {
 "type": "stop",
 "stopwords": [" "]
 },
 "by_sfr": {
 "type": "synonym",
 "synonyms_path": "analysis/aliyun_synonyms.txt"
 }
 },
 "char_filter": {
 "by_cfr": {
 "type": "mapping",
 "mappings": ["| => |"]
 }
 }
 }
}
}

```

**4. Run the following command to configure the title field:**

```

PUT aliyun-index-test/_mapping/doc
{
 "properties": {
 "title": {
 "type": "text",
 "index": "analyzed",
 "analyzer": "by_max_word",
 "search_analyzer": "by_smart"
 }
 }
}

```

**5. Run the following command to verify the synonyms:**

```

GET aliyun-index-test/_analyze
{
 "analyzer": "by_smart",
 "text": "begin"
}

```

**The following results are returned if the configuration takes effect:**

```

{
 "tokens": [
 {
 "token": "begin",
 "start_offset": 0,
 "end_offset": 5,
 "type": "ENGLISH",
 "position": 0
 }
],
}

```

```
{
 "token": "start",
 "start_offset": 0,
 "end_offset": 5,
 "type": "SYNONYM",
 "position": 0
}
]
```

**6. Run the following command to add data for further testing:**

```
PUT aliyun-index-test/doc/1
{
 "title": "Shall I begin?"
}
```

```
PUT aliyun-index-test/doc/2
{
 "title": "I start work at nine."
}
```

**7. Run the following command to perform a query test:**

```
GET aliyun-index-test/_search
{
 "query" : { "match" : { "title" : "begin" }},
 "highlight" : {
 "pre_tags" : ["<red>", "<bule>"],
 "post_tags" : ["</red>", "</bule>"],
 "fields" : {
 "title" : {}
 }
 }
}
```

**If the synonyms are verified, the following results are returned:**

```
{
 "took": 11,
 "timed_out": false,
 "_shards": {
 "total": 5,
 "successful": 5,
 "failed": 0,
 },
 "hits": {
 "total": 2,
 "max_score": 0.41048482,
 "hits": [
 {
 "_index": "aliyun-index-test",
 "_type": "doc",
 "_id": "2",
 "_score": 0.41048482,
 "_source": {
 "title": "I start work at nine."
 },
 "highlight": {
 "title": [
 "I <red>start</red> work at nine."
]
 }
 }
]
 }
}
```

```

]
 }
},
{
 "_index": "aliyun-index-test",
 "_type": "doc",
 "_id": "1",
 "_score": 0.39556286,
 "_source": {
 "title": "Shall I begin?"
 },
 "highlight": {
 "title": [
 "Shall I <red>begin</red>?"
]
 }
}
]
}
}

```

## Example 2

**Follow these steps to directly import the synonyms and use the IK analyzer to filter the synonyms:**

1. **Configure the synonym filter `my_synonym_filter` and a synonym dictionary.**
2. **Configure the `my_synonyms` analyzer, and use the IK analyzer `ik_smart` to split words.**

**The IK analyzer `ik_smart` splits the words and then changes all letters to lowercase.**

```

PUT /my_index
{
 "settings": {
 "analysis": {
 "analyzer": {
 "my_synonyms": {
 "filter": [
 "lowercase",
 "my_synonym_filter"
],
 "tokenizer": "ik_smart"
 }
 },
 "filter": {
 "my_synonym_filter": {
 "synonyms": [
 "begin,start"
],
 "type": "synonym"
 }
 }
 }
 }
}

```

```
}
```

**3. Run the following command to configure the title field:**

```
PUT /my_index/_mapping/doc
{
 "properties": {
 "title": {
 "type": "text",
 "index": "analyzed",
 "analyzer": "my_synonyms"
 }
 }
}
```

**4. Run the following command to verify the synonyms:**

```
GET /my_index/_analyze
{
 "analyzer": "my_synonyms",
 "text": "Shall I begin?"
}
```

**If the synonyms are verified, the following results are returned:**

```
{
 "tokens": [
 {
 "token": "shall",
 "start_offset": 0,
 "end_offset": 5,
 "type": "ENGLISH",
 "position": 0
 },
 {
 "token": "i",
 "start_offset": 6,
 "end_offset": 7,
 "type": "ENGLISH",
 "position": 1
 },
 {
 "token": "begin",
 "start_offset": 8,
 "end_offset": 13,
 "type": "ENGLISH",
 "position": 2
 },
 {
 "token": "start",
 "start_offset": 8,
 "end_offset": 13,
 "type": "SYNONYM",
 "position": 2
 }
]
}
```

**5. Run the following command to add data for further testing:**

```
PUT /my_index/doc/1
```

```
{
 "title": "Shall I begin?"
}
```

```
PUT /my_index/doc/2
{
 "title": "I start work at nine."
}
```

**6. Run the following command to perform a query test:**

```
GET /my_index/_search
{
 "query" : { "match" : { "title" : "begin" }},
 "highlight" : {
 "pre_tags" : ["<red>", "<bule>"],
 "post_tags" : ["</red>", "</bule>"],
 "fields" : {
 "title" : {}
 }
 }
}
```

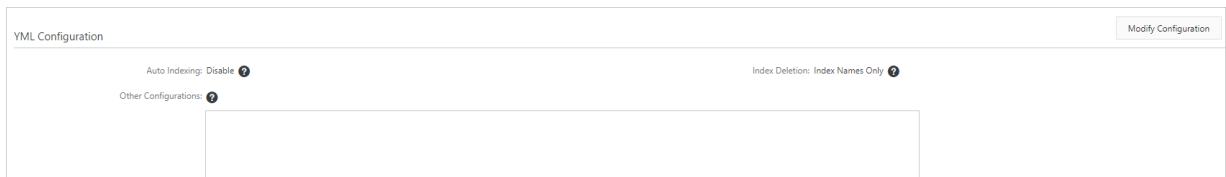
**7. If the synonyms are verified, the following results are returned:**

```
{
 "took": 11,
 "timed_out": false,
 "_shards": {
 "total": 5,
 "successful": 5,
 "failed": 0,
 },
 "hits": {
 "total": 2,
 "max_score": 0.41913947,
 "hits": [
 {
 "_index": "my_index",
 "_type": "doc",
 "_id": "2",
 "_score": 0.41913947,
 "_source": {
 "title": "I start work at nine."
 },
 "highlight": {
 "title": [
 "I <red>start</red> work at nine."
]
 }
 },
 {
 "_index": "my_index",
 "_type": "doc",
 "_id": "1",
 "_score": 0.39556286,
 "_source": {
 "title": "Shall I begin?"
 },
 "highlight": {
 "title": [
 "Shall I <red>begin</red>?"
]
 }
 }
]
 }
}
```

```
]
}
}
]
```

### 10.4.5.3 YML configuration

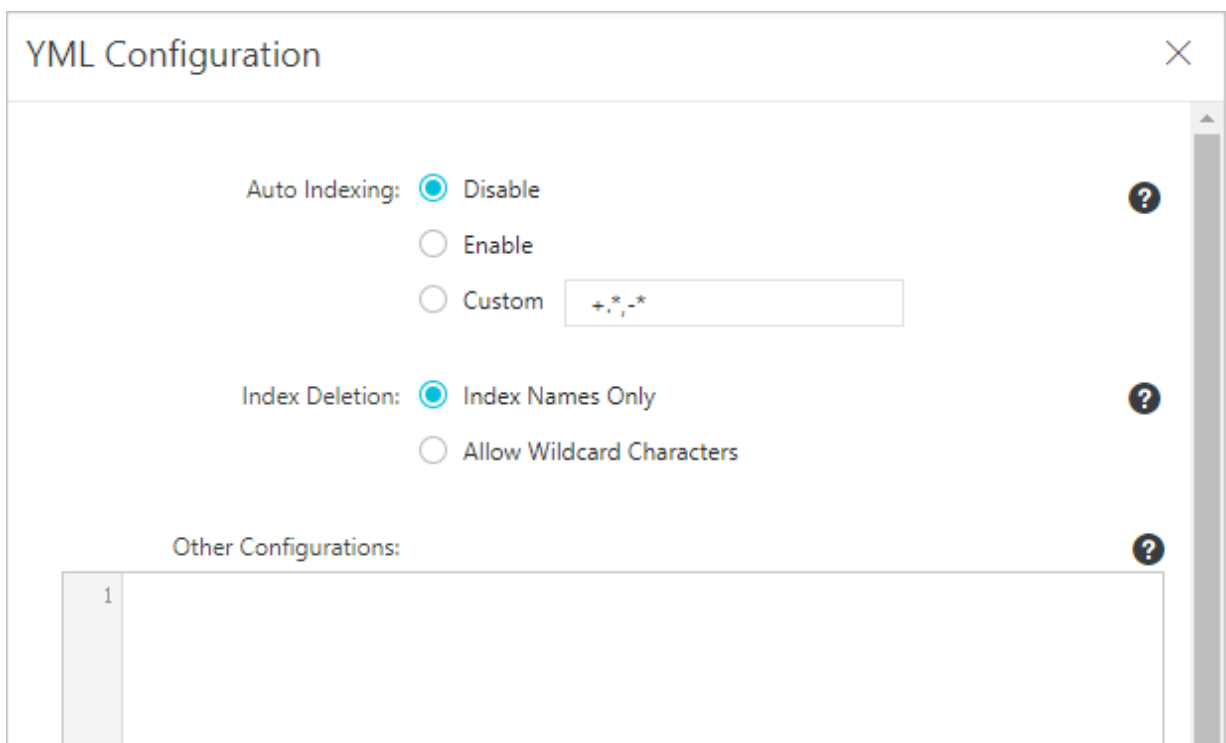
The YML configuration section in the Elasticsearch console displays the configuration of the current Elasticsearch instance.



The screenshot shows the 'YML Configuration' section of the Elasticsearch console. It includes a 'Modify Configuration' button in the top right corner. Below the title, there are two sections: 'Auto Indexing: Disable' with a help icon, and 'Index Deletion: Index Names Only' with a help icon. Under 'Auto Indexing', there is a sub-section 'Other Configurations:' with a help icon and a text input field.

Modify configuration

Click **Modify Configuration** to modify the YML configuration.



The screenshot shows the 'YML Configuration' dialog box. It has a title bar with a close button. The main content area contains three sections: 'Auto Indexing:' with radio buttons for 'Disable' (selected), 'Enable', and 'Custom' (with a text input field containing '+,\*,-\*'); 'Index Deletion:' with radio buttons for 'Index Names Only' (selected) and 'Allow Wildcard Characters'; and 'Other Configurations:' with a table. The table has one row with a column index '1' and a text input field. Each section has a help icon (question mark) to its right.

- **Auto Indexing:** The auto-indexing feature allows the Elasticsearch instance to automatically create new indexes for documents uploaded to an instance if these documents do not have indexes. We recommend that you disable auto-indexing. Indexes created by this feature may not meet your requirements.
- **Index Deletion:** This feature specifies whether you need to specify the name of an index before you delete the index. If you select Allow Wildcard Characters,

you can use wildcard characters to specify multiple indexes. After an index is deleted, it cannot be recovered. Proceed with caution.

- **Other Configurations:**

Some of the supported configuration items are as follows. For more information, see [Configuration parameters](#).

- `http.cors.enabled`
- `http.cors.allow-origin`
- `http.cors.max-age`
- `http.cors.allow-methods`
- `http.cors.allow-headers`
- `http.cors.allow-credentials`
- `reindex.remote.whitelist`
- `action.auto_create_index`
- `action.destructive_requires_name`

Reindex from a remote Elasticsearch instance. You can recreate indexes from a remote Elasticsearch instance that uses any Elasticsearch version. This feature allows you to migrate indexes from an earlier Elasticsearch version to the newly released Elasticsearch version. For more information about how to reindex from a remote Elasticsearch instance, see [Custom remote reindexing \(whitelisting\)](#).

#### 10.4.5.3.1 Configuration parameters

The following table lists HTTP-based custom configuration parameters available in Elasticsearch.



**Note:**

These configuration parameters support only the static configuration mode, but not the hot deployment mode. To activate any of these configuration parameters, write it to the `elasticsearch.yml` configuration file.

Table 10-5: Configuration parameter description

Parameter	Description
<code>http.cors.enabled</code>	<p>The cross-origin resource sharing (CORS) configuration parameter. It is used to enable or disable CORS.</p> <ul style="list-style-type: none"><li>• Elasticsearch can receive requests from browsers of resources in other domains. When this configuration parameter is set to <code>true</code>, Elasticsearch can process the <code>OPTIONS CORS</code> request.</li><li>• If the domain information in the requests has been declared in <code>http.cors.allow-origin</code>, Elasticsearch adds <code>Access-Control-Allow-Origin</code> in the header to respond to the CORS request.</li><li>• When this configuration parameter is set to <code>false</code> (the default value is <code>false</code>), Elasticsearch neglects the domain information in the header and does not add the <code>Access-Control-Allow-Origin</code> to the header to disable CORS.</li><li>• If the client cannot send preflight requests that use the domain information header or does not check <code>Access-Control-Allow-Origin</code> in the header of the response from the server, secure CORS is affected.</li><li>• If CORS is disabled for Elasticsearch, the client can try to send an <code>OPTIONS</code> request to check whether this response exists.</li></ul>



Parameter	Description
<b>http.cors.allow-origin</b>	<p>The CORS resource configuration parameter. It can be used to configure to receive requests from which domains. No domain is allowed and the parameter is left blank by default.</p> <ul style="list-style-type: none"> <li>• If <code>/</code> is added before and after the parameter value, the configuration is identified as a regular expression.</li> <li>• You can use regular expressions to match HTTP- and HTTPS-based domain requests. For example, <code>/https?:\\\/localhost(:[0-9]+)?/</code> allows Elasticsearch to respond to the request satisfying this regular expression.</li> <li>• <code>*</code> is deemed as a valid configuration and indicates that the cluster supports CORS requests from any domain. This poses security risks to the Elasticsearch cluster.</li> </ul>
<b>http.cors.max-age</b>	<p>The browser can send an OPTIONS request to obtain the CORS configuration. <code>max-age</code> specifies how long the browser can retain the output result. The default value is 1,728,000 seconds (20 days).</p>
<b>http.cors.allow-methods</b>	<p>The request method configuration parameter. Valid values are <code>OPTIONS</code>, <code>HEAD</code>, <code>GET</code>, <code>POST</code>, <code>PUT</code>, and <code>DELETE</code>.</p>
<b>http.cors.allow-headers</b>	<p>The request header configuration parameter. Valid values are <code>X-Requested-With</code>, <code>Content-Type</code>, and <code>Content-Length</code>.</p>
<b>http.cors.allow-credentials</b>	<p>The credential configuration parameter. It is used to configure whether to return <code>Access-Control-Allow-Credentials</code> in the response header. If it is set to true, <code>Access-Control-Allow-Credentials</code> is returned. The default value is false.</p>

Parameter	Description
<b>reindex.remote.whitelist</b>	The remote host address whitelist that can access the cluster. Host-port pairs are allowed. Separate multiple pairs with commas (,) (such as otherhost:9200, another:9200, 127.0.10. *:9200, localhost:*). The whitelist only uses the host and port information for security policy configuration.
<b>action.auto_create_index</b>	The auto create index configuration parameter. When it is set to false, the auto create index feature is disabled.
<b>action.destructive_requires_name</b>	Indicates whether you need to specify the name of an index when you delete the index. When it is set to false (the default value), you can use regular expressions or <code>_all</code> to delete indexes. When it is set to true, you must specify index names to delete indexes, but cannot use <code>_all</code> or wildcards.

#### 10.4.5.3.2 Custom remote reindexing (whitelisting)

The reindexing component allows you to reindex data from the remote Elasticsearch cluster. This feature is applicable to remote Elasticsearch instances of any version. It allows you to use the latest version to reindex data of old versions.

