ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

Data Management

User Guide

Product Version: v3.16.2

Document Version: 20220916

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	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.
Warning	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.
Notice	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.
Note	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 `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.What is DMS?

Data Management (DMS) is a one-stop data management platform that allows you to manage data throughout the data lifecycle. You can use DMS to seamlessly access data sources such as online transaction processing (OLTP) databases, online analytical processing (OLAP) databases, and NoSQL databases, and manage more than 10 types of data sources in a unified manner. You can use DMS to manage global data assets, design and develop databases, and integrate, develop, consume, and govern data. These features help enterprises mine value from data in an efficient and secure manner and facilitate the digital transformation of enterprises.

2.Quick start

2.1. Log on to the DMS console

This topic uses Google Chrome as an example to describe how to log on to the DMS console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
   - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator:
     a. On the Bind Virtual MFA Device page, bind an MFA device.
     b. Enter the username and password again as in Step 2 and click **Log On**.
     c. Enter a six-digit MFA verification code and click **Authenticate**.
   - You have enabled MFA and bound an MFA device:

     Enter a six-digit MFA verification code and click **Authenticate**.

     > **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Database Services > Data Management**.
6. Set the **Organization** and **Region** parameters and click **Access as Administrator**.

> **Note** If you log on to the DMS console as a DMS administrator and your account is added to multiple tenants, you can move the pointer over the  icon in the upper-right corner and select **Switch tenant** to switch to another tenant.

2.2. Register database instances with DMS

To manage database instances in DMS, you must register the database instances with DMS. DMS allows you to register ApsaraDB instances and self-managed databases that are hosted over the Internet. This topic describes how to register an ApsaraDB RDS for MySQL instance with DMS.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**. In the left-side navigation pane, click **Instances**.
3. On the Instance List tab, click **New**.
4. In the **Add Instance** dialog box, click the **Cloud** tab.
5. On the Cloud tab, select the type of database instance that you want to register.
6. In the Add Instance dialog box, set the parameters as required.

   This example shows you how to register an ApsaraDB RDS for MySQL instance with DMS. The following table describes the parameters.

Section	Parameter	Description
Basic Information	**Data Source**	The source of the database instance. In this example, **Cloud** is selected.
	Database Type	The type of the database instance. In this example, **MySQL** is selected.
	Instance Region	The region in which the database instance resides. Select a region from the drop-down list.
	Entry mode	The method that you use to log on to the database instance. Default value: **Connection string address**. This value cannot be changed.
	Connection string address	The endpoint of the database instance. The endpoint contains a port number.
	Database Account	The username of the database account used to log on to the database instance.
	Database password	The password of the database account.

Section	Parameter	Description
	Control Mode	The control mode that is used to manage the database instance. For more information, see Control modes. **Note** If you set this parameter to **Security Collaboration**, you must set the **Security Rules** parameter.
Advanced Information	**Environment type**	The type of the environment in which the database instance is deployed.
	Instance Name	The name of the database instance.
	Enable DSQL	Specifies whether to enable the cross-database query feature. To enable the cross-database query feature, you must specify a database link name. For more information, see Cross-database query.
	Lock-free Schema Change	Specifies whether to enable the lock-free schema change feature. **Note** This parameter is required only for MySQL databases.
	Enable SSL	Specifies whether to allow DMS to connect to the database instance by using SSL connections. If this feature is enabled, DMS can connect to the database instance by using SSL connections. SSL encrypts network connections at the transport layer to improve the security and integrity of data in transit. However, SSL increases the response time of network connections. Before you use SSL connections, make sure that the SSL encryption feature is enabled for the database instance. Valid values: ◦ **Default (DMS automatically checks whether self-negotiation is enabled for the database instance.)**: DMS automatically checks whether the SSL encryption feature is enabled for the database instance. If the SSL encryption feature is enabled, DMS connects to the database instance by using SSL connections. Otherwise, DMS connects to the database instance without encryption. ◦ **Open**: DMS connects to the database instance by using SSL connections. This value is invalid if you disable the SSL encryption feature for the database instance. ◦ **Close**: DMS does not connect to the database instance by using SSL connections. **Note** This parameter is required only for MySQL databases.

Section	Parameter	Description
	DBA	The DBA of the database instance. The DBA can grant permissions to users.
	query timeout(s)	The timeout period for the execution of an SQL query statement. If the execution of an SQL query statement lasts longer than the specified timeout period, the execution of the statement is terminated to protect the database.
	export timeout(s)	The timeout period for the execution of an SQL export statement. If the execution of an SQL export statement lasts longer than the specified timeout period, the execution of the statement is terminated to protect the database.

7. After you set the preceding parameters, click **Basic Information**, and then click **Test connection** in the lower-left corner of the dialog box to verify the settings.

   > **Note** If the connectivity test fails, an error message appears. Modify the parameters as prompted.

8. Click **Submit**.

Result

After you register an ApsaraDB RDS for MySQL instance with DMS, the instance appears in the instance list of the DMS console. You can view and manage the instance in the DMS console.

2.3. Add a user

Data Management (DMS) allows you to manage users. You can add users and assign the required roles to each user based on your business requirements.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **System > User**.

   > **Note** On the User tab, you can perform the required operations on the existing users. For example, you can edit, disable, enable, or delete a user.

3. Click **New**.
4. In the Add User dialog box, set the required parameters. The following table describes the parameters.

Parameter	Description

Parameter	Description
Alibaba Cloud Account	The ID of an Apsara Stack tenant account. You can enter one of the following IDs: ○ The ID of an Apsara Stack tenant account. You can obtain the ID from the account owner. ○ The ID of a Resource Access Management (RAM) user. You can obtain the required ID from the Service-linked Roles page of the Apsara Uni-manager Management Console.
Role	The role that you want to assign to the user based on your business requirements. Valid values: ○ Regular User ○ DBA ○ Administrator ○ Security Administrator ○ Technical Support **Note** For more information about the features that are supported by each role, see Features that are supported by each role.

5. Click **OK**.

2.4. Control modes

DMS provides three control modes for you to manage instances: Flexible Management, Stable Change, and Security Collaboration. You can specify a control mode for each instance.