```
POST _REINDEX
{
 "SOURCE": {
 "REMOTE": {
 "HOST": "HTTP://OTHERHOST:9200",
 "USERNAME": "USER",
 "PASSWORD": "PASS"
 },
 "INDEX": "SOURCE",
 "QUERY": {
 "MATCH": {
 "TEST": "DATA"
 }
 }
 },
 "DEST": {
 "INDEX": "DEST"
 }
}
```

- host **must** contain the protocol, domain name, and port (such as `https://otherhost:9200`).

- username and password are optional. If the remote Elasticsearch cluster needs to use the basic authorization scheme, the username-password pair is required. If you use the basic authorization scheme, we recommend that you use the HTTPS protocol. Otherwise, the password is transmitted as a text.
- The API can be called remotely only after the remote host address is declared in the `elasticsearch.yaml` configuration file by using the `reindex.remote.whitelist` attribute. `reindex.remote.whitelist` can use host-port pairs. Separate multiple pairs with commas (,) (such as, `otherhost:9200`, `another:9200`, `127.0.10.*:9200`, `localhost:*`). The whitelist does not identify the protocol and only uses the host and port information to configure security policies.
- If the host address is already listed in the whitelist, the query request is not verified or modified, but is directly sent to the remote Elasticsearch cluster.

**Note:**

Remote reindexing does not support manual or automatic slicing.

The remote Elasticsearch cluster uses a stack to cache indexed data. The default maximum size is 100 MB. If a large document is involved in remote reindexing, set the size of the batch settings to a small value.

In the following example, the size of the batch settings is 10, which is the minimum value.

```
POST _reindex
{
 "source": {
 "remote": {
 "host": "http://otherhost:9200"
 },
 "index": "source",
 "size": 10,
 "query": {
 "match": {
 "test": "data"
 }
 }
 },
 "dest": {
 "index": "dest"
 }
}
```

- `socket_timeout`: the timeout period for socket reading. The default value is 30 seconds.
- `connect_timeout`: the connection timeout period. The default value is 1 second.

In the following example, `socket_timeout` is 1 minute and `connect_timeout` is 10 seconds.

```
POST _reindex
{
 "source": {
 "remote": {
 "host": "http://otherhost:9200",
 "socket_timeout": "1m",
 "connect_timeout": "10s"
 },
 "index": "source",
 "query": {
 "match": {
 "test": "data"
 }
 }
 },
 "dest": {
 "index": "dest"
 }
}
```

## 10.4.6 Cluster upgrade

Apsara Stack Elasticsearch instances support upgrading the instance specification, storage space per data node, and number of nodes. Currently, you cannot downgrade Elasticsearch instances.

### Procedure

1. Log on to the Elasticsearch console.
2. Click the ID of the target Elasticsearch instance or click Manage to navigate to the instance details page.
3. Click Upgrade on the right side of the page to open the cluster upgrade dialog box.
4. Edit the attributes of the Elasticsearch instance to meet your business demands, and then click OK.



#### Note:

- You can only edit one attribute at a time, such as the number of nodes, disk space per data node, or instance specification.
- If your business requires a cluster upgrade, we recommend that you make an upgrade assessment before upgrading the cluster.
- After you submit the upgrade order, the Elasticsearch instance will be billed based on the upgraded configuration.

## 10.4.7 Plug-ins

Plug-ins extend the capabilities of Elasticsearch in data pre-processing and data analysis.

Built-in plug-ins

The supported built-in plug-ins are as follows:

Plug-in	Type	Description
analysis-icu	Built-in plug-in	ICU analysis plug-in for Elasticsearch. It integrates the Lucene ICU module into Elasticsearch and adds ICU analysis components.
analysis-ik	Built-in plug-in	IK analysis plug-in for Elasticsearch.
analysis-kuromoji	Built-in plug-in	Japanese (Kuromoji) analysis plug-in for Elasticsearch. It integrates the Lucene Kuromoji analysis module into Elasticsearch.
analysis-phonetic	Built-in plug-in	Phonetic analysis plug-in for Elasticsearch. It integrates the phonetic token filter into Elasticsearch.
analysis-pinyin	Built-in plug-in	Pinyin analysis plug-in for Elasticsearch.
analysis-smartcn	Built-in plug-in	Smart Chinese analysis plug-in for Elasticsearch. It integrates the Lucene smart Chinese analysis module into Elasticsearch.
elasticsearch-repository-oss	Built-in plug-in	The plug-in allows you to use Alibaba Cloud Object Storage Service (OSS) to store Elasticsearch snapshots.

Plug-in	Type	Description
<b>ingest-attachment</b>	<b>Built-in plug-in</b>	Ingest processor for Elasticsearch. It uses Apache Tika to extract content.
<b>ingest-geoip</b>	<b>Built-in plug-in</b>	Ingest processor for Elasticsearch. It queries geo data in MaxMind geo databases based on IP addresses.
<b>ingest-user-agent</b>	<b>Built-in plug-in</b>	Ingest processor for Elasticsearch. It extracts information from a user agent.
<b>mapper-attachments</b>	<b>Built-in plug-in</b>	The mapper attachments plug-in enables Elasticsearch index file attachments based on the Apache text extraction library Tika.
<b>mapper-murmur3</b>	<b>Built-in plug-in</b>	The Mapper Murmur3 plug-in allows you to compute the hashes of a field's values at index time and store them in the index.
<b>mapper-size</b>	<b>Built-in plug-in</b>	The Mapper Size plug-in allows documents to record their uncompressed size at index time.
<b>repository-hdfs</b>	<b>Built-in plug-in</b>	The HDFS repository plug-in enables support for Hadoop Distributed File System (HDFS) repositories.
<b>search-guard-5</b>	<b>Built-in plug-in</b>	This plug-in provides access control related features for Elasticsearch 5.

Plug-in	Type	Description
sql	Built-in plug-in	The plug-in allows you to query Elasticsearch data by using SQL statements.

The analysis-ik plug-in is the Elasticsearch IK analyzer plug-in, which allows you to use the standard update or rolling update method to update IK dictionaries.

#### Standard update

The standard update method requires Elasticsearch to restart all nodes in an Elasticsearch cluster to update the dictionary. Elasticsearch will send the uploaded dictionary file to all nodes in the cluster, modify the `IKAnalyzer.cfg.xml` file, and restart the nodes to load the uploaded dictionary file.

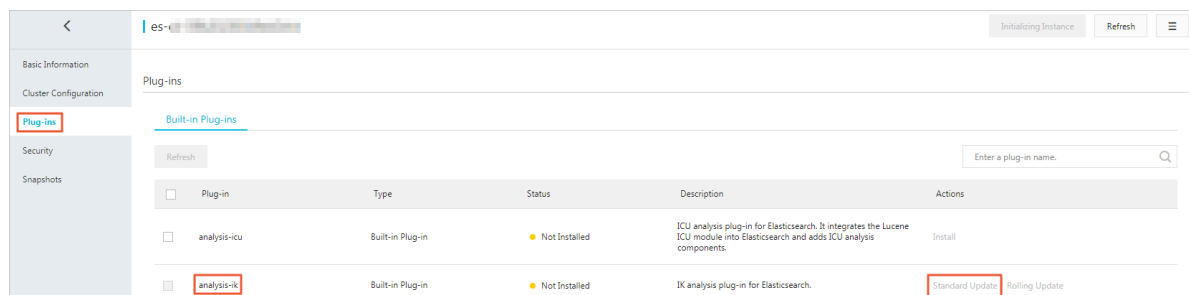
You can use the standard update method to update the IK main dictionary and stopword list. The standard update page shows the built-in main dictionary `SYSTEM_MAIN.dic` and stopword list `SYSTEM_STOPWORD.dic`.

- If you want to update the built-in main dictionary, upload a dictionary file named as `SYSTEM_MAIN.dic`.
- If you want to update the built-in stopword list, upload a dictionary file named as `SYSTEM_STOPWORD.dic`.

#### Standard update example

1. In the Elasticsearch console, click the ID of the Elasticsearch instance that you want to update dictionaries for.
2. Click Plug-ins, locate the analysis-ik plug-in, and click Standard Update.

Figure 10-3: Standard update



### 3. Click Configure.

Figure 10-4: Plug-in configuration

Plug-ins

IK Main Dictionary ?

SYSTEM\_MAIN.dic

IK Stopword List ?

SYSTEM\_STOPWORD.dic

Configure Cancel



#### 4. Click Upload DIC File, and select a main dictionary file.

Figure 10-5: Upload a main dictionary file

IK Main Dictionary ?

SYSTEM\_MAIN.dic X

Upload DIC File ^

Upload DIC File ✓

Add OSS File

SYSTEM\_STOPWORD.dic X

Upload DIC File v

Upload DIC File

! ☐ This operation will restart the instance. Continue?



#### Note:

- By default, you are required to upload a .dic file. You can also choose to upload an OSS file.
- If the content of the dictionary file stored on OSS is changed, you have to manually re-upload the OSS file to Elasticsearch to update the dictionary.

5. Scroll down to the bottom, select **This operation will restart the instance.**

**Continue?**, and click **Save**. The Elasticsearch cluster is then restarted.

6. After the cluster is restarted, log on to the Kibana console and run the following command to verify that the updated dictionary is effective:

```
GET _analyze
{
 "analyzer": "ik_smart",
 "text": ["tokens in your updated dictionary"]
}
```



**Note:**

- You cannot delete the built-in main dictionary and stopword list.
- Whether you upload a new dictionary file, remove a dictionary file, or update the dictionary content, the standard update operation always requires Elasticsearch to restart the cluster.
- You can perform the standard update operation only when the status of the cluster is healthy.

## Rolling update

When the content of your dictionary file changes, you can use the rolling update method to update the dictionary. After you upload the latest dictionary file, the Elasticsearch nodes will automatically load the file.

When you perform a rolling update, if the dictionary file list changes, all nodes in the cluster need to reload the dictionary configuration. For example, when you upload a new dictionary file or delete an existing dictionary file, the changes will be updated to the `IKAnalyzer.cfg.xml` file.

The procedure of rolling update is similar to the standard update. If this is the first time that you have uploaded a dictionary file, you must edit the `IKAnalyzer.cfg.xml` file. This means that Elasticsearch needs to restart the cluster to reload the configuration file. Subsequently, if you upload a dictionary file with the same name, Elasticsearch does not need to restart the cluster for the updates to take effect.

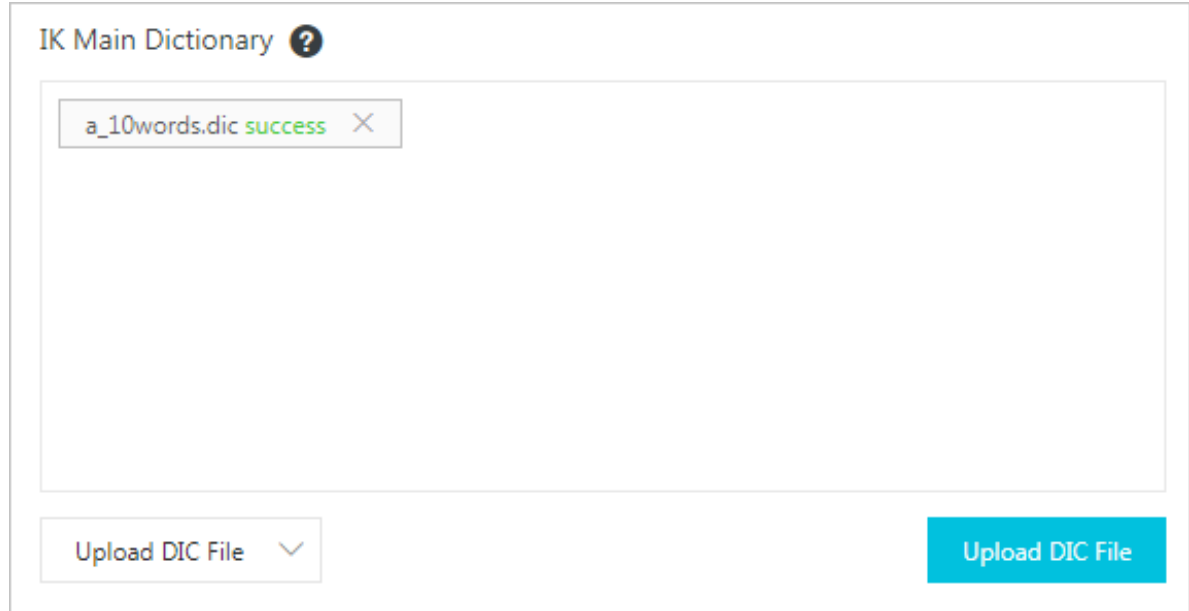
## Rolling update example

1. In the Elasticsearch console, click the ID of the Elasticsearch instance that you want to update the dictionaries for.
2. Click **Plug-ins**, locate the **analysis-ik** plug-in, and click **Rolling Update**.

3. Click **Configure**.

4. Click **Upload DIC File**, and select a main dictionary file.

Figure 10-6: Upload a dic file



**Note:**

- By default, you are required to upload a .dic file. You can also choose to upload an OSS file.
- If the content of the dictionary file stored on OSS is changed, you have to manually re-upload the OSS file to Elasticsearch to update the dictionary.

5. Scroll down to the bottom, select **This operation will restart the instance.**

**Continue?**, and click **Save**. The cluster is then restarted. After the cluster is restarted, the uploaded dictionary automatically takes effect.

If you need to add tokens to or remove tokens from the dictionary, upload a dictionary file with the same name to replace the `a_10words.dic` file. In the rolling update dialog box, delete the dictionary file `a_10words.dic`, and then upload a new dictionary file with the same name. Elasticsearch does not need to restart the cluster because you are editing an existing dictionary file on the cluster. Scroll down to the bottom and then click **Save**.

The plug-in on the nodes of the Elasticsearch cluster will automatically load the dictionary file. The time that each node takes to load the dictionary file varies. Please wait for the new dictionary to take effect. It may take about two minutes for

all nodes to load the dictionary file. You can log on to the Kibana console and run the following command to verify that the new dictionary is effective.