Control mode	Description	Scenario	Logon method
Flexible management	This control mode allows you to manage the visualized data and schemas of multiple types of databases. It also provides a variety of data management solutions. This simplifies the use of databases and facilitates management.	• Database instances do not require strict control. • Database instances are used by a single user.	A database account and the related password.
Stable change	• This control mode provides multiple solutions to ensure database reliability. These solutions allow you to change data without the need to lock the related table or schema. • All features that are included in the flexible management control mode are available.	• Database instances require a high level of availability. This ensures that these database instances function as expected for an extended period of time. • Database instances are used by a small-sized group that includes multiple users.	A database account and the related password.

Control mode	Description	Scenario	Logon method
Security collaboration	• This control mode provides multiple solutions to ensure data security. These solutions include fine-grained access control at the database, table, or field level and sensitive data management. • This control mode allows you to produce enterprise-specific database DevOps solutions through custom design specifications and approval processes. • All features that are included in the flexible management and stable change control modes are available.	• Ensure the data security of database instances. • Implement strict access control over development or change workflows. • Manage compliance for enterprises.	Logon-free through authorization.

Note The instances that are managed in Stable Change mode consume the billing quota of the instances that are managed in Security Collaboration mode.

2.5. Features that are supported by each role

DMS provides the following roles: regular user, DBA, security administrator, and DMS administrator. This topic describes the features that are supported by each role.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Permission	Permission management	√	√	√	√	You can use this feature to apply for permissions on instances, databases, tables, and sensitive fields. You can also view permissions that you have.
	Data Changes	√	√	√	√	You can use this feature to initialize data for a newly published project, clean up historical data, fix bugs, or run a test.
	Data Import	√	√	√	√	You can use this feature to import a large amount of data to your databases at a time.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Data Plans	Data Export	√	√	√	√	You can use this feature to export a large amount of data for analysis or export the required data.
	Data Tracking	√	√	√	√	If specific data fails to meet your requirements due to reasons such as misoperation, you can use this feature to restore data to the normal state.
	Test Data Generate	√	√	√	√	Some business scenarios may require frequent data preparation. In this case, you can use this feature to generate test data to ensure data security and discreteness and improve production efficiency.
	Data Warehouse Development	√	√	√	√	DMS uses a database as a computing engine and integrates various tools and services, such as Data Transmission Service (DTS) and Data Lake Analytics (DLA), in the database ecosystem for data warehouse development. You can use this feature to develop and manage data warehouses in DMS with ease.
	Data Service	√	√	√	√	You can use this feature to export data at the field or row level, display data in a visualized manner, and publish API operations to the Alibaba Cloud Marketplace for sale.
	Database Clone	√	√	√	√	You can use this feature to clone MySQL databases.
	Schema Design	√	√	√	√	When you develop or optimize projects or process new business requirements, you can use this feature to change schemas. For example, you can use this feature to create a table or modify an existing table.

Schemas Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
	Table Sync	√	√	√	√	You can use this feature to compare and synchronize the schemas of tables in different environments, such as online and offline environments. This feature helps ensure the consistency of schemas.
Optimization	SQL Review	√	√	√	√	You can use this feature to prevent SQL statements that do not use indexes or do not conform to database development standards. This feature helps protect against SQL injection attacks.
SQLConsole	Single Database query	√	√	√	√	You can write SQL statements to query data in a single database. This feature can be used to verify business code, analyze product effects, and identify issues in an online environment.
	Cross-database Query	√	√	√	√	You can use this feature to perform join queries across online heterogeneous databases that are deployed in different environments.
System Management	Instance management	×	√	×	√	You can use this feature to manage instances. For example, you can register, view, or edit instances.
	User management	×	×	×	√	You can use this feature to manage users. For example, you can add, view, or edit users as needed.
	Task management	×	√	×	√	You can use this feature to manage tasks. For example, you can create, start, or stop tasks.
	Configuration management	×	×	×	√	You can use this feature to view and modify system configurations, or view the historical modifications of the configurations.

Category	Feature	Regular user	DBA	Security administrator	DMS administrator	Description
Security management	Security Rules	×	√	×	√	You can use this feature to configure security rules. Only SQL statements that conform to the security rules can be executed.
	Approval Processes	×	√	×	√	Approval processes are associated with security rules. You can configure different approval processes for different types of tickets.
	Operation Logs	×	√	√	√	Operations logs record data changes. Each record contains information such as the user who performed the operation, operation details, and time at which the operation was performed. You can use this feature to track historical user operations at any time.
	Access IP Whitelists	×	×	×	√	After you configure an access IP whitelist, only the IP addresses or Classless Inter-Domain Routing (CIDR) blocks in the whitelist can access the resources within your DMS tenant. This effectively enhances data security.
	Sensitive Data	×	√	√	√	You can use this feature to manage sensitive data. For example, you can use algorithms to de-identify sensitive data or adjust the security levels of sensitive data.
Tickets	Ticket management	√	√	√	√	You can use this feature to configure notification methods. DMS can notify you of the approval or execution status of tickets by using DingTalk notifications or emails.

3.Data Assets

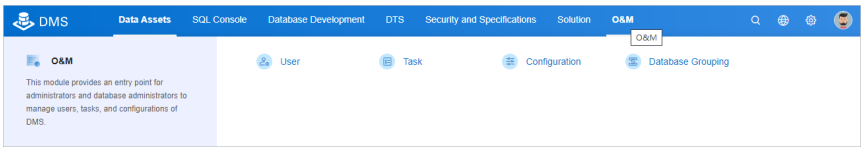
3.1. Home page

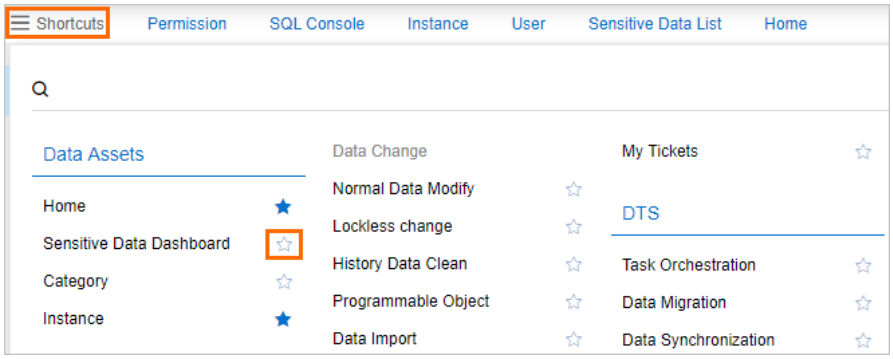
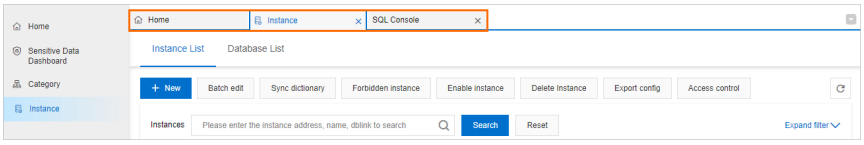
3.1.1. Customize the layout and style of the DMS console