```
GET _analyze
{
 "analyzer": "ik_smart",
 "text": ["tokens in your updated dictionary"]
}
```

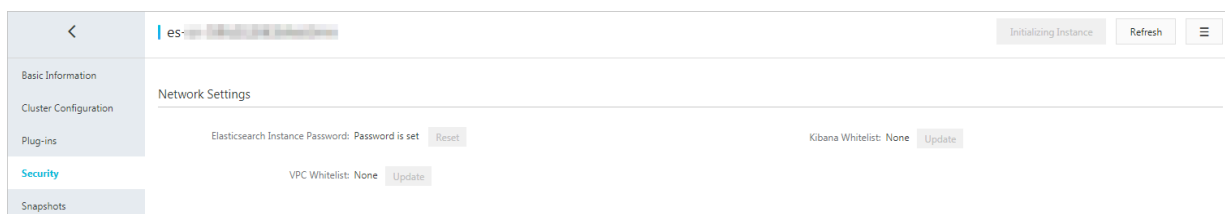
**Note:**

You cannot use the rolling update method to edit the built-in main dictionary. If you want to modify the built-in main dictionary, use the standard update method.

## 10.4.8 Security

On the security page, you can reset the password of the Elasticsearch instance, edit the Kibana whitelist, and edit the VPC whitelist.

Figure 10-7: Security



### Reset the Elasticsearch instance password

The reset Elasticsearch instance password feature only resets the password of the administrator account elastic. After you reset the password, you must use the new password to log on to the Elasticsearch instance and Kibana console.

**Note:**

- The password reset operation does not reset the password of other accounts that are used to log on to the Elasticsearch instance. We recommend that you do not use the elastic account to access your Elasticsearch instance.
- After you confirm the password reset operation, it takes up to five minutes for the new password to take effect.
- After you reset the password, Elasticsearch does not need to restart the instance to apply the new password.

## Kibana whitelist

You can add IP addresses and CIDR blocks to the Kibana whitelist in the format of `192.168.0.1` and `192.168.0.0/24`, respectively. Separate them with commas (,).

You can enter `127.0.0.1` to forbid all IPv4 addresses or enter `0.0.0.0/0` to allow all IPv4 addresses.

If your Elasticsearch instance is deployed in the China (Hangzhou) region, then you can add IPv6 addresses and CIRD blocks to the whitelist in the format of `2401:b180:1000:24::5` and `2401:b180:1000::/48`, respectively. Enter `::1` to forbid all IPv6 addresses or enter `::/0` to allow all IPv6 addresses.



### Note:

The Kibana console can only be accessed from an ECS instance connected to the same VPC network as the Elasticsearch instance.

## VPC whitelist

You can add IP addresses and CIDR blocks to the VPC whitelist in the format of `192.168.0.1` and `192.168.0.0/24`, respectively. Separate them with commas (,). You can enter `127.0.0.1` to forbid all IPv4 addresses or enter `0.0.0.0/0` to allow all IPv4 addresses.



### Note:

- By default, the VPC whitelist allows all IPv4 addresses.
- The VPC whitelist is used to control access from internal network addresses in VPC networks.

## 10.4.9 Snapshots

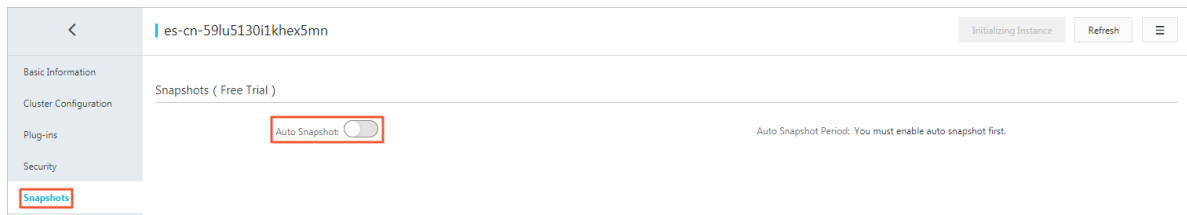
### 10.4.9.1 Auto snapshot

Currently, the snapshots feature is a preview version which is free to use. This feature allows you to periodically create snapshots.

1. Log on to the Apsara Stack Elasticsearch console.
2. Click the ID of the target Elasticsearch instance to navigate to the Elasticsearch instance information page.
3. In the left-side navigation pane, click Snapshots.

4. On the Snapshots page, toggle on the Auto Snapshot switch.

Figure 10-8: Auto snapshot

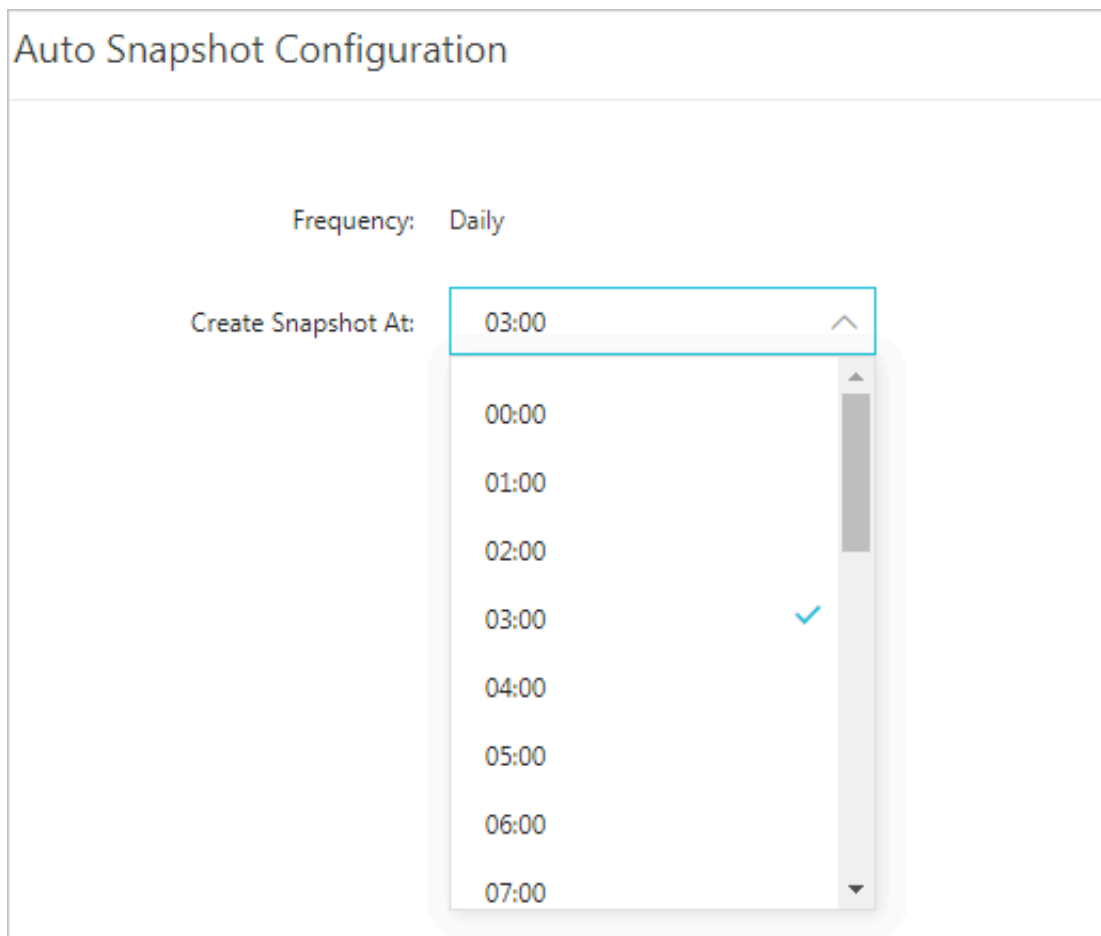


**Note:**

Auto snapshot uses the system time in the current region. Do not perform any snapshot operations when the system is creating snapshots.

5. Click Modify Configuration in the upper-right corner to set the time when the daily snapshot is created.

Figure 10-9: Modify configuration



**Note:**

The auto snapshot feature creates snapshots on a daily basis. You can set the specific snapshot creation time from 0 to 23 hours.

### 10.4.9.2 View snapshot status

After you have enabled the auto snapshot feature, you can call the Elasticsearch `snapshot` operation in the Dev Tools of the Kibana console to view the status of the snapshots. We have integrated the Kibana console into Apsara Stack Elasticsearch.

View all snapshots

You can call the following operation to view all snapshots stored in the `aliyun_auto_snapshot` repository:

```
GET _snapshot/aliyun_auto_snapshot/_all
```

The following result is returned:

```
{
 "snapshots": [
 {
 "snapshot": "<yourSnapshotName>",
 "uuid": "n7YIayyZTm2hwg8BeW****",
 "version_id": 5050399,
 "version": "5.5.3",
 "indices": [
 ".kibana"
],
 "state": "SUCCESS",
 "start_time": "2018-06-28T01:22:39.609Z",
 "start_time_in_millis": 1530148959609,
 "end_time": "2018-06-28T01:22:39.923Z",
 "end_time_in_millis": 1530148959923,
 "duration_in_millis": 314,
 "failures": [],
 "shards": {
 "total": 1,
 "failed": 0,
 "successful": 1
 }
 },
 {
 "snapshot": "<yourSnapshotName>",
 "uuid": "frdl1YFzQ5Cn5xN9ZW****",
 "version_id": 5050399,
 "version": "5.5.3",
 "indices": [
 ".kibana"
],
 "state": "SUCCESS",
 "start_time": "2018-06-28T01:25:00.764Z",
 "start_time_in_millis": 1530149100764,
 "end_time": "2018-06-28T01:25:01.482Z",
 "end_time_in_millis": 1530149101482,
 "duration_in_millis": 718,
 "failures": [],
 "shards": {
```

```

 "total": 1,
 "failed": 0,
 "successful": 1
 }
]
 }
}

```

The **state** field indicates the status of a snapshot. A snapshot can be in one of the following statuses:

- **IN\_PROGRESS**: The snapshot is being created.
- **SUCCESS**: The snapshot task is completed and all shards are stored successfully.
- **FAILED**: The snapshot task is completed, but it fails to store the data.
- **PARTIAL**: The cluster data is stored, but at least one shard fails to be stored.
- **INCOMPATIBLE**: The snapshot is incompatible with the Elasticsearch instance version.

View a specified snapshot

You can call the following operation to view all snapshots stored in the repository **aliyun\_auto\_snapshot**:

GET `_snapshot/aliyun_auto_snapshot/<snapshot>/_status`

- **<snapshot>**: replace it with the name of the snapshot, for example, `<yourSnapshotName>`.

The following result is returned:

```

{
 "snapshots": [
 {
 "snapshot": "<yourSnapshotName>",
 "repository": "aliyun_auto_snapshot",
 "uuid": "n7YIayyZTm2hwg8BeW****",
 "state": "SUCCESS",
 "shards_stats": {
 "initializing": 0,
 "started": 0,
 "finalizing": 0,
 "done": 1,
 "failed": 0,
 "total": 1
 },
 "stats": {
 "number_of_files": 4,
 "processed_files": 4,
 "total_size_in_bytes": 3296,
 "processed_size_in_bytes": 3296,
 "start_time_in_millis": 1530148959688,
 "time_in_millis": 77
 }
 }
],
}

```



```

 "indices": {
 ".kibana": {
 "shards_stats": {
 "initializing": 0,
 "started": 0,
 "finalizing": 0,
 "done": 1,
 "failed": 0,
 "total": 1
 },
 "stats": {
 "number_of_files": 4,
 "processed_files": 4,
 "total_size_in_bytes": 3296,
 "processed_size_in_bytes": 3296,
 "start_time_in_millis": 1530148959688,
 "time_in_millis": 77
 },
 "shards": {
 "0": {
 "stage": "DONE",
 "stats": {
 "number_of_files": 4,
 "processed_files": 4,
 "total_size_in_bytes": 3296,
 "processed_size_in_bytes": 3296,
 "start_time_in_millis": 1530148959688,
 "time_in_millis": 77
 }
 }
 }
 }
 }
 }
}

```

### 10.4.9.3 Restore data from snapshots

If you have enabled the snapshot feature for an Apsara Stack Elasticsearch instance, then the system automatically creates snapshots for the Elasticsearch instance.

You can call the Elasticsearch `snapshot` API operation to restore data from the snapshots.

When you use snapshots, follow these guidelines:

- The first snapshot is a complete copy created on the Elasticsearch instance. The subsequent snapshots are created on the incremental data of the Elasticsearch instance. Therefore, the process of creating the first snapshot is time-consuming. However, it only takes a short period of time to create subsequent snapshots.
- Apsara Stack Elasticsearch only retains snapshots that are created within the last five days.
- A snapshot does not store monitoring data generated by an Elasticsearch instance, such as the `.monitoring` and `.security_audit` indexes.

- A snapshot repository is automatically created when the system creates the first snapshot.

View all repositories

You can call the following operation to view all repositories:

GET `_snapshot`

The following result is returned:

```
{
 "aliyun_auto_snapshot": {
 "type": "oss",
 "settings": {
 "compress": "true",
 "base_path": "xxxx",
 "endpoint": "xxxx"
 }
 }
}
```

- `aliyun_auto_snapshot`: the name of the repository.
- `type`: the storage where the snapshots are stored. In this example, Alibaba Cloud Object Storage Service (OSS) is used.
- `compress: true`: enables compression of the metadata files of the snapshots.
- `base_path`: the location of the snapshots.
- `endpoint`: the endpoint of the OSS bucket.

### Default parameters

The auto snapshots feature also supports the following parameters that are not displayed:

- `max_snapshot_bytes_per_sec: 40mb`: throttles per node snapshot rate. The default snapshot rate is 40 MB per second.
- `max_restore_bytes_per_sec: 40mb`: throttles per node restore rate. The default restore rate is 40 MB per second.
- `chunk_size: Max 1Gb`: large files can be broken into smaller chunks during the snapshot process if needed. The maximum size of a chunk is 1 GB.

View all snapshots

**You can call the following operation to view all snapshots stored in the repository**

`aliyun_auto_snapshot:`

`GET _snapshot/aliyun_auto_snapshot/_all`

**The following result is returned:**

```
{
 "snapshots": [
 {
 "snapshot": "<yourSnapshotName>",
 "uuid": "MMRniVLPRaiawSCm8D****",
 "version_id": 5050399,
 "version": "5.5.3",
 "indices": [
 "index_1",
 ".security",
 ".kibana"
],
 "state": "SUCCESS",
 "start_time": "2018-06-27T01:16:01.009Z",
 "start_time_in_millis": 1530062161009,
 "end_time": "2018-06-27T01:16:05.632Z",
 "end_time_in_millis": 1530062165632,
 "duration_in_millis": 4623,
 "failures": [],
 "shards": {
 "total": 12,
 "failed": 0,
 "successful": 12
 }
 }
]
}
```

Restore index data from a snapshot

**You can call the `_restore` operation to restore index data from a snapshot.**

- **Call the following operation to restore all indexes from a specified snapshot stored in the `aliyun_auto_snapshot` repository. The restore task is executed in the background.** `POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore`  
`<snapshot>`: **replace it with the name of the snapshot, for example, `<yourSnapshotName>`.**

- Call the following operation to restore all indexes from the specified snapshot stored in the `aliyun_auto_snapshot` repository, and receive a response after the restore task is completed:

The `_restore` operation runs restore tasks in the background. The Elasticsearch instance will return a response immediately if the restore operation is executable. You can set the `wait_for_completion` parameter in the request to require the Elasticsearch instance to return the response only after the restore task is completed.