To meet the requirements of different users, DMS allows you to customize the layout and style of the DMS console based on your business requirements. This topic describes how to customize the relevant configurations.

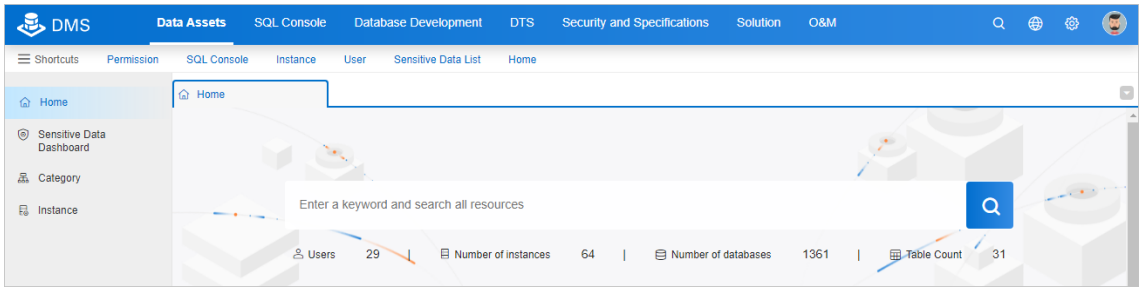
Procedure

1. Log on to the DMS console.
2. In the right section of the top navigation bar, click the icon to view the items that you can configure.
3. Configure the items based on your business requirements.

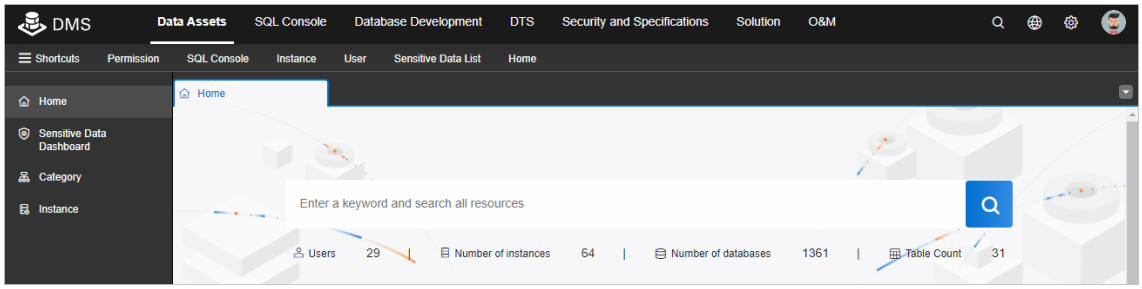
Section	Description
Drop-down Menus in Top Navigation Bar	○ By default, the **Drop-down Menus in Top Navigation Bar** switch is turned off. You can turn on the switch. ○ If you turn on Drop-down Menus in Top Navigation Bar and move the pointer over a tab in the top navigation bar, a drop-down list appears. You can access each feature by clicking the name of the feature in the drop-down list with ease.  **Note** If you turn off Drop-down Menus in Top Navigation Bar, no drop-down lists appear when you move the pointer over to a tab in the top navigation bar. You can access each feature only by clicking the name of the feature in the left-side navigation pane.

Section	Description
Shortcuts	○ By default, the Shortcuts switch is turned on. ○ If you turn on Shortcuts, you can click Shortcuts below the top navigation bar to add the features that you regularly use to the section on the right side of Shortcuts. You can add features to or remove features from the section on the right side of Shortcuts by performing the following operations: a. In the upper-left corner of the DMS console, move the pointer over **Shortcuts**. b. ■ In the shortcut menu that appears, click the ☆ icon to the right of a feature to add the feature to the section on the right side of Shortcuts. ■ Click the ★ icon to the right of a feature to remove the feature from the section on the right side of Shortcuts. 
Show Tabs	○ By default, the Show Tabs switch is turned on. ○ If you turn on Show Tabs, you can view and use multiple features. Each feature is displayed on a tab. You can drag the tabs to adjust the order in which the tabs are displayed.  **Note** If you turn off Show Tabs, you can view and use only one feature at a time. The feature that you view and use is displayed on a tab.

4. Select a blue skin or a dark skin for the DMS console.
   ○ Blue, which is the default skin



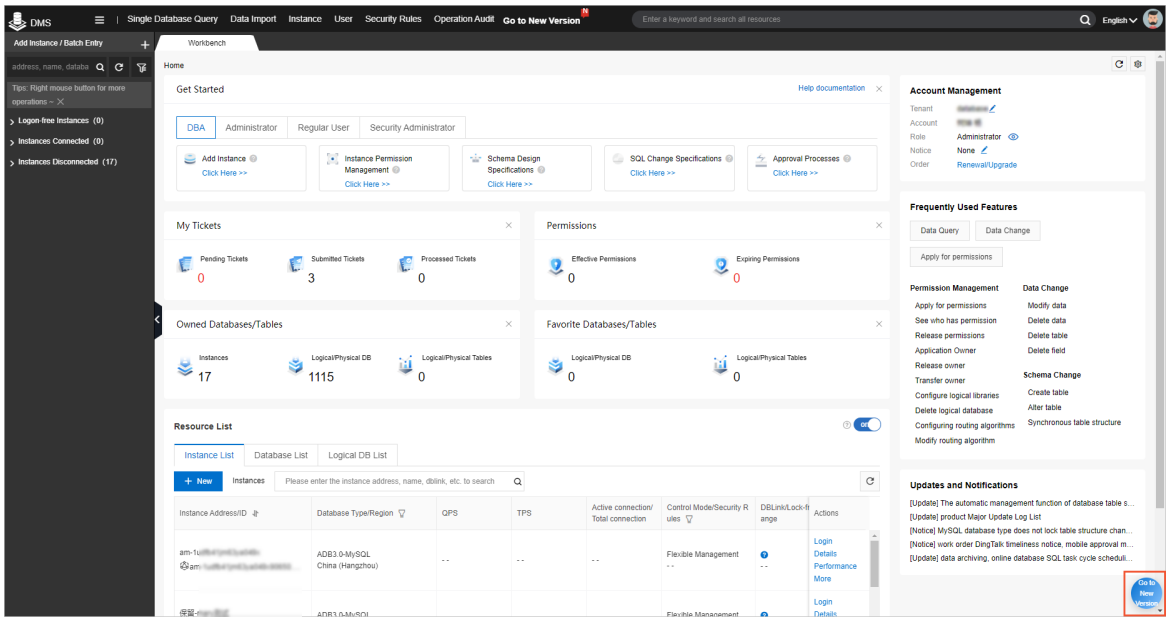
- Dark



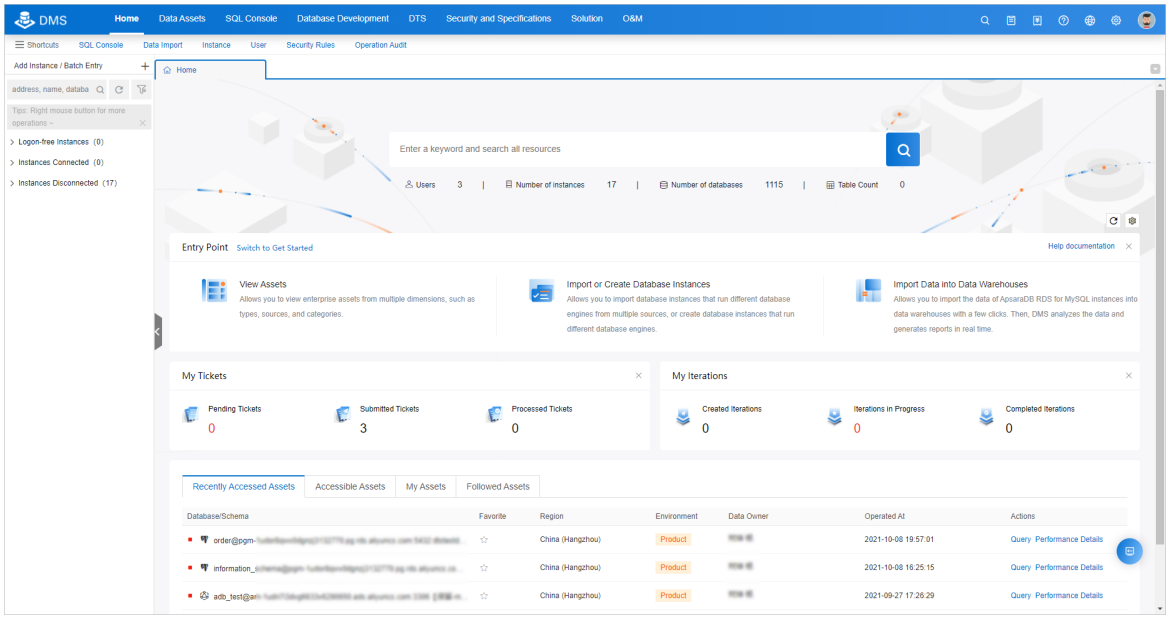
3.1.2. Experience the new version of the DMS console

The new version of the DMS console classifies functional modules based on scenarios, and integrates the features of Data Transmission Service (DTS). This topic describes how to go to the new version of the DMS console and return to the previous version of the DMS console.

1. Log on to the DMS console.
2. On the **Workbench** tab, click the **Go to New Version** icon in the lower-right corner.



3. In the message that appears, click **Leave** to go to the new version of the DMS console, as shown in the following figure.



Note For more information about how to return to the previous version of the DMS console, see Return to the previous version of the DMS console.

Return to the previous version of the DMS console

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**.
3. In the left-side navigation pane, click **Home**.
4. Click the icon in the lower-right corner of the page to return to the previous version of the DMS console.

3.2. Use the category feature

As the business develops and the number of tables increases, Data Management (DMS) provides the category feature to help you classify tables. This way, administrators, developers, and O&M engineers can manage or use the tables more conveniently.

Prerequisites

- A relational database or data warehouse is used. For more information, see Supported databases.
- You are a DMS administrator or DBA. For information about user roles, see Features that are supported by each role.

Manage categories

1. Log on to the DMS console.
2. In the top navigation bar, move the pointer over the **More** icon and choose **Data Factory > Category**.
3. On the tab that appears, click **Create Category**.

4. In the **Category Name** field of the dialog box that appears, enter the name of a category and click **OK**.
   The created category is displayed in the left-side category tree.

   > **Note** By default, DMS creates the **Uncategorized** category. All the tables that are not added to categories belong to this category.

5. On the right side of a category, move the pointer over the ··· icon and perform an operation to manage the category. You can perform the following operations:
   - **Modify the name of a category**

     To modify the name of a category, move the pointer over the ··· icon and select **Change**.
   - **Create a subcategory**

     To create a subcategory, move the pointer over the ··· icon and select **Create Subcategory**.

     You can create up to four category levels. If a table is added to a category, you cannot create a subcategory for this category.
   - **Delete a category**

     To delete a category, move the pointer over the ··· icon and select **Delete**.

     > **Note** If a subcategory is created in this category or tables are added to this category, this category cannot be deleted.

Add a table to a category

Each table can be added to only one category. If you add a table to another category, the table is removed from the existing category.

1. Log on to the DMS console.
2. In the top navigation bar, move the pointer over the **More** icon and choose **Data Factory > Category**.
3. Click the category to which you want to add a table.

   > **Note** To view subcategories, you can click the > icon on the left side of the category.