```
POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore? wait_for_completion=true
```

`<snapshot>`: replace it with the name of the snapshot, for example, `<yourSnapshotName>`.

- Call the following operation to restore the specified indexes from a specific snapshot stored in the `aliyun_auto_snapshot` repository, and rename the restored indexes. The restore task is executed in the background.

```
POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore
{
 "indices": "index_1",
 "rename_pattern": "index_(.+)",
 "rename_replacement": "restored_index_$1"
}
```

- `<snapshot>`: replace it with the name of the snapshot, for example, `<yourSnapshotName>`.
- `indices`: specifies the indexes that you want to restore.
- `rename_pattern`: uses a regular expression to match the restored indexes. This parameter is optional.
- `rename_replacement`: uses a pattern to rename the matching indexes. This parameter is optional.

## 10.5 Document operations

### 10.5.1 Create a document

This topic describes how to use Elasticsearch to create a document.

[Connect to an Elasticsearch instance](#), and call the POST operation to create a document on the instance.

```
curl -XPOST http://<HOST>:<PORT>/my_index/my_type -d '{"title": "One", "tags": ["ruby"]}'
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the document.



#### Note:

- Each document has an ID and type. The response contains the ID and type of the document. If you do not specify an ID or type when you create a document, the system will automatically specify one for the document.
- If you have enabled [Auto Create Index](#) and the specified index name does not exist, the system automatically creates the index when it creates the document. By default, auto-indexing is disabled.

If the document is created, the following response is returned:

```
{
 "_index": "my_index",
 "_type": "my_type",
 "_id": "AV4JIVI15ny3i8DCdK1H",
 "_version": 1,
 "result": "created",
 "_shards": {
 "total": 2,
 "successful": 1,
 "failed": 0,
 },
 "created": true
}
```

```
}
```

## 10.5.2 Update a document

This topic describes how to use Elasticsearch to update a document.

*Connect to an Elasticsearch instance*, and then call the POST operation to update a document on the Elasticsearch instance.

```
curl -XPOST http://<HOST>:<PORT>/my_index/my_type/<doc_id>
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the document.
- **<doc\_id>**: the ID of the document.

### Example:

```
curl -XPOST http://<HOST>:<PORT>/my_index/my_type/AV4JIVI15NY3I8DCDK1H
-d '{"title": "FOUR UPDATED", "TAGS": ["RUBY", "PHP"]}'
```

If the document is updated, the following response is returned:

```
{
 "_index": "my_index",
 "_type": "my_type",
 "_id": "AV4JIVI15ny3i8DCdK1H",
 "_version": 2,
 "result": "updated",
 "_shards": {
 "total": 2,
 "successful": 1,
 "failed": 0,
 },
 "created": false
}
```

You can also call the bulk operation to update multiple documents.

## 10.5.3 Retrieve a document

This topic describes how to use Elasticsearch to retrieve a document.

*Connect to an Elasticsearch instance*, and then call the GET operation to retrieve a document.

**Example:**

```
curl -XGET http://<HOST>:<PORT>/my_index/my_type/AV4JIVI15NY3I8CDK1H
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the document.
- **AV4JIVI15NY3I8CDK1H**: the ID of the document. In this example, the document ID is AV4JIVI15NY3I8CDK1H.

If the document is retrieved, the following response is returned:

```
{
 "_INDEX" : "MY_INDEX",
 "_TYPE" : "MY_TYPE",
 "_ID" : "AV4JIVI15NY3I8CDK1H",
 "_VERSION" : 2,
 "_EXISTS" : TRUE,
 "_SOURCE" : {
 "TITLE": "FOUR UPDATED", "TAGS": ["RUBY", "PHP"]
 }
}
```

## 10.5.4 Search documents

This topic describes how to use Elasticsearch to search documents.

*Connect to an Elasticsearch instance*, call the GET or POST operation to search documents.

You can set URI parameters to specify a query string.

```
curl -XGET http://<HOST>:<PORT>/_search
curl -XGET http://<HOST>:<PORT>/{index_name}/_search
curl -XGET http://<HOST>:<PORT>/{index_name}/{type_name}/_search
```

To search for documents in which the `title` field contains the `T` keyword, send the following request:

```
curl -XGET http://<HOST>:<PORT>/my_index/my_type/_search?q=title:T*
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the documents.

## 10.5.5 Complex searches

This topic describes how to use Elasticsearch to perform complex searches.

[Connect to an Elasticsearch instance](#), and call the POST operation to perform a complex search on the documents as follows:

```
$ curl -XPOST http://<HOST>:<PORT>/my_index/my_type/_search?pretty=true -d '{
 "query": {
 "query_string": {"query": "*"}
 },
 "facets": {
 "tags": {
 "terms": {"field": "tags"}
 }
 }
}'
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the documents.



### Note:

Set the `? pretty=true` parameter to make the response more readable.

## 10.5.6 Delete documents

This topic describes how to use Elasticsearch to delete documents.

[Connect to an Elasticsearch instance](#), and then call the DELETE operation to delete documents.

Delete a document with a specified ID

```
curl -XDELETE http://<HOST>:<PORT>/my_index/my_type/{ID}
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the document.
- **ID**: the ID of the document.



Delete a specified type of documents

```
curl -XDELETE http://<HOST>:<PORT>/my_index/my_type/
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.
- **my\_type**: the type of the documents.

Delete documents in a specified index

```
curl -XDELETE http://<HOST>:<PORT>/{my_index}
```

- **<HOST>**: the internal network endpoint of the Elasticsearch instance.
- **<PORT>**: the internal network port of the Elasticsearch instance. The default port is 9200.
- **my\_index**: the name of the index.

## 10.6 Snapshots and restoration

You can use the `_snapshot` operation to back up your Elasticsearch cluster. The operation obtains the current status information and data of your cluster and saves them to a shared repository. This backup process is intelligent.

The first snapshot is a complete copy of data in the cluster. All subsequent snapshots only save the difference between existing snapshots and the new data. As you create snapshots for data from time to time, the backups are incrementally added and deleted. It means that the number of subsequent backups increases quite fast because only a very small amount of data is recorded.

### 10.6.1 Create a repository

```
PUT _snapshot/my_backup
{
 "type": "oss",
 "settings": {
 "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",
 "access_key_id": "xxxx",
 "secret_access_key": "xxxxxx",
 "bucket": "xxxxxx",
 "compress": true
 }
}
```

```
}
```

- **endpoint:** The OSS bucket must be in the same region as your Elasticsearch cluster. The endpoint must be an internal URL in this region. For more information about how to obtain OSS endpoints, see the endpoint section in *Apsara Stack OSS API Reference*.
- **bucket:** must be an existing OSS bucket.

If a large amount of data is to be uploaded, you can set the part size limit. If the data size is greater than this limit, data is uploaded to the OSS bucket in the multipart mode.

```
POST _snapshot/my_backup/
{
 "type": "oss",
 "settings": {
 "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",
 "access_key_id": "xxxx",
 "secret_access_key": "xxxxxx",
 "bucket": "xxxxxx",
 "chunk_size": "500mb",
 "base_path": "snapshot/"
 }
}
```



#### Note:

- **POST \_snapshot/my\_backup/:** Note that POST is used here instead of PUT. The repository settings are modified in this way.
- **base\_path:** the starting position of the repository. It is the root directory by default.

## 10.6.2 Obtain repository information

Run the following command to obtain repository information:

```
GET _snapshot
```

Run the following command to obtain the information of a specific repository:

```
GET _snapshot/my_backup
```

## 10.6.3 Migrate a snapshot

Perform the following steps to migrate a snapshot to another cluster:

1. Back up the snapshot to an OSS bucket.

2. Register a snapshot repository (in the same OSS bucket) in the new cluster.
3. Set `base_path` to the directory of the backup file.
4. Run the related command to restore data from the backup file.

### 10.6.4 Create a snapshot for all running indexes

A repository can contain multiple snapshots. Each snapshot is related to a series of indexes, such as all indexes, several indexes, or a simple index. You can specify which indexes you want to create a snapshot for and set a unique name for the snapshot.

The basic command to create a snapshot is as follows:

```
PUT _snapshot/my_backup/snapshot_1
```

When the preceding command is executed, a snapshot is created for all running indexes and saved in the *my\_backup* repository as *snapshot\_1*. A response for the call request is immediately returned, and the snapshot process runs in the background.



#### Note:

Typically, you want your snapshots to run as background processes. If you want to wait until the execution finishes, add the `wait_for_completion` tag in the script.

```
PUT _snapshot/my_backup/snapshot_1? wait_for_completion=true
```

This command blocks the call request until the snapshot is created. For large snapshots, this period may be very long.

### 10.6.5 Create a snapshot for a specific index

When a snapshot is created, all running indexes are backed up by default. If Kibana is used and you do not want to back up all *.kibana* indexes, you can back up only the specified indexes when creating a snapshot for your cluster.

```
PUT _snapshot/my_backup/snapshot_2
{
 "indices": "index_1,index_2"
}
```

Only *index1* and *index2* are backed up when the preceding command is executed.

## 10.6.6 Obtain snapshot information

Sometimes you may forget details about snapshots in the repository, especially when snapshot names are based on the creation time, such as `backup_2014_10_28`. You can run the command to view the information of a snapshot.

You can initiate a GET request with `repository/snapshot_name` specified.

```
GET _snapshot/my_backup/snapshot_2
```

The following output is displayed:

```
{
 "snapshots": [
 {
 "snapshot": "snapshot_1",
 "indices": [
 ".marvel_2014_28_10",
 "index1",
 "index2"
],
 "state": "SUCCESS",
 "start_time": "2014-09-02T13:01:43.115Z",
 "start_time_in_millis": 1409662903115,
 "end_time": "2014-09-02T13:01:43.439Z",
 "end_time_in_millis": 1409662903439,
 "duration_in_millis": 324,
 "failures": [],
 "shards": {
 "total": 10,
 "failed": 0,
 "successful": 10
 }
 }
]
}
```

To obtain the information of all snapshots in a repository, replace the snapshot name with the `_all` parameter.

```
GET _snapshot/my_backup/_all
```

## 10.6.7 Delete a snapshot



### Notice:

- If you delete snapshots manually, the backups may be seriously damaged because the deleted snapshots may contain data that is still being used.
- We recommend that you use the operation to delete snapshots. Snapshots are incremental and many snapshots depend on historical snapshots. The Delete

operation can determine the data that is still being used by recent snapshots and delete only those snapshots that are no longer used.

To delete a snapshot that is no longer used, you can use the Delete operation with `repository/snapshot_name` specified to initiate an HTTP-based call request.

```
DELETE _snapshot/my_backup/snapshot_2
```

## 10.6.8 Monitor snapshot progress

`wait_for_completion` tag provides the basic monitoring mode. However, it may be insufficient if you want to restore data in a medium-sized cluster from a snapshot. You can use the following methods to view more details about a specific snapshot.

- Initiate a GET request with a snapshot ID.

```
GET _snapshot/my_backup/snapshot_3
```

If the snapshot is still running when you run the preceding command, you can see more of its information, such as the start time and elapsed time.



### Note:

This operation uses the same thread pool as snapshots. If your snapshot contains very large shards, the status update interval is long, because the same thread pool is used.

- A better alternative is to pull data by using the `_status` operation.

```
GET _snapshot/my_backup/snapshot_3/_status
```

The output of the `_status` operation:

```
{
 "snapshots": [
 {
 "snapshot": "snapshot_3",
 "repository": "my_backup",
 "state": "IN_PROGRESS",
 "shards_stats": {
 "initializing": 0,
 "started": 1,
 "finalizing": 0,
 "done": 4,
 "failed": 0,
 "total": 5
 },
 "stats": {
 "number_of_files": 5,
 "processed_files": 5,
 "total_size_in_bytes": 1792,
 "processed_size_in_bytes": 1792,

```

```
 "start_time_in_millis": 1409663054859,
 "time_in_millis": 64
 },
 "indices": {
 "index_3": {
 "shards_stats": {
 "initializing": 0,
 "started": 0,
 "finalizing": 0,
 "done": 5,
 "failed": 0,
 "total": 5
 },
 "stats": {
 "number_of_files": 5,
 "processed_files": 5,
 "total_size_in_bytes": 1792,
 "processed_size_in_bytes": 1792,
 "start_time_in_millis": 1409663054859,
 "time_in_millis": 64
 },
 "shards": {
 "0": {
 "stage": "DONE",
 "stats": {
 "number_of_files": 1,
 "processed_files": 1,
 "total_size_in_bytes": 514,
 "processed_size_in_bytes": 514,
 "start_time_in_millis": 1409663054862,
 "time_in_millis": 22
 }
 }
 }
 },
 },
```

...

- **state:** The state of a running snapshot is `IN_PROGRESS`.
- **started:** One shard of this snapshot is still being transmitted (the other four shards have been transmitted).

The response contains the overall information of the snapshot and statistics on each drilling-down index and shard. Different shards of the snapshot can be in different states.

- **initializing:** The shard is checking the cluster status to determine whether the snapshot task can be processed. This process is generally very short.
- **started:** Data is being transmitted to the repository.
- **finalizing:** Data has been transmitted and the shard is sending the snapshot metadata.
- **done:** The snapshot task is completed for the shard.
- **failed:** An error occurs when the snapshot is being created. The snapshot task for this shard, index, or snapshot cannot be completed. You can check the log for more information.

### 10.6.9 Cancel a snapshot

Creating a snapshot is a time-consuming process and consumes valuable resources. It takes a long time to resolve even a minor error when you are creating a snapshot. You can cancel a snapshot when it is running.

To cancel a snapshot, you can just delete it.

```
DELETE _snapshot/my_backup/snapshot_3
```

After the preceding command is executed, the snapshot process is interrupted and the running snapshot is deleted from the repository.

## 10.6.10 Restore data from a snapshot

You can restore data from a snapshot after you create a snapshot from a backup.