4. On the page of the category, click **Quick Add** in the upper-right corner of the page.

   > **Note** You can also use the following method to add a table to a category.
   > - On the page of the **Uncategorized** category, find the table that you want to manage and click **Associate Category** in the Actions column.

5. In the dialog box that appears, search for and select the table that you want to add, and then click **OK**.

> **Note** You can add multiple tables to a category at a time.

Remove a table from a category

To remove a table from a category, find the table on the page of the category and click **Remove from Category** in the Actions column.

3.3. Manage instances

Data Management (DMS) allows you to manage database instances. For example, you can export the information about instance configurations.

Prerequisites

You are a database administrator (DBA) or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, move the pointer over the **More** icon and choose **System > Instance**.
3. On the **Instance List** tab, select one or more database instances that you want to manage. Then, you can perform the following operations based on your business requirements:

   > **Note** You can click **Expand filter** to show more filter conditions.

   - Add an instance

     Click **New** and register a database instance with DMS. For more information, see Register database instances with DMS.

   - Modify the information for multiple instances at a time

     Click **Batch edit**. In the dialog box that appears, modify the instance information and click **OK**.

     > **Note** The database instances that you select must be of the same database type, such as MySQL.

   - Synchronize the data dictionary

     Click **Sync dictionary**. In the message that appears, click **OK**.

     > **Note**
     > - If you change schemas for a database instance by using DMS, DMS automatically synchronizes the data dictionary of the instance.
     > - If you change schemas for a database instance by using a service other than DMS, you must manually synchronize the data dictionary of the instance.

   - Disable or enable one or more instances

     Click **Forbidden instance** or **Enable instance**. In the message that appears, click **OK**.

> **Note**
>
> - After you disable a database instance, the instance is removed from the left-side instance list. DMS users can no longer find databases or tables in this instance in the DMS console.
> - After you enable a database instance, the instance appears in the left-side instance list. Databases in this instance become available. Relevant permissions that have been granted to DMS users on this instance also become valid.

- Remove one or more instances

  Click **Delete Instance**. In the message that appears, click **OK**. After you remove a database instance, the instance is removed from the left-side instance list. DMS users can no longer use databases in this instance in the DMS console. Relevant permissions that have been granted to DMS users on this instance also become invalid and are revoked.

  > **Note** On the **Instance List** tab, you can find database instances in the Delete state and enable these instances to recover them.

- Export configuration information

  Click **Export config**. The browser automatically downloads a CSV file named *instances*. You can use Excel or a text editor to view this file.

- Configure access control

  Click **Access control**. In the dialog box that appears, turn on or off the switch for access control and click **OK**. The IP addresses of DMS servers are automatically added to the whitelists of the specified database instances.

  > **Note** The destination database instances must be ApsaraDB instances.

- More operations

  You can find a database instance and click Details in the Actions column to view the details about databases and tables in this instance. You can also move the pointer over More and perform other operations. For example, you can log on to the instance or modify the information about the instance.

3.4. Manage databases

On the Database List tab, you can manage databases. For example, you can specify database owners, transfer the ownership, revoke owner permissions, grant and revoke user permissions, and export the information about database configurations or permissions.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**. In the left-side navigation pane, click **Instances**.
3. On the Instances page, click the **Database List** tab.
4. Set filter conditions and select one or more databases that you want to manage. Then, you can perform the following operations based on your business requirements:

> **Note** You can click **Expand filter** to show more filter conditions.

- Specify owners

  Specify one or more owners for the selected databases.

- Transfer the ownership

  Transfer the ownership of the selected databases to a user. If you transfer the ownership of multiple databases at a time, you can select only users who own all of the databases.

- Revoke owner permissions

  Revoke owner permissions from one or more owners of the selected databases.

- Grant permissions to users

  Grant the query, export, or change permission on the selected databases to one or more users. You can also specify a validity period for the granted permissions.

- Revoke permissions from users

  Revoke the query, export, or change permission on the selected databases from one or more users. If a user does not have the preceding permissions, the following message appears: `No corresponding permissions. You do not need to recycle or release permissions`.

- Export configurations

  Export the configurations of the selected databases to an Excel file. The configurations include the instance status, environment, DBA, and owner.

- Export permission information

  Export the permission information about the selected databases to an Excel file. The permission information includes the database information, users, permissions, and users who grant the permissions.

- Implement access control

  Click **Access control**. In the Metadata access control dialog box, turn on the **Metadata access control** switch and click **OK**. After the access control feature is enabled, only users who have permissions on the database can find the database.

- Perform other operations

  You can click Tables in the Actions column of a database to view the details about tables in the database. You can also move the pointer over **More** and select the operation that you want to perform. For example, you can query data in a single database, manage permissions, view the details of the instance to which the database belongs, and locate the instance on the Instance List tab.

3.5. Enable metadata access control

DMS provides the metadata access control feature. You can use this feature to allow users to view the information about and access a database or database instance on which they have permissions. Before this feature is enabled, regular users can query all databases and database instances within the current tenant account. After you enable this feature as an administrator, you can allow specific users to view the information about and access the databases or database instances on which they have permissions. This further enhances the data security of your enterprise.

Background information

As a centralized data management service, DMS provides different roles that are assigned different permissions. This helps you manage data in your enterprise in a secure manner. After you enable metadata access control for a database instance or database, only users who have permissions on the database instance or database can view and access the instance or database. This way, users can view and access only databases on which they have permissions. This further enhances data security.

> **Note** In DMS, permissions on a database include the query, export, and change permissions. If you have one of these permissions on a database, you can view the following information about the database:

- Information about the database. You can search for the database by entering a keyword in the search box that appears on the Home page of Data Assets or in the top navigation bar of the DMS console. Alternatively, you can search for the database in the Select the databases, tables, or columns on which you want to apply for permissions field on the Permission Application Ticket page. You can query the data in the database only if you have the query permission on the database.
- Information about the database instance to which the database belongs. To view the information about other databases in this database instance, you must have permissions on the other databases.

You can enable metadata control access for the following objects:

- A user: The user can view and access only databases on which the user has permissions.
- A database: Only users who have permissions on the database can view and access the database.
- An instance: Only users who have permissions on the instance can view and access the instance and the databases in this instance.

Enable metadata access control for a user

1. Log on to the DMS console.
2. In the top navigation bar, click **O&M**. In the left-side navigation pane, click **Users**.

   > **Note** To enable metadata access control for a user, you must be a DMS administrator.

3. Find the user for whom you want to enable metadata access control, move the pointer over **More** in the **Actions** column, and then select **Access control**.
4. In the User access control dialog box, turn on **Metadata access control** and click **OK**.

Enable metadata access control for an instance

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**. In the left-side navigation pane, click **Instances**.

   > **Note** To enable metadata access control for an instance, you must be a DBA or a DMS administrator.

3. On the **Instance List** tab of the Instances page, find the instance for which you want to enable metadata access control, move the pointer over **More** in the **Actions** column, and then select **Access control**.

> **Note** To enable metadata access control for multiple instances at a time, select multiple instances and click **Access control** in the upper part of this tab.

4. In the Metadata access control dialog box, turn on **Metadata access control** and click **OK**.

Enable metadata access control for a database

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**. In the left-side navigation pane, click **Instances**.

> **Note** To enable metadata access control for a database, you must be a DBA or a DMS administrator.

3. On the **Database List** tab, find the database for which you want to enable metadata access control. Move the pointer over **More** in the **Actions** column and select **Access control**.

> **Note** To enable metadata access control for multiple databases at a time, select multiple databases and click **Access control** in the upper part of this tab.

4. In the Metadata access control dialog box, turn on **Metadata access control** and click **OK**.

4.SQLConsole

4.1. Single database query

The single database query feature allows you to execute various SQL statements in the SQLConsole of the Data Management (DMS) console with ease. You can use this feature to visualize the add, delete, modify, and query operations on the data in a database. This feature applies to scenarios, such as data queries and data development.

Prerequisites

You are granted the query permission on the database or table that you want to query.

Precautions

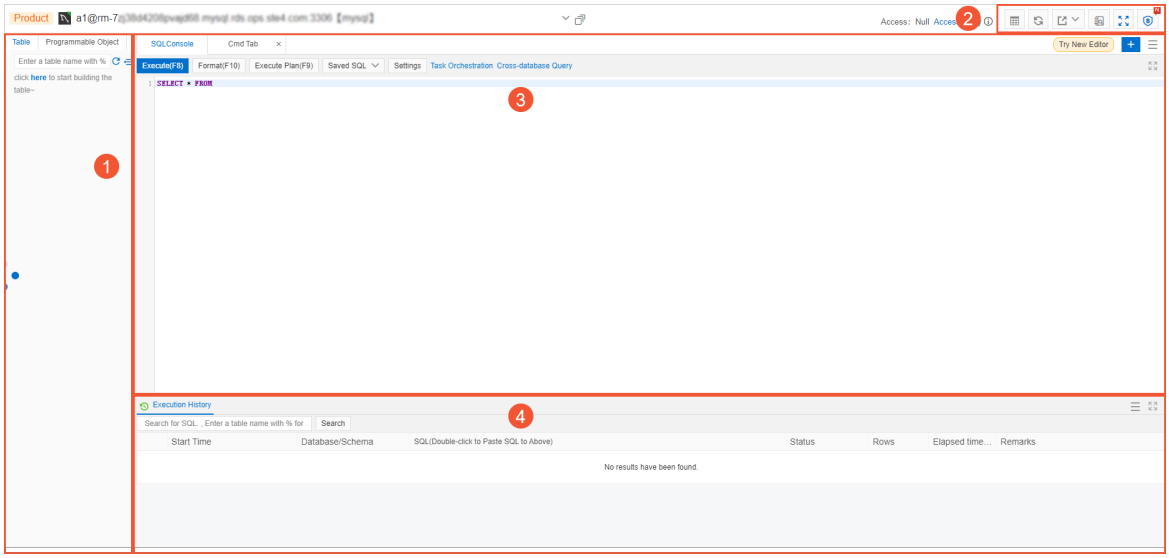
- A table may contain sensitive or confidential fields. You do not have permissions to access these fields. Therefore, the values of these fields are displayed as `******` in the query results. For more information, see Manage sensitive data.
- By default, a maximum of 200 data rows can be returned for each query. If you are an administrator, you can change this value based on your business requirements. To change this value, perform the following steps: 1. Log on to the Data Management (DMS) console. 2. In the top navigation bar, choose **System > Security > Security Rules**.
- A full scan can be performed on a table that does not exceed 10 GB in size. If you are an administrator, you can change this value based on your business requirements. To change this value, perform the following steps: 1. Log on to the DMS console. 2. In the top navigation bar, choose **System > Security > Security Rules**.
- By default, the timeout period to execute a single SQL statement is 60 seconds. If you are an administrator, you can change this value in the **Advanced information** section of the **Edit instance** dialog box.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **More > SQLConsole > Single Database Query**.

   > **Note** To go to the **SQLConsole** tab, you can also double-click the required database in the left-side instance list of the DMS console.

3. Select the database that you want to query from the drop-down list. You can also search for databases by keyword. After you find and select the required database, click **Confirm**.
4. Enter the SQL statement to be executed on the SQLConsole tab and click **Execute**.



GUI of the SQLConsole

No.	Section	Description
①	Visual operation section	In this section, you can visually manage your database. ◦ **Tables** You can view all tables, fields, and indexes of the current database. You can also right-click a table in the database to modify the table schema, import data to the table, or export data from the table. ◦ **Programmable objects** You can create, view, execute, and manage programmable objects, such as views, stored procedures, functions, triggers, and events. **Note** A maximum of 1,000 entries can be displayed. ◦ Key-value pair information **Note** The key-value pair information can be displayed only for a NoSQL database.