Add the `_restore` parameter after the ID of the snapshot that you want to use in restoration.

```
POST _snapshot/my_backup/snapshot_1/_restore
```

All indexes in this snapshot are restored by default. If `snapshot_1` contains five indexes, all the five indexes are restored to the cluster. You can also choose to restore a specific index, just like in the `_snapshot` operation.

Rename an index

You can use an additional option to rename an index. The option allows you to use a method to match the index name and rename the index through the restoration process. If you want to restore historical data to verify the content or perform other operations without replacing existing data, this option also can be used.

The following example shows how to restore a single index from a snapshot and rename it:

```
POST /_snapshot/my_backup/snapshot_1/_restore
{
 "indices": "index_1",
 "rename_pattern": "index_(.+)",
 "rename_replacement": "restored_index_$1"
}
```

- `indices`: restores only the specified index and ignores other indexes. Only `index_1` is restored in this example.
- `rename_pattern`: searches for the index being restored, which needs to match the provided mode.
- `rename_replacement`: renames the index, which uses the new mode.

In the preceding example, you restore `index_1` to you cluster and rename it `restored_index_1`.

Like the snapshot operation, a response is immediately returned for the `restore` command and the restoration process runs in the background. If you want HTTP



call requests to be blocked until the restoration process is completed, add the `wait_for_completion` tag:

```
POST _snapshot/my_backup/snapshot_1/_restore? wait_for_completion=true
```

### 10.6.11 Monitor the restoration operation

The existing restoration mechanism of Elasticsearch is used to restore data from the repository. For internal implementation, shard restoration from a repository is equivalent to the restoration from another node.

To monitor the restoration progress, call the `_recovery` operation. This operation is for general purpose and is used to display the status of moving shards in your cluster.

- You can call this option on its own to monitor the restoration operation on a specific index:

```
GET restored_index_3/_recovery
```

- You can also use this operation to view all indexes in your cluster, including moving shards that are unrelated to the restoration process:

```
GET /_recovery/
```

A similar output is displayed (note that a large quantity of content may be output if your cluster is highly active):

```
{
 "restored_index_3" : {
 "shards" : [{
 "id" : 0,
 "type" : "snapshot",
 "stage" : "index",
 "primary" : true,
 "start_time" : "2014-02-24T12:15:59.716",
 "stop_time" : 0,
 "total_time_in_millis" : 175576,
 "source" : {
 "repository" : "my_backup",
 "snapshot" : "snapshot_3",
 "index" : "restored_index_3"
 },
 "target" : {
 "id" : "ryqJ5l05S4-lSFbGntkEkg",
 "hostname" : "my.fqdn",
 "ip" : "10.0.1.7",
 "name" : "my_es_node"
 },
 "index" : {
 "files" : {
 "total" : 73,
 "reused" : 0,
```

```

 "recovered" : 69,
 "percent" : "94.5%"
 },
 "bytes" : {
 "total" : 79063092,
 "reused" : 0,
 "recovered" : 68891939,
 "percent" : "87.1%"
 },
 "total_time_in_millis" : 0
 },
 "translog" : {
 "recovered" : 0,
 "total_time_in_millis" : 0
 },
 "start" : {
 "check_index_time" : 0,
 "total_time_in_millis" : 0
 }
 }
}
]
}

```

- **type:** indicates the restoration type. The shard is restored from a snapshot in this example.
- **source:** indicates the snapshot and repository from which the shard is restored.
- **percent:** indicates the restoration status. The specified shard is currently 94% restored. The restoration task will soon be finished.

All indexes that are being restored and all shards in these indexes are listed in the output. Statistics on the start and end time, duration, restoration progress in percentage, and number of transmitted bytes are displayed for each shard.

### 10.6.12 Cancel a restoration task

You can delete an index that is being restored to cancel a restoration task. You can modify the cluster status by calling the Delete operation to stop a restoration process. Example:

```
DELETE /restored_index_3
```

If *restored\_index\_3* is being restored, after you run the delete command, the restoration process stops and all the data that has been restored to the cluster is deleted.

## 10.7 Elasticsearch test

After you create an Elasticsearch instance, you can log on to the Kibana console integrated into the Elasticsearch console and test the search function on the Dev

**Tools page. You can also run the curl command in an ECS instance that meets the requirements to perform the test.**

### 10.7.1 Use curl to connect to Elasticsearch through port 9200

**The following example shows how to connect to Elasticsearch with the username and password specified:**

```
curl -u username:password 'http://<HOST>:9200/filebeat/my_type/'?pretty -d '{"title": "One", "tags": ["ruby"]}'
```

**The following response is returned:**

```
{
 "_index" : "filebeat",
 "_type" : "my_type",
 "_id" : "AV-bTkaTwdiHxfaSqlAt",
 "_version" : 1,
 "result" : "created",
 "_shards" : {
 "total" : 2,
 "successful" : 2,
 "failed" : 0
 },
 "created" : true
}
```

**The following example shows how to connect to Elasticsearch without the username and password:**

```
curl http://<HOST>:9200/my_index/my_type -XPOST -d '{"title": "One", "tags": ["ruby"]}'
```

### 10.7.2 Use Python to connect to Elasticsearch through port 9200

```
from elasticsearch import Elasticsearch, RequestsHttpConnection
import certifi
es = Elasticsearch(
 ['<HOST>'],
 http_auth=('username', 'password'),
 port=9200,
 use_ssl=False
)
res = es.index(index="my_index", doc_type="my_type", id=1, body={"title": "One", "tags": ["ruby"]})
print(res['_source'])
res = es.get(index="my-index", doc_type="my-type", id=1)
print(res['_source'])
```

## 10.7.3 Use Java REST client to connect to Elasticsearch through port 9200

### Considerations

- **The official Elasticsearch team no longer maintains the TransportClient. We recommend that you do not use the TransportClient to connect to Apsara Stack Elasticsearch. A NoNodeAvailableException error occurs when you use TransportClient 5.5.3 to connect to Apsara Stack Elasticsearch. We recommend that you use the [Java Low-Level REST client](#) to connect to Apsara Stack Elasticsearch.**
- **The Java REST client described in this topic is only compatible with Apsara Stack Elasticsearch V5.5.3. It is incompatible with Apsara Stack Elasticsearch V6.3.2. If you are using Apsara Stack Elasticsearch V6.3.2, reference the official Elasticsearch documentation [Java REST client 6.3.2](#).**
- **The version of the Java REST client must be the same as the Elasticsearch instance version.**

### Prerequisites

- **Create an Apsara Stack Elasticsearch instance and enable auto-indexing.**
- **Install the JDK and configure environment variables. The JDK version must be 1.8 or later.**

### Sample code

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.entity.ContentType;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.entity.NStringEntity;
import org.apache.http.util.EntityUtils;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import java.io.IOException;
import java.util.Collections;
public class RestClientTest {
 public static void main(String[] args){
 final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
 credentialsProvider.setCredentials(AuthScope.ANY,
 new UsernamePasswordCredentials("<USER NAME>", "<
PASSWORD>"));
```

```

 RestClient restClient = RestClient.builder(new HttpHost("<HOST>", 9200))
 .setHttpClientConfigCallback(new RestClientBuilder.
 HttpClientConfigCallback() {
 @Override
 public HttpAsyncClientBuilder customizeHttpClient(
 HttpAsyncClientBuilder httpClientBuilder) {
 return httpClientBuilder.setDefaultCredentialsProvider(
 credentialsProvider);
 }
 }).build();

 try {
 //index a document
 HttpEntity entity = new NStringEntity("{\n\"user\" : \"kimchy\"\n}",
 ContentType.APPLICATION_JSON);
 Response indexResponse = restClient.performRequest("PUT",
 "/index/type/123",
 Collections.<String, String>emptyMap(),
 entity);
 //search a document
 Response response = restClient.performRequest("GET", "/"
 + "index/type/123",
 Collections.singletonMap("pretty", "true"));
 System.out.println(EntityUtils.toString(response.getEntity()));
 } catch (IOException e) {
 e.printStackTrace();
 }
 }
}

```

- **<USER NAME>: replace USER NAME with the username of your Elasticsearch instance.**
- **<PASSWORD>: replace PASSWORD with the password of your Elasticsearch instance.**
- **<HOST>: replace HOST with the public or internal network endpoint shown on the Basic Information page of your Elasticsearch instance.**

# 11 DataHub

---

## 11.1 What is DataHub?

**DataHub is a real-time data distribution platform designed to process streaming data.**

**You can publish and subscribe to applications for streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data.**

**DataHub collects, stores, and processes streaming data from mobile devices, applications, website services, and sensors. You can use your own applications or Alibaba Cloud Realtime Compute to process streaming data in DataHub, such as real-time website access logs, application logs, and events. The processing results such as alerts and statistics presented in graphs and tables are updated in real time.**

**Based on the Apsara system of Alibaba Cloud, DataHub features high availability, low latency, high scalability, and high throughput. DataHub is seamlessly integrated with Realtime Compute, allowing you to use SQL to analyze streaming data.**

**DataHub also supports synchronizing streaming data to various Alibaba Cloud services such as MaxCompute and OSS.**

**The features of DataHub are described as follows:**

- **Data queue:** DataHub automatically generates a cursor for each record in a shard. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number shards in the topic.
- **Checkpoint-based data restoration:** DataHub supports saving checkpoints in the system. You can restore data from any checkpoint you saved when your application fails.
- **Data synchronization:** Data in DataHub can be automatically synchronized to other Alibaba Cloud platforms, including MaxCompute, Object Storage Service (OSS), AnalyticDB, ApsaraDB RDS for MySQL, Table Store, and Elasticsearch.
- **Scalable topics:** DataHub allows you to scale in or out the topic by splitting a shard into two or merging two shards.

## 11.2 Limits

Before using DataHub, learn the limits on certain features.

The following table describes the limits.

Table 11-1: Limits

Item	Range	Description
Active shards	(0,10]	Each topic can contain up to 10 active shards.
Shards	(0,512]	You can create up to 512 shards in each topic.
HTTP body size	$\leq 4$ MB	The HTTP request body size cannot exceed 4 MB.
String size	$\leq 1$ MB	The size of a string cannot exceed 1 MB.
Merge and split operations on new shards	$\geq 5$ s	You cannot merge a shard with another shard or split the shard in less than 5 seconds after it is created.
Queries per second (QPS)	$\leq 1,000$	The write QPS limit for each shard is 1,000. Multiple queries in one batch are considered one query.
Throughput	$\leq 1$ MB/s	Each shard provides a throughput of up to 1 MB/s.
Projects	$\leq 5$	You can create up to 5 projects with each account.
Topics	$\leq 20$	You can create up to 20 topics in each project. Contact the administrator if you need to create more topics.

Item	Range	Description
Time-to-live of records	[1,7]	The time-to-live of each record in the topic is from 1 to 7 days.

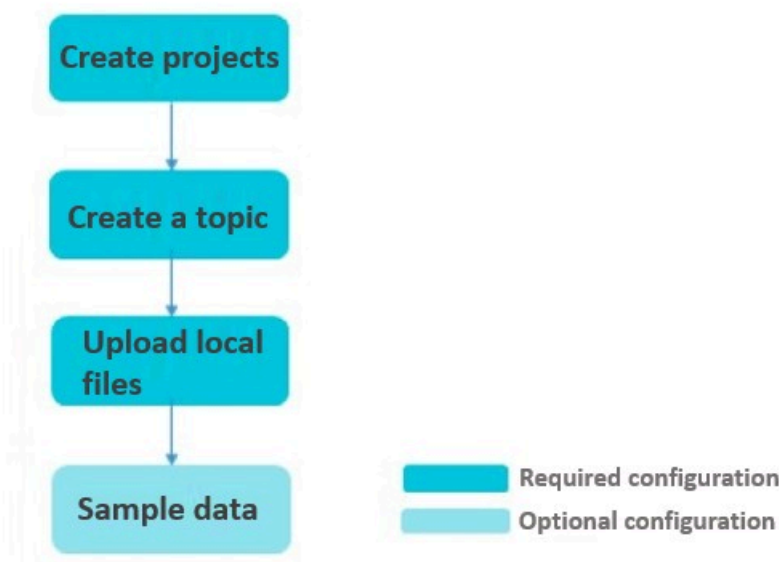
## 11.3 Quick Start

### 11.3.1 Overview

This topic describes the procedure about how to use DataHub.

*Figure 11-1: Procedure* shows the procedure.

Figure 11-1: Procedure



- *Create projects*

A project is an organizational unit in DataHub and contains one or more topics. When using DataHub, you must create a project first.

- *Create a topic*

A topic is the smallest unit for data subscription and publishing. You can use topics to distinguish different types of streaming data.

- *Upload local files*

DataHub allows you to upload local files such as a TXT file to the system. You no longer need to deploy log collection or use SDKs to parse the data.



- [Sample data](#)

DataHub supports data sampling. You can sample data of a specific shard.

### 11.3.2 Log on to the DataHub console

This topic describes how to log on to the DataHub console by using the Google Chrome browser.

#### Prerequisites

Before logging on to the DataHub console, confirm the following information:

- You have obtained the IP address or domain name that is required to access the Apsara Stack console.

For example, the logon address for Apsara Stack is `http://x.x.x.x/manage`, and `x.x.x.x` indicates the IP address or server domain name.

- You have upgraded your Google Chrome browser to version 42.0.0 or later.

#### Procedure

1. Open your Google Chrome browser.
2. In the address bar, enter the *logon address for Apsara Stack* and press Enter.
3. Enter the username and password of your account.
  - By default, an administrator account is set in the system. The username is `super`. The administrator can create system users and notify the users of their default passwords through SMS messages and e-mails.
  - When you log on to the Apsara Stack console for the first time, you must modify the password as instructed. To ensure the security of your account, the password must meet the minimum password requirements. It must contain at least two types of the following characters: letters (A-Z or a-z), numbers (0-9), or special characters, such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%). The password must be 8 to 20 characters in length.
4. Click Log On to enter the Dashboard page.
5. In the left-side navigation pane, choose Big Data > DataHub.

6. Select a Region and a Department, and then click DataHub. Enter the AccessKey of the selected department to log on to the DataHub console.



**Note:**

You can perform the following operations to obtain an AccessKey.

To obtain the AccessKey of a department, follow these steps:

- a. Log on to the Apsara Stack console as an administrator.
- b. In the left-side navigation pane, choose User Center > Department Management.
- c. On the Department Management page that appears, select a department for which you want to obtain an AccessKey and click Get AccessKey.



**Note:**

The AccessKey of the level-1 department is automatically generated by the system. Departments at lower levels share the same AccessKey with the level-1 department.

To obtain the AccessKey of a RAM user account, follow these steps:

- a. Log on to the Apsara Stack console as an administrator.
- b. In the upper-right corner of the home page, move the pointer over the user profile picture and click Personal Information.

On the Personal Information page, locate the AccessKey ID in the Alibaba Cloud AccessKey section.

- c. Click Show in the AccessKey Secret column to view the AccessKey Secret.

### 11.3.3 Create projects

A project is an organizational unit in DataHub, which contains one or more topics. When using DataHub, you must create a project first. This section describes how to create a project in the DataHub console.

#### Prerequisites

You must have created an Alibaba Cloud account.

#### Note

- DataHub projects are independent from MaxCompute projects. Projects you created in MaxCompute cannot be used in DataHub.

- **You can create up to five projects with each account.**

#### Procedure

1. **Log on to the DataHub console.**
2. **On the Projects page, click Create Project in the upper-right corner. In the dialog box that appears, set a name and a description for the project and click Create.**



#### Note:

**The Name and Description fields are required.**

### 11.3.4 Create a topic

**A topic is the smallest unit for data subscription and publication. You can use topics to distinguish different types of streaming data. This section describes how to create a topic in the DataHub console.**

#### Prerequisites

**You must have created a project.**

#### Note

**You can create up to 20 topics in each project. Contact the administrator if you need to create more topics.**

#### Procedure

1. **Log on to the DataHub console.**
2. **On the Projects page, click View in the Actions column of a project.**
3. **On the details page of the project, click Create Topic in the upper-right corner.**
4. **In the dialog box that appears, specify the required information and click Create.**

### 11.3.5 Upload local files

**DataHub allows you to upload local files such as a TXT file to the system, eliminating additional effort to parse the data or deploy log collection. This section describes how to upload local files to DataHub in the console.**

#### Prerequisites

**You must have created a project and a topic in the project.**

#### Note

**For more information about the file upload requirements, see the descriptions in the DataHub console.**

#### Procedure

1. **Log on to the DataHub console.**
2. **In the left-side navigation pane, click Data Acquisition. On the Data Acquisition page, select Upload File.**
3. **In the dialog box that appears, configure the information as required and click Upload.**
4. **After the file is uploaded, a message will appear in the upper-right corner.**

### 11.3.6 Sample data

**DataHub supports data sampling. You can sample data of a specific shard. This topic describes how to sample data in the DataHub console.**

#### Prerequisites

**You must have created a project, a topic in the project, and ingested data to the topic.**

#### Background

**Before sampling data, you must specify a time and the maximum number of records you want to sample.**

#### Procedure

1. **Log on to the DataHub console.**
2. **On the Projects page, click View in the Actions column of a project. On the project details page that appears, click View in the Actions column of a topic.**
3. **Then, on the details page of the topic, select Data Sampling in the Actions column of a shard.**
4. **In the dialog box that appears, specify a time and the maximum number of records you want to sample and click Sample. A sample of records that are ingested to the shard after the specified time will be shown in the following table.**

## 11.4 Access Control

### 11.4.1 Overview

DataHub allows you to improve data security by granting different permissions to Apsara Stack tenant accounts and RAM user accounts.

DataHub uses Resource Access Management (RAM) for access control. Only users that have been granted the required permissions can access the resources in your department. By default, users do not have permission to access resources in your department. This topic describes how access control for DataHub is achieved by using RAM.

**Note:**

An Apsara Stack tenant account is owned by a department and requires no authorization. A RAM user account must be granted permissions by the tenant account.

### 11.4.2 DataHub resources in RAM

The DataHub resources in RAM are project, topic, and subscription. Subscription is the action that you specify an application to read and process the records in DataHub. DataHub supports RAM authorization of project, topic, and subscription. RAM authorization is not supported at the shard level.

In RAM, each resource type has a general description and each specific object of the resource type has a description. For example, the description of a project that resides in a certain region is `acs:dhs:$region:$accountid:projects/$projectName`. `$region`, `$accountid`, and `$projectName` indicate the region that the project resides, the user ID, and the project name.

Table 11-2: Resource description

Resource type	Description
SingleProject	<code>acs:dhs:\$region:\$accountid:projects/\$projectName</code>
AllProject	<code>acs:dhs:\$region:\$accountid:projects/*</code>
SingleTopic	<code>acs:dhs:\$region:\$accountid:projects/\$projectName/topics/\$topicName</code>
AllTopic	<code>acs:dhs:\$region:\$accountid:projects/\$projectName/topics/*</code>

Resource type	Description
SingleSubscription	acs:dhs:\$region:\$accountid:projects/\$projectName/topics/\$topicName/subscriptions/\$subId
AllSubscription	acs:dhs:\$region:\$accountid:projects/\$projectName/topics/\$topicName/subscriptions/*

### 11.4.3 API

DataHub provides application programming interfaces (APIs) for projects, topics, shards, subscriptions, and records. Before you can call the API operations, you must grant corresponding permissions to the RAM user by using RAM authorization policies.

The RAM authorization policy and resource type for each API operation is described as follows:

API operations for projects

Table 11-3: API operations for projects

Operation name	RAM authorization policy	Resource type
CreateProject	dhs:CreateProject	AllProject
ListProject	dhs:ListProject	AllProject
DeleteProject	dhs>DeleteProject	SingleProject
GetProject	dhs:GetProject	SingleProject
UpdateProject	dhs: UpdateProject	SingleProject

API operations for topics

Table 11-4: API operations for topics

Operation name	RAM authorization policy	Resource type
CreateTopic	dhs:CreateTopic	AllTopic
ListTopic	dhs:ListTopic	AllTopic
DeleteTopic	dhs>DeleteTopic	SingleTopic
GetTopic	dhs:GetTopic	SingleTopic
UpdateTopic	dhs: UpdateTopic	SingleTopic

## API operations for subscriptions

Table 11-5: API operations for subscriptions

Operation name	RAM authorization policy	Resource type
CreateSubscription	dhs:CreateSubscription	AllSubscription
ListSubscription	dhs:ListSubscription	AllSubscription
DeleteSubscription	dhs>DeleteSubscription	SingleSubscription
GetSubscription	dhs:GetSubscription	SingleSubscription
UpdateSubscription	dhs: UpdateSubscription	SingleSubscription
CommitOffset	dhs:CommitOffset	SingleSubscription
GetOffset	dhs:GetOffset	SingleSubscription

## API operations for shards

Table 11-6: API operations for shards

Operation name	RAM authorization policy	Resource type
ListShard	dhs:ListShard	SingleTopic
MergeShard	dhs:MergeShard	SingleTopic
SplitShard	dhs:SplitShard	SingleTopic

## API operations for shards

Table 11-7: API operations for shards

Operation name	RAM authorization policy	Resource type
PutRecords	dhs:PutRecords	SingleTopic
GetRecords	dhs:GetRecords	SingleTopic
GetCursor	dhs:GetRecords	SingleTopic

## 11.4.4 Conditions

This section describes conditions that can be applied to the RAM authorization policies for DataHub.

Conditions that can be applied to the RAM authorization policies for DataHub are as follows:

Table 11-8: RAM authorization policy conditions for DataHub

Condition keyword	Description	Valid value
<b>acs:SourceIp</b>	The IP address range that can access the specified object.	Any valid IP address. Wildcard masks are supported.
<b>acs:SecureTransport</b>	Indicates whether HTTPS is used to access the specified object.	true/false
<b>acs:MFAPresent</b>	Indicates whether the specified object can be accessed by multiple clients.	true/false
<b>acs:CurrentTime</b>	The time that the specified object can be accessed.	This keyword must be described in ISO 8601 format.

## 11.4.5 Sample RAM authorization policy content

### 11.4.5.1 AliyunDataHubFullAccess

This section describes how to set the AliyunDataHubFullAccess policy content.

The authorization policy content can be set as follows:

```
{
 "Version": "1",
 "Statement": [
 {
 "Action": "dhs:*",
 "Resource": "*",
 "Effect": "Allow"
 }
]
}
```

### 11.4.5.2 AliyunDataHubReadOnlyAccess

This section describes how to set the AliyunDataHubReadOnlyAccess policy content.

The authorization policy content can be set as follows:

```
{
 "Version": "1",
 "Statement": [
 {
 "Action": ["dhs:List*", "dhs:Get*"],
 "Resource": "*",
 "Effect": "Allow"
 }
]
}
```



```
}
]
}
```

## 11.5 Data Acquisition

### 11.5.1 Overview

In addition to SDK and local file uploads, DataHub supports various data acquisition tools to help you quickly collect data to DataHub.

This section describes how to acquire data by using Fluentd, Logstash, and Oracle GoldenGate (OGG).

### 11.5.2 Fluentd

This section describes how to install and use the DataHub agent for Fluentd.

Developed on the basis of an open source data collector Fluentd, the DataHub agent for Fluentd is easy to install and is used to write the collected data to DataHub.

Install DataHub agent for Fluentd

- **Install the agent by using RubyGems:**

```
gem install fluent-plugin-datahub
```



**Notice:**

We recommend that you change the gem source to <https://ruby.taobao.org/>.

- **Install the agent locally**

The agent must be installed in a Linux environment. Before installation, you must have installed Ruby. For users that have not installed Fluentd, an installation package of Fluentd and DataHub agent for Fluentd is provided. For users that have installed Fluentd, an installation package of the DataHub writer is provided.

- If you have not installed Fluentd, click [Fluentd full installation package](#) to download the full installation package and run the following command to install the agent:



**Notice:**

**Version 0.12.23 is provided in the full installation package.**

```
$ tar -xzvf fluentd-with-datahub-0.12.23.tar.gz
$ cd fluentd-with-datahub
$ sudo sh install.sh
```

- If you have installed Fluentd, click [Fluentd DataHub Plug-in package](#) to download the DataHub writer and run the following gem command to install the writer:

```
$ sudo gem install --local fluent-plugin-datahub-0.0.2.gem
```

## Use case

### Case A: Write CSV files into DataHub

This example shows how to write incremental content of a CSV file into DataHub in near real-time by using the Fluentd plugin. The format of the CSV file is as follows:

```
0,qe614c760fuk8judu01tn5x055rpt1,true,100.1,14321111111
1,znv1py74o8ynn87k66o32ao4x875wi,true,100.1,14321111111
2,7nm0mtpgo1q0ubuljjx9b000ybltl,true,100.1,14321111111
3,10t0n6pvonnan16279w848ukko5f6l,true,100.1,14321111111
4,0ub584kw88s6dczd0mta7itmta10jo,true,100.1,14321111111
5,1ltfpf0jt7fhvf0oy4lo8m3z62c940,true,100.1,14321111111
6,zpqsfxqy9379lmcehd7q8kftntrozb,true,100.1,14321111111
7,ce1ga9aln346xcj761c3iytshyzuxg,true,100.1,14321111111
8,k5j2id9a0ko90cykl40s6ojq6gruyi,true,100.1,14321111111
9,ns2zcx9bdip5y0aqd1tdicf7bkdsms,true,100.1,14321111111
10,54rs9cm1xau2fk66pzyz62tf9tsse4,true,100.1,14321111111
```

Each line is a record written into DataHub. Columns are separated by commas (.). The CSV file is saved in the local /temp/test.csv path. The schema of the DataHub topic for the CSV file is as follows:

Table 11-9: DataHub topic schema

Column name	Data type
id	BIGINT
name	STRING
gender	BOOLEAN
salary	DOUBLE
my_time	TIMESTAMP

**Edit the Fluentd configuration file based on the CSV file and DataHub topic schema. Run the following command to start the DataHub agent for Fluentd to write the CSV file into DataHub:**

```
${FLUENTD_HOME}/fluentd-with-dataHub/bin/fluentd -c fluentd_test.conf
```

**The corresponding Fluentd configuration file is as follows:**

```
<source>
 @type tail
 path /xxx/yyy (Specify a file path.)
 tag test1
 format csv
 keys id,name,gender,salary,my_time
</source>

<match test1>
 @type dataHub
 access_id your_app_id
 access_key your_app_key
 endpoint http://ip:port
 project_name test_project
 topic_name fluentd_performance_test_1
 column_names ["id", "name", "gender", "salary", "my_time"]
 flush_interval 1s
 buffer_chunk_limit 3m
 buffer_queue_limit 128
 dirty_data_continue true
 dirty_data_file /xxx/yyy (Specify a path for dirty record files.)
 retry_times 3
 put_data_batch_size 1000
</match>
```

### Case B: Collect Log4j logs

**The format of Log4j logs is as follows:**

```
11:48:43.439 [qtp1847995714-17] INFO AuditInterceptor - [c2un5sh7cu
52ek6am1ui1m5h] end /web/v1/project/tefe4mfurtix9kwwyrvfqd0m/node/
0m0169kapshvgc3ujskwkk8g/health GET, 4061 ms
```

**The corresponding Fluentd configuration file is as follows:**

```
<source>
 @type tail
 path bayes.log
 tag test
 format /(? <request_time>\d\d:\d\d:\d\d.\d+)\s+\[(? <thread_id>[\w
-]+)\]\s+(? <log_level>\w+)\s+(? <class>\w+)\s+-\s+\[(? <request_id>\w+)\]\s+(? <detail>.+)/
</source>

<match test>
 @type dataHub
 access_id your_access_id
 access_key your_access_key
 endpoint http://ip:port
 project_name test_project
```

```
topic_name dataHub_fluentd_out_1
column_names ["thread_id", "log_level", "class"]
</match>
```

## DataHub reader and writer configuration

Table 11-10: Reader configuration

Parameter	Description
<b>tag test1</b>	The tag, which is mapped to the destination information by using the specified regular expression.
<b>format csv</b>	The comma-separated value (CSV) files from where the data is acquired.
<b>keys id,name,gender,salary,my_time</b>	The column names to be acquired must be the same with those in the destination DataHub table.

Table 11-11: Configuration of the writer plugin

Parameter	Description
<b>shard_id 0</b>	The ID of the shard that all records are written into. By default, a polling model is used.
<b>shard_keys ["id"]</b>	The key of the shard. The hash of the key value is mapped to a shard ID.
<b>flush_interval 1</b>	The interval in seconds to wait before invoking the next buffer flush. Default value: 60.
<b>buffer_chunk_limit 3m</b>	The maximum size of a chunk. Unit: KB or MB. We recommend you set the maximum size to 3 MB.
<b>buffer_queue_limit 128</b>	The maximum length of the output queue. The values of <code>buffer_chunk_limit</code> and <code>buffer_queue_limit</code> determine the size of the buffer chunks.
<b>put_data_batch_size 1000</b>	Every 1,000 records are written into DataHub.
<b>retry_times 3</b>	The number of retries.
<b>retry_interval 3</b>	The interval between retries. Unit: seconds.
<b>dirty_data_continue true</b>	Indicates whether to ignore dirty records. False : Retry the operation for a specified number of times before writing the dirty records into the dirty record file.

Parameter	Description
dirty_data_file /xxx/yyy	The directory where the dirty record file is stored .
column_names ["id"]	The name of the columns to be acquired.

### 11.5.3 Logstash

This section describes how to install Logstash, use Logstash to collect logs, and import log data from Logstash into DataHub.

Logstash is a distributed log collection framework. It is often used with Elasticsearch and Kibana, known as the ELK Stack, for log data analysis. Logstash supports the input of more than 30 types of data such as files, syslog, Redis, log4j, Apache logs, and Nginx logs. Logstash provides filter plugins to customize fields. To support a wider variety of data inputs, DataHub offers Output and Input plugins for data transfer with Logstash.

#### Install Logstash

You must install Logstash in Java Runtime Environment (JRE) 7 or later versions. Some features are unavailable in earlier versions. You can either install the DataHub agent for Logstash or use DataHub Input and Output plugins.

- **Install Logstash and the DataHub agent for Logstash:** Click [CxOneKey](#) to download the installation package. Extract the package and Logstash is installed.

Run the following command to install the DataHub agent for Logstash:

```
$ tar -xzvf logstash-with-datahub-2.3.0.tar.gz
$ cd logstash-with-datahub-2.3.0
```

- **Use DataHub Input and Output plugins.**
  - **Install Logstash.** For more information, see the documentation on the official website of Logstash: <https://www.elastic.co/guide/en/logstash/current/index.html>.
  - **Write data into DataHub:** Click [DataHub agent for Logstash Output](#) to install the DataHub Output plugin.
  - **Download data from DataHub:** Click [DataHub agent for Logstash Input](#) to install the DataHub Input plugin.

#### Use case

**Case A: Collect log4j logs.**

This example shows how to collect the unstructured log4j log and derive a structure out of it by using Logstash. The format of a log4j log is as follows:

```
20:04:30.359 [qtp1453606810-20] INFO AuditInterceptor - [13pn9kdr5t
l84stzkmaa8vmg] end /web/v1/project/fhp4clxfbu0w3ym2n7ee6ynh/
statistics? executionName=bayes_poc_test GET, 187 ms
```

In this example, the user wants to derive a structure out of the log file and transfer the data into DataHub. The schema of the DataHub topic for the log4j file is as follows:

Table 11-12: DataHub topic schema

Column name	Data type
request_time	STRING
thread_id	STRING
log_level	STRING
class_name	STRING
request_id	STRING
detail	STRING

The configuration of the Logstash task is as follows:

```
input {
 file {
 path => "${APP_HOME}/log/bayes.log"
 start_position => "beginning"
 }
}

filter{
 grok {
 match => {
 "message" => "(? <request_time>\d\d:\d:\d\d:\d\d\.\d+)\s+\[(?
 <thread_id>[\w\~]+\)]\s+(? <log_level>\w+)\s+(? <class_name>\w+)\s+\-
 \s+\[(? <request_id>\w+)\]\s+(? <detail>.+)"
 }
 }
}

output {
 datahub {
 access_id => "Your accessId"
 access_key => "Your accessKey"
 endpoint => "Endpoint"
 project_name => "project"
 topic_name => "topic"
 #shard_id => "0"
 #shard_keys => ["thread_id"]
 dirty_data_continue => true
 dirty_data_file => "/Users/ph0ly/trash/dirty.data" }
}
```

```
 dirty_data_file_max_size => 1000
 }
}
```

### Case B: Collect CSV files.

This example shows how to use Logstash to collect CSV files. The format of the CSV file is as follows:

```
1111,1.23456789012E9,true,14321111111000000,string_dataxxx0,
2222,2.23456789012E9,false,14321111111000000,string_dataxxx1
```

The schema of the DataHub topic for the CSV file is as follows:

Table 11-13: DataHub topic schema

Column name	Data type
col1	BIGINT
col2	DOUBLE
col3	BOOLEAN
col4	TIMESTAMP
col5	STRING

The configuration of the Logstash task is as follows:

```
input {
 file {
 path => "${APP_HOME}/data.csv"
 start_position => "beginning"
 }
}

filter{
 csv {
 columns => ['col1', 'col2', 'col3', 'col4', 'col5']
 }
}

output {
 datahub {
 access_id => "Your accessId"
 access_key => "Your accessKey"
 endpoint => "Endpoint"
 project_name => "project"
 topic_name => "topic"
 #shard_id => "0"
 #shard_keys => ["thread_id"]
 dirty_data_continue => true
 dirty_data_file => "/Users/ph0ly/trash/dirty.data"
 dirty_data_file_max_size => 1000
 }
}
```





Name	Description
<b>project_name</b>	(Required) The name of the DataHub project.
<b>topic_name</b>	(Required) The name of the DataHub topic.
<b>retry_times</b>	(Optional) The maximum number of retries. -1: Unlimited retries. 0: No retries. >0: The specified number of retries. Default value: -1.
<b>retry_interval</b>	(Optional) The interval between retries. Unit: seconds. Default value: 5.
<b>shard_keys</b>	(Optional) The key of the shard. The hash of the key value is used to map a shard ID where the records are written. If the <b>shard_keys</b> and <b>shard_id</b> parameters are not specified, the system polls the shards to decide which shard to be written into.
<b>shard_id</b>	(Optional) The ID of the shard where records are written. If the <b>shard_keys</b> and <b>shard_id</b> parameters are not specified, the system polls the shards to decide which shard to be written into.
<b>dirty_data_continue</b>	(Optional) Indicates whether dirty records are ignored. true : Ignore the dirty records. Default value: false. If you set the value to true, you must specify the <b>dirty_data_file</b> parameter.
<b>dirty_data_file</b>	(Optional) The name of the dirty record file. The dirty record file is divided into .part 1 and .part 2. The most recent records are stored in part 2.
<b>dirty_data_file_max_size</b>	(Optional) The maximum size of the dirty record file. This value is for reference only.

The parameters of the DataHub Input plugin are described as follows:

Table 11-15: DataHub Input plugin parameter description

Name	Description
<b>access_id</b>	(Required) The AccessKey ID of Alibaba Cloud.
<b>access_key</b>	(Required) The AccessKey Secret of Alibaba Cloud.
<b>endpoint</b>	(Required) The endpoint of Alibaba Cloud DataHub.
<b>project_name</b>	(Required) The name of the DataHub project.
<b>topic_name</b>	(Required) The name of the DataHub topic.

Name	Description
retry_times	(Optional) The maximum number of retries. -1: Unlimited retries. 0: No retries. >0: The specified number of retries. Default value: -1.
retry_interval	(Optional) The interval between retries. Unit: seconds. Default value: 5.
shard_ids	(Optional) A list of shards to be consumed. If this remains blank, records in all the shards are consumed.
cursor	(Optional) The sequence number of the record from which the consumption begins. The consumption starts from the earliest record in the system by default.
pos_file	(Required) The checkpoint file, which is used to reset the starting point of consumption.

#### 11.5.4 Oracle GoldenGate

This topic describes how to install and use Oracle GoldenGate (OGG).

OGG is a tool for log-based structured data replication across heterogeneous environments. It is used for data backup between primary and secondary Oracle databases. It is also used to synchronize data from Oracle databases to other databases such as IBM Db2 and MySQL databases. OGG must be deployed in the source and destination databases. It is composed of the following components: Manager, Extract, data pump, Collector, and Replicat.

- Manager is the control process of OGG. A Manager process must be running on the source and destination databases. It is responsible for starting, stopping, and monitoring other processes.
- Extract is a process that captures data from the source database or transaction logs. You can configure the Extract process for initial data loads and incremental data synchronization. For initial data loads, Extract captures a set of data directly from their source objects. To keep source data synchronized to the destination database, Extract captures incremental DML and DDL operations after the initial data loading has taken place. This topic describes incremental data synchronization.
- A data pump is a secondary Extract group within the source OGG configuration. In a typical configuration with a data pump, the primary Extract group writes to

a trail on the source database. The data pump reads the trail and sends the DML or DDL operations over the network to a remote trail on the destination database.

- **Collector** is a process on the destination database, which receives data from the source database and generates trail files.
- **Replicat** is a process that reads the trail on the destination database, reconstructs the DML or DDL operations, and then applies them to the destination database.

The DataHub agent for OGG offers the Replicat feature that applies the updated data to DataHub by analyzing the trail. The data in DataHub is processed in real time by using Realtime Compute and can be archived into MaxCompute.

The following example shows how to synchronize incremental data from an Oracle database to DataHub and process the data in DataHub.

## Install OGG

### Prerequisites:

- You have installed the Oracle database client.
- You have obtained the OGG installation package for the source database. We recommend that you use OGG V12.1.2.1.
- You have obtained the OGG Adapters installation package for the destination database. We recommend that you use OGG Application Adapters 12.1.2.1.
- You have installed Java 7.

Follow these steps to install OGG:

#### 1. Install OGG for the source database.

- a. Extract the OGG installation package for the source database and the following directories appear:

```
drwxr-xr-x install
drwxrwxr-x response
-rwxr-xr-x runInstaller
drwxr-xr-x stage
```

- b. Install dependencies in response/oggcore.rsp. The OGG response file template is as follows:

```
oracle.install.responseFileVersion=/oracle/install/rspfmt_ogg
install_response_schema
#The installation option, which must reflect the installed Oracle
version. Specify ORA11g for installing OGG for Oracle Database 11g
.
INSTALL_OPTION=ORA11g
#The location in which OGG is installed.
```

```

SOFTWARE_LOCATION=/home/oracle/u01/ggate
#Indicates whether to start the Manager after installation.
START_MANAGER=false
#The port number of the Manager process.
MANAGER_PORT=7839
#The location of the Oracle database.
DATABASE_LOCATION=/home/oracle/u01/app/oracle/product/11.2.0/
dbhome_1
#The location that stores the inventory files. This parameter is
not required to be configured.
INVENTORY_LOCATION=
#The UNIX group of the inventory directory. In this example, OGG
is installed by using the ogg_test Oracle account. You can also
create a dedicated account for OGG as necessary.
UNIX_GROUP_NAME=oinstall

```

**c. Run the following command to install OGG:**

```
runInstaller -silent -responseFile {YOUR_OGG_INSTALL_FILE_PATH}/
response/oggcore.rsp
```



**Note:**

**In this example, OGG is installed in `/home/oracle/u01/ggate` and the installation logs are stored in `/home/oracle/u01/ggate/cfgtoollogs/oui`. The OGG installation is complete when the following message appears in the `silentInstall{time}.log` file:**

```
The installation of Oracle GoldenGate Core was successful.
```

**d. Run the following command and enter `CREATE SUBDIRS` as required to create OGG directories:**

```
/home/oracle/u01/ggate/ggsci
```

**2. Perform Oracle configurations in the source database.**

**Navigate to `sqlplus`: `sqlplus / as sysdba` as the database administrator and complete the following configurations:**

```

#Create a tablespace.
create tablespace ATMV datafile '/home/oracle/u01/app/oracle/oradata
/uprr/ATMV.dbf' size 100m autoextend on next 50m maxsize unlimited;

#Create a user named ogg_test. The password is also set to ogg_test
.
create user ogg_test identified by ogg_test default tablespace ATMV;

#Grant required privileges to ogg_test.
grant connect,resource,dba to ogg_test;

#Check whether supplemental logging is enabled for the database.
Select SUPPLEMENTAL_LOG_DATA_MIN, SUPPLEMENTAL_LOG_DATA_PK,
SUPPLEMENTAL_LOG_DATA_UI, SUPPLEMENTAL_LOG_DATA_FK, SUPPLEMENT
AL_LOG_DATA_ALL from v$database;

```

```
#If the result is NO, enable supplemental logging.
alter database add supplemental log data;
alter database add supplemental log data (primary key, unique,
foreign key) columns;
#Enable rollback.
alter database drop supplemental log data (primary key, unique,
foreign key) columns;
alter database drop supplemental log data;

#Enable all column logging at the database level. Note: Even when
all column logging is enabled, only primary key columns are logged
for a delete operation.
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
#Enable the forced logging mode.
alter database force logging;
#Run the marker_setup.sql script.
@marker_setup.sql
#Run the ddl_setup.sql script.
@ddl_setup.sql
#Run the role_setup.sql script.
@role_setup.sql
#Grant the GGS_GGSUSER_ROLE to ogg_test.
grant GGS_GGSUSER_ROLE to ogg_test;
#Run the ddl_enable.sql script to enable the DDL trigger.
@ddl_enable.sql
#Run the ddl_pin script to improve the performance of the DDL
trigger.
@ddl_pin ogg_test
#Run the sequence.sql script.
@sequence.sql
#
alter table sys.seq$ add supplemental log data (primary key) columns
;
```

### 3. Configure the Manager process on the source database.

**Start the Oracle GoldenGate Software Command Interface (GGSCI) and perform the following steps:**

#### **a. Run the following command to configure the Manager process:**

```
edit params mgr
PORT 7839
DYNAMICPORTLIST 7840-7849
USERID ogg_test, PASSWORD ogg_test
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPDAYS 7
LAGREPORTHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
PURGEDDLHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 7
```

```
PURGEMARKERHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 7
```

- b. Run the following command to start the Manager process. The logs are stored in ggate/dirrpt.**

```
start mgr
```

- c. Run the following command to check whether the Manager process is running:**

```
info mgr
```

- d. Run the following command to view the Manager parameter file:**

```
view params mgr
```

#### 4. Configure the Extract process on the source database.

**Start the GGSCI and perform the following steps:**

- a. Run the following command to configure the Extract process. In the following example, the group name of the process is extract.**

```
edit params extractEXTRACT extract
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
DBOPTIONS ALLOWUNUSEDCOLUMN
USERID ogg_test, PASSWORD ogg_test
REPORTCOUNT EVERY 1 MINUTES, RATE
NUMFILES 5000
DISCARDFILE ./dirrpt/ext_test.dsc, APPEND, MEGABYTES 100
DISCARDROLLOVER AT 2:00
WARNLONGTRANS 2h, CHECKINTERVAL 3m
EXTTRAIL ./dirdat/st, MEGABYTES 200
DYNAMICRESOLUTION
TRANLOGOPTIONS CONVERTUCS2CLOBS
TRANLOGOPTIONS RAWDEVICEOFFSET 0
DDL &
INCLUDE MAPPED OBJTYPE 'table' &
INCLUDE MAPPED OBJTYPE 'index' &
INCLUDE MAPPED OBJTYPE 'SEQUENCE' &
EXCLUDE OPTYPE COMMENT
DDOPTIONS NOCROSSRENAME REPORT
TABLE OGG_TEST. *;
SEQUENCE OGG_TEST. *;
```

```
GETUPDATEBEFORES
```

- b. Run the following command to add an Extract process. Replace `extract` in the following command with your actual group name.

```
add ext extract,tranlog, begin now
```

- c. Run the following command to delete an Extract process. In the following example, the process name is `DP_TEST`.

```
delete ext DP_TEST
```

- d. Run the following command to create a trail, associate the trail with the Extract group named `extract`, and set the maximum file size in the trail to 200 megabytes:

```
add exttrail ./dirdat/st,ext extract, megabytes 200
```

- e. Run the following command to start the Extract process. The logs are stored in `ggate/dirrpt`.