No.	Section	Description
②	Extended feature section	In this section, shortcuts to extended features are provided. You can click the icons of the features to use the features. The following table describes the icons. ○ : the **Tables** icon. You can click the icon to view the details about the table. Then, click the icon to return to the SQLConsole tab. ○ : the **Sync Metadata** icon. After you click this icon, DMS collects most recent metadata information about the database, such as tables, fields, indexes, and programmable objects. This way, you can manage permissions on tables, fields, and programmable objects based on the security level. ○ : the Export icon. You can click this icon to export the data of the database, table schemas of the database, or table creation statements. ○ : the **Operation audit** icon. You can click this icon to view the information about all data query and data change records. For example, you can query the information about an operation, the operator, and the time when the operation is performed. For more information, see View operations logs.
③	Command running section	In this section, you can write and execute SQL statements to manage the current database. You can also format SQL statements, create execution plans, save commonly used SQL statements, and configure display settings. **Note** You can click the icon to add multiple query tabs.
④	Execution result section	In this section, you can view the execution results after SQL statements are executed. You can also view the details about a single row and add, delete, or modify data. **Note** You can click the **Execution History** tab to view the historical execution records. For example, you can view the time at which the execution of an SQL statement started, the affected database, and the details about the SQL statement. You can also export the execution results as required.

4.2. Query data across multiple database instances

You can use the cross-database query feature of DMS to query data in databases and tables across multiple instances. This topic describes how to query data across multiple database instances.

Prerequisites

- The type of the database instance whose data you want to query is MySQL, SQL Server, PostgreSQL, PolarDB-X, or Redis.
- Cross-database query can be enabled for only physical databases.

  > **Note**
  > - A physical database is a specific database.
  > - A logical database consists of one or more physical databases. For more information, see Logical database.

- The cross-database query feature is enabled for each database instance whose data you want to query.

  > **Note** If you have not enabled this feature, you can choose **Data Assets > Instances** to modify the advanced information of the instance, enable cross-database query, and customize a database link name.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **SQL Console > Cross-database Query**.
3. Check whether you are granted the permissions to access the database instances.

   > **Note** Parameter descriptions:
   > - Permission:
   >   - Authorized: You are authorized to access the displayed database instances for which the cross-database query feature is enabled and the databases and tables on the database instances. If you are a DMS administrator or a DBA, you are authorized to access all database instances.
   >   - All: You are authorized to access all database instances for which the cross-database query feature is enabled and the databases on the instances within the tenant.
   > - Object:
   >   - DBLink: the database links that are filtered based on the value of the Permission parameter. A database link is established for each database instance.
   >   - Database: the database instances that are filtered based on the value of the Permission parameter.

   In the left-side navigation pane, right-click a database instance and select **View Database Permissions**. In the **Information** message, check whether you are authorized to access the database instance.

- If the message "You have access to the database." appears, you are authorized to access the database instance and you can perform the next step.
- If the message "You do not have permission for this database. Do you want to apply for permission now?" appears, you are not authorized to access the database instance. Perform the following steps to apply for access to the database instance:
  a. Click **Apply for Database Permission**.
  b. In the **Access apply** dialog box, select the permissions for which you want to apply.
  c. Select a period of time from the **Duration** drop-down list based on your business requirements.
  d. In the **Permission** field, enter the reason and background for your application to reduce unnecessary communication and simplify the approval process.

  > Note By default, **You are not authorized to use SQL Console to query data. Click to submit a ticket.** is used.

  e. Click **OK**.

  After your application is approved, you are authorized to access the database instance and you can perform the next step.

4. In the SQLConsole section, enter the SQL statement that is used to query data across the database instances and click **Execute**.

   Sample SQL statement:

   ```
   SELECT *
   FROM dblink1.db1.table1 t1,
        dblink2.db2.table2 t2
   where t1.id= t2.id
   ```

   In the Execution History section, view the execution result of the SQL statement.

   > Note

   - To view the operation logs of cross-database queries.You can also choose **Security and Specifications > Operation Audit > Operation Logs**, and then click Operation Logs.
   - By default, up to 100 rows can be returned for a cross-database query. To view more rows of data, choose **O&M > Configuration Management**. Find the **Maximum number of returned rows for cross-database queries** parameter and click **Change** in the **Actions** column. In the dialog box that appears, specify the maximum number of rows that can be returned for a cross-database query. The maximum number cannot exceed 3,000.
   - If you want to view more than 3,000 rows for a cross-database query, you can use the task orchestration feature to configure a cross-database Spark SQL node, write the execution result set of the SQL statement to a temporary table, and then perform a single-database query on the temporary table.

4.3. Manage schema versions

After you change the schema of a table in a database in Data Management (DMS), DMS adds the latest schema to the schema version list of the database. You can download and compare schema versions and restore an earlier schema version in the schema version list.

Prerequisites

Permissions are granted to your account to query the data of the table or the database to which the table belongs. For more information, see Apply for permissions.

Overview

Schema versions are defined based on a database and store the schema information of all the tables in the database. If the schema of a table in the database is changed, a new schema version is saved. For more information, see Save new schema versions.

If a database instance that has five databases is managed in Security Collaboration mode, each database can contain 50 schema versions. In other words, for a database instance that is managed in Security Collaboration mode, a maximum of 50 schema versions can be retained for each database in the instance.

Usage notes

- The following database engines are supported:
  - MySQL: ApsaraDB RDS for MySQL databases, PolarDB-X databases, AnalyticDB for MySQL databases, and MySQL databases from other sources.
  - SQL Server: ApsaraDB RDS for SQL Server databases and SQL Server databases from other sources.
  - PostgreSQL: ApsaraDB RDS for PostgreSQL databases, AnalyticDB for PostgreSQL databases, and PostgreSQL databases from other sources.
  - OceanBase
  - DamengDB
- The following section shows the maximum number of schema versions that can be retained for each database in instances that are managed in different control modes:
  - Flexible Management: 3
  - Stable Change: 20
  - Security Collaboration: 50
- You cannot manage schema versions for the following databases:
  - Databases that contain more than 1,024 tables
  - System databases such as the information_schema and sys databases in a MySQL database instance

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **SQL Console > SQL Console**.
3. In the dialog box that appears, search for and select a database, and then click **Confirm**.
4. On the SQLConsole tab of the database that you want to manage, move the pointer over the  icon and select **Version management**.
5. On the **Database version list** page, find the version number of the schema that you want to

manage. The following table describes the operations that you can perform.

Operation	Description
View	View the details about the schema version.
Preview script	Preview the SQL script that is used to generate the schema version.
Table structure comparison	Synchronize the schema version to another database, or compare the schema version with a schema version in another database. For more information, see Schema synchronization.
Structural recovery	Synchronize the schema version that you want to restore to an empty database. For more information, see Initialize empty databases.