```
start extract extract
```



**Note:**

After the Extract process configuration is complete, you can view the changes to the database in the files stored in the `ggate/dirdat` directory.

**5. Create a DEFGEN parameter file.**

- a. Start the GGSCI in the source database. In GGSCI, run the following command to create a DEFGEN parameter file and copy the file to the `dirdef` directory in the destination database:

```
edit params defgen
DEFSFILE ./dirdef/ogg_test.def
USERID ogg_test, PASSWORD ogg_test
```

```
table OGG_TEST. *;
```

- b. Run the following command from the shell to create a DEFGEN parameter file named `ogg_test.def`:**

```
./defgen paramfile ./dirprm/defgen.prm
```

**6. Install and configure OGG in the destination database.**

- a. Extract the OGG installation package to the destination database.**
- b. Copy the `dirdef/ogg_test.def` file in the source database to `dirdef` of the destination database.**
- c. Start the GGSCI and run the following command to create the default directories of OGG:**

```
create subdirs
```

- d. Run the following command to configure the Manager process:**

```
edit params mgr
PORT 7839
DYNAMICPORTLIST 7840-7849
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPDAYS 7
LAGREPORTHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
PURGEDDLHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 7
PURGEMARKERHISTORY MINKEEPDAYS 3, MAXKEEPDAYS 7
```

- e. Run the following command to start the Manager process:**

```
start mgr
```

**7. Configure a data pump in the source database.**

**Start the GGSCI and perform the following steps:**

- a. Run the following command to configure a data pump:**

```
edit params pump
EXTRACT pump
RMTHOST xx.xx.xx.xx, MGRPORT 7839, COMPRESS
PASSTHRU
NUMFILES 5000
RMTTRAIL ./dirdat/st
DYNAMICRESOLUTION
TABLE OGG_TEST. *;
```



```
SEQUENCE OGG_TEST. *;
```

- b. Run the following command to create a data-pump Extract process. The process reads from the specified trail.**

```
add ext pump,exttrailsource ./dirdat/st
```

- c. Run the following command to create a trail and set the maximum file size in the trail to 200 megabytes:**

```
add rmttrail ./dirdat/st,ext pump,megabytes 200
```

- d. Run the following command to start the data pump:**

```
start pump
```



**Note:**

After the data pump is started, you can view the trail files in the dirdat directory of the destination database.

**8. Install and configure the DataHub agent for OGG.**

- a. Run the following command to configure the `JAVA_HOME` and `LD_LIBRARY_PATH` environment variables and specify the configurations in the `~/.bash_profile`:**

```
export JAVA_HOME=/xxx/xxx/jrexx
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${JAVA_HOME/lib/amd64:$
JAVA_HOME/lib/amd64/server
```

- b. After the environment variables are configured, restart the Manager process in the destination database.**
- c. Download the [DataHub agent for OGG](#) and extract the installation package.**
- d. Modify the `javaue.properties` and `log4j.properties` files in the `conf` sub-directory of the installation directory and replace `{YOUR_HOME}` with the target path of the extracted files:**

```
gg.handlerlist=ggdatahub
gg.handler.ggdatahub.type=com.aliyun.odps.ogg.handler.datahub.
DatahubHandler
gg.handler.ggdatahub.configureFileName={YOUR_HOME}/datahub-ogg-
plugin/conf/configure.xml
goldengate.userexit.nockpt=false
goldengate.userexit.timestamp=utc
gg.classpath={YOUR_HOME}/datahub-ogg-plugin/lib/*
gg.log.level=debug
```

```
jvm.bootoptions=-Xmx512m -Dlog4j.configuration=file:{YOUR_HOME}/
datahub-ogg-plugin/conf/log4j.properties -Djava.class.path=ggjava/
ggjava.jar
```

**e. Modify the configure.xml file in the conf sub-directory of the installation directory as follows:**

```
<? xml version="1.0" encoding="UTF-8"? >
<configure>

 <defaultOracleConfigure>
 <!--(Required) The Oracle database system identifier (SID)
 .-->
 <sid>100</sid>
 <!--The schema of the Oracle table, which can be
 overwritten by oracleSchema in the column mappings. At least one
 of them must be specified.-->
 <schema>ogg_test</schema>
 </defaultOracleConfigure>

 <defalutDatahubConfigure>
 <!--(Required) The endpoint of DataHub.-->
 <endPoint>YOUR_DATAHUB_ENDPOINT</endPoint>
 <!--The DataHub project, which can be overwritten by
 datahubProject in the column mappings. At least one of them must
 be specified.-->
 <project>YOUR_DATAHUB_PROJECT</project>
 <!--The AccessKey ID for accessing DataHub, which can be
 overwritten by datahubAccessId in the column mappings. At least
 one of them must be specified.-->
 <accessId>YOUR_DATAHUB_ACCESS_ID</accessId>
 <!--The AccessKey Secret for accessing DataHub, which
 can be overwritten by datahubAccessKey in the column mappings. At
 least one of them must be specified.-->
 <accessKey>YOUR_DATAHUB_ACCESS_KEY</accessKey>
 <!--The column in DataHub that indicates the data update
 type, which can be overwritten by ctypeColumn in the column
 mappings.-->
 <ctypeColumn>optype</ctypeColumn>
 <!-- The column in DataHub that indicates the data update
 time, which can be overwritten by ctimeColumn in the column
 mappings.-->
 <ctimeColumn>readtime</ctimeColumn>
 <!-- The column in DataHub that indicates the sequence
 number of the updated data, which can be overwritten by cidColumn
 in the column mappings. The sequence number increases as more data
 are updated, but may not be consecutive.-->
 <cidColumn>record_id</cidColumn>
 </defalutDatahubConfigure>

 <!--The approach to handling errors. If an error occurs, the
 system either ignores the error and continues running or retries
 the operation repeatedly.-->

 <!--(Optional) The maximum number of records operated at one
 time. Default value: 1000.-->
 <batchSize>1000</batchSize>

 <!--(Optional) The format that the timestamp is converted
 into. Default: yyyy-MM-dd HH:mm:ss.-->
 <defaultDateFormat>yyyy-MM-dd HH:mm:ss</defaultDateFormat>
```

```

 <!--(Optional) Indicates whether the system needs to ignore
 dirty records. Default value: false.-->
 <dirtyDataContinue>true</dirtyDataContinue>

 <!--(Optional) The dirty record file name. Default value:
 datahub_ogg_plugin.dirty-->
 <dirtyDataFile>datahub_ogg_plugin.dirty</dirtyDataFile>

 <!--(Optional) The maximum size of the dirty record file.
 Unit: MB. Default value: 500.-->
 <dirtyDataFileMaxSize>200</dirtyDataFileMaxSize>

 <!--(Optional) The maximum number of retries if an error
 occurs. -1: Unlimited. 0: No retries. n: The number of retries.
 Default value: -1.-->
 <retryTimes>0</retryTimes>

 <!--(Optional) The interval between retries. Unit: millisecon
 ds. Default value: 3000.-->
 <retryInterval>4000</retryInterval>

 <!--(Optional) The checkpoint file name. Default value:
 datahub_ogg_plugin.chk.-->
 <checkPointFileName>datahub_ogg_plugin.chk</checkPointFileName
 >

 <mappings>
 <mapping>
 <!--The schema of the Oracle table.-->
 <oracleSchema></oracleSchema>
 <!--(Required) The Oracle table name.-->
 <oracleTable>t_person</oracleTable>
 <!--The DataHub project name.-->
 <datahubProject></datahubProject>
 <!--The AccessKey ID for accessing DataHub.-->
 <datahubAccessId></datahubAccessId>
 <!--The AccessKey Secret for accessing DataHub.-->
 <datahubAccessKey></datahubAccessKey>
 <!--(Required) The DataHub topic name.-->
 <datahubTopic>t_person</datahubTopic>
 <ctypeColumn></ctypeColumn>
 <ctimeColumn></ctimeColumn>
 <cidColumn></cidColumn>
 <columnMapping>
 <!--
 src: (Required) The column names in the Oracle
table.
 dest: (Required) The column names in the DataHub
topic.
 destOld: (Optional) The DataHub topic column that
records the data before it is updated.
 isShardColumn: (Optional) Indicates whether the
shard ID is generated based on the hash key value, which can be
overwritten by shardId. Default value: false.
 isDateFormat: Indicates whether the timestamp
is converted into a string based on dateFormat. Default value:
true. If you set the value to false, the data type in the source
database must be long.
 dateFormat: The format that the timestamp is
converted into. If this parameter is left blank, the default
format is used.
 -->
 <column src="id" dest="id" isShardColumn="true"
isDateFormat="false" dateFormat="yyyy-MM-dd HH:mm:ss"/>

```

```

"/>
 <column src="name" dest="name" isShardColumn="true
 <column src="age" dest="age"/>
 <column src="address" dest="address"/>
 <column src="comments" dest="comments"/>
 <column src="sex" dest="sex"/>
 <column src="temp" dest="temp" destOld="temp1"/>
 </columnMapping>

 <!--(Optional) The ID of the shard prioritized to be
written into.-->
 <shardId>1</shardId>
 </mapping>
</mappings>
</configure>

```

**f. Run the following command in GGSCI to start the DataHub writer:**

```

edit params dhwriter
extract dhwriter
getenv (JAVA_HOME)
getenv (LD_LIBRARY_PATH)
getenv (PATH)
CUSEREXIT ./libggjava_ue.so CUSEREXIT PASSTHRU INCLUDEUPD
ATEBEFORES, PARAMS "{YOUR_HOME}/datahub-ogg-plugin/conf/javaue.
properties"
sourcedefs ./dirdef/ogg_test.def
table OGG_TEST. *;

```

**g. Run the following command to add a DataHub writer:**

```
add extract dhwriter, exttrailsource ./dirdat/st
```

**h. Run the following command to start the writer:**

```
start dhwriter
```

#### Use case

For example, you have an Oracle table that stores order information. The table has three columns. The column names are `oid`, `pid`, and `num`, which indicate order ID, product ID, and product quantity. You can synchronize incremental data to DataHub by using the DataHub agent for OGG. The steps are as follows:



#### Note:

Before performing incremental data synchronization, you must synchronize existing data from the source table to MaxCompute by using DataX.

**1. Create a topic in DataHub. The schema of the topic is as follows:**

```
string record_id, string optype, string readtime, bigint oid_before
, bigint oid_after, bigint pid_before, bigint pid_after, bigint
num_before, bigint num_after
```

**2. Make sure that you have completed the deployment of the DataHub agent for OGG. Then configure the column mappings as follows:**

```
<ctypeColumn>optype</ctypeColumn>
 <ctimeColumn>readtime</ctimeColumn>
 <cidColumn>record_id</cidColumn>
 <columnMapping>
 <column src="oid" dest="oid_after" destOld="oid_before"
isShardColumn="true"/>
 <column src="pid" dest="pid_after" destOld="pid_before"/>
 <column src="num" dest="num_after" destOld="num_before"/>
 </columnMapping>
```



**Note:**

The **optype** parameter indicates the type of the data update. Valid values of the **optype** parameter are I, D, and U, which represent an insert, delete, and update operation, respectively. The **readtime** parameter indicates the time of the data update.

**3. When the agent can run properly, data updates are synchronized from the source table to DataHub.**

## 11.6 Data Archive

### 11.6.1 Overview

In the DataHub console, you can archive data in DataHub to other data warehouses by using DataConnector so that you can easily analyze and process historical data.

This section describes how to archive data in DataHub to MaxCompute.

### 11.6.2 Archive to MaxCompute

#### 11.6.2.1 Create a DataConnector

This topic describes how to archive data from DataHub to MaxCompute.

1. Log on to the DataHub console.
2. On the Projects page, click View in the Actions column of a project. On the project details page that appears, click View in the Actions column of a topic.

3. On the topic details page, click +DataConnector in the upper-right corner and select MaxCompute. Then the Create DataConnector dialog box appears.
4. In the Create DataConnector dialog box, configure all the required information and click Create.

**Note:**

The parameters required to be configured are described as follows.

Table 11-16: Parameter description

Name	Description
Project	The name of the MaxCompute project to which data in the topic is archived.
Table	The name of the MaxCompute table to which data in the topic is archived.
AccessKey ID and AccessKey Secret	The AccessKey for accessing MaxCompute. The AccessKey must belong to a RAM user that has CreateInstance, Desc , and Alter permissions on the MaxCompute table.
Partition Mode	The following modes are available: system_time, event_time, and user_define. If you select system_time, partitions are created based on the recording time. If you select event_time, partitions are created based on the specified event_time value. When you create the topic, you must define a topic field as event_time and set its data type to TIMESTAMP. The field values must be accurate to microseconds. If you select user_define, partitions are created based on the user-defined partition field.
Partitioning Interval	If you set Partition Mode to system_time or event_time, you must set a partitioning interval. The minimum value is 15 minutes.
Partition Key	Only the default format is supported. Customized format will be supported at a later date. If the partitioning interval is set to 15 minutes, the partition key format is <i>ds=20170704,hh=01,mm=15</i> . Note that a corresponding partition field must be contained in the MaxCompute table.

### 11.6.2.2 View archive details

This section describes how to view archive details after a DataConnector has been created.

The Details page of DataConnector displays the basic information including the name of the topic involved, the destination MaxCompute project and table names, and the archive progress. The archive progress is indicated by Latest Record Time, Last Synced Record Time, and Latency.



**Note:**

Latest Record Time indicates the timestamp at which the latest record was ingested to DataHub. Last Synced Record Time indicates the timestamp at which the latest record that has been synchronized to the destination platform was ingested to DataHub. Latency, in seconds, indicates the gap between the latest record time and the last synced record time.

You can also check the archive progress of each shard. If a shard is in ERROR status, place your mouse pointer over the question mark next to ERROR to show the cause of the issue. After the issue is fixed, you can click Resume in the Actions column of the shard to resume the DataConnector. For example, if the task is terminated because the corresponding MaxCompute partition is deleted, you can re-create the MaxCompute partition by using SDK or in the MaxCompute console. Then click Resume in the Actions column of the shard to resume archiving. If multiple shards are not in RUNNING status, you can click Resume Archiving to resume archiving for all these shards.

## 11.7 Performance monitoring

This topic describes how to monitor the performance of DataHub.

In the DataHub console, you can view near real-time performance statistics of topics, such as queries per second (QPS) and throughput. The available metrics are listed as follows:

- Read and write QPS
- Read and write records per second (RPS)
- Read and write throughput, measured in KB per second
- Read and write latency, measured in microseconds per request

On the Projects page, click View in the Actions column of a project. On the project details page that appears, click View in the Actions column of a topic. Then, on the topic details page, click Metric to view the performance statistics of the topic.

Shards	DataConnector	Metric	Schema	Subscription
		2019-02-24 10:16	2019-02-28 11:16	
Datahub Read QPS(request/second)		Datahub Read RPS(record/second)		
Datahub Read Throughput(KB/second)		Datahub Read Latency(us/request)		

You can obtain the performance statistics for a specified time range.