Save new schema versions

New schema versions are saved when the following operations are performed in DMS:

- SQL statements are executed on the SQLConsole tab to change schemas.
- SQL statements are executed to change schemas when you submit tickets for normal data change, lock-free data change, schema design, or schema synchronization.
- SQL statements are executed to change schemas by a DMS administrator.

> **Note**
>
> If the schema of a table in a database is changed in environments other than DMS, you can synchronize and save the latest schema of the database in DMS.

4.4. Generate a risk audit report

Data Management (DMS) allows you to generate risk audit reports for database instances. Risk audit reports collect and assess various risks that are involved in the O&M of database instances. Risk audit reports also provide optimization suggestions for you to improve the security and stability of your instances.

Overview

A risk audit report is generated based on a database instance in DMS. The report diagnoses and analyzes the risks that are involved in the O&M of the instance or a specific database in the instance.

The following table describes the risk audit items that are contained in risk audit reports.

Risk audit item	Description	Supported database engines

Risk audit item	Description	Supported database engines
SQL audit	For this item, DMS checks whether the SQL statements that are executed in the DMS console to manage the current database instance conform to the R&D specifications. By default, DMS checks the SQL statements that are executed in the previous week. The statements include those that are executed on the SQL Console tab and those that are executed after tickets are submitted, such as Normal Data Modify and Lockless change tickets. For example, DMS may find the following accidental operation: A whole table was accidentally updated because the `WHERE` clause was missing in an `UPDATE` statement. **Note** This audit item is checked based on optimization suggestions for SQL review.	MySQL Self-managed MySQL databases, ApsaraDB RDS for MySQL databases, PolarDB for MySQL databases, PolarDB-X databases, and AnalyticDB for MySQL databases
Metadata	For this item, DMS assesses the risks of all the schemas in the current database instance. For example, DMS may identify the following risk: An auto-increment primary key of the INT type runs out of valid values. **Note** This audit item is checked based on optimization suggestions for SQL review.	MySQL Self-managed MySQL databases, ApsaraDB RDS for MySQL databases, PolarDB for MySQL databases, PolarDB-X databases, and AnalyticDB for MySQL databases
Sensitive Data	For this item, DMS checks whether the current database instance contains sensitive fields. For example, if the instance contains sensitive fields, such as mobile numbers, ID card numbers, or passwords, DMS checks whether these fields are prone to sensitive data breaches.	• MySQL Self-managed MySQL databases, ApsaraDB RDS for MySQL databases, PolarDB for MySQL databases, PolarDB-X databases, and AnalyticDB for MySQL databases • SQL Server Self-managed SQL Server databases and ApsaraDB RDS for SQL Server databases • PostgreSQL Self-managed PostgreSQL databases and PolarDB for PostgreSQL databases • MaxCompute

Limits

- Only DMS administrators, security administrators, database administrators (DBAs), instance owners, and database owners can generate risk audit reports.
- You can keep only a limited number of risk audit reports for an instance. The number varies based on the control mode of the instance.
  - For an instance that is managed in Flexible Management mode, you can keep up to three reports. You cannot view the details of the reports.
  - For an instance that is managed in Stable Change mode, you can keep up to 20 reports.
  - For an instance that is managed in Security Collaboration mode, you can keep up to 50 reports.

Procedure

1. Log on to the DMS console.
2. In the left-side navigation pane of the DMS console, right-click the instance for which you want to generate a risk audit report and choose **Audit > Risk Audit**.

   > **Note** On the SQL Console tab of the database, move the pointer over the icon and select **Risk Audit**.

3. Click **Real-time Diagnostics**.

   > **Note** By default, DMS does not automatically diagnose an instance. If this is the first time for the instance to be diagnosed, you can click **Diagnose**.

4. In the dialog box that appears, select the risk audit items and click **Diagnose**.

   Wait until the status of the report that is being generated changes to **Completed**.
5. After the report is generated, you can view the diagnosis details of each database in the instance.

4.5. Super SQL mode

Data Management (DMS) provides the super SQL mode feature. After you enable this feature as a DMS administrator or a database administrator (DBA), all SQL statements that you execute on the SQLConsole tab are executed without being affected by security rules.

Prerequisites

- You are a DMS administrator or a DBA.
- An instance is managed in Security Collaboration mode.

Context

To enhance the stability and security of databases, DMS administrators and DBAs may configure security rules for the databases. For example, a security rule is configured to prevent unauthorized users from executing DML statements in a production database on the SQLConsole tab. They can execute those statements only by submitting a ticket. However, these security rules may cause inconvenience to privileged users, such as DMS administrators and DBAs.

In view of this, DMS provides the **super SQL mode** feature. If you enable this feature as a DMS administrator or a DBA, all SQL statements that you execute on the SQLConsole tab are executed without being affected by security rules.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, move the pointer over the **More** icon and choose **SQLConsole > Single Database query**.

   > **Note** To go to the **SQLConsole** tab, you can also double-click the database that you want to query in the left-side navigation pane of the DMS console.

3. Select the database that you want to query from the drop-down list or enter a keyword in the field to search for the database. After you select the database, click **Confirm**.
4. On the SQLConsole tab, click the icon in the upper-right corner. In the message that appears, click **OK**.
   Then, the outside borders of the SQLConsole tab turn orange. This indicates that the **super SQL mode** feature is enabled. The SQL statements that you enter on the SQLConsole tab are directly executed.

   > **Notice** After you enable this feature as a DMS administrator or a DBA, all SQL statements that you execute on the SQLConsole tab are executed without being affected by security rules.

   To disable the **super SQL mode** feature, click the icon in the upper-right corner.

5.Data plans

5.1. Schemas

5.1.1. Schema design

Data Management (DMS) provides the schema design feature. This feature allows you to change schemas with ease. This topic describes how to change schemas.

Prerequisites

The destination database is a MySQL, a PolarDB-X, or an ApsaraDB for OceanBase database.

Context

When you create projects, process new business requirements, or optimize business operations, you may need to change schemas. These schema operations include creating and editing tables. For example, you may need to add or delete fields or indexes, adjust field attributes, or adjust the index composition. In these scenarios, you can use the schema design feature of DMS.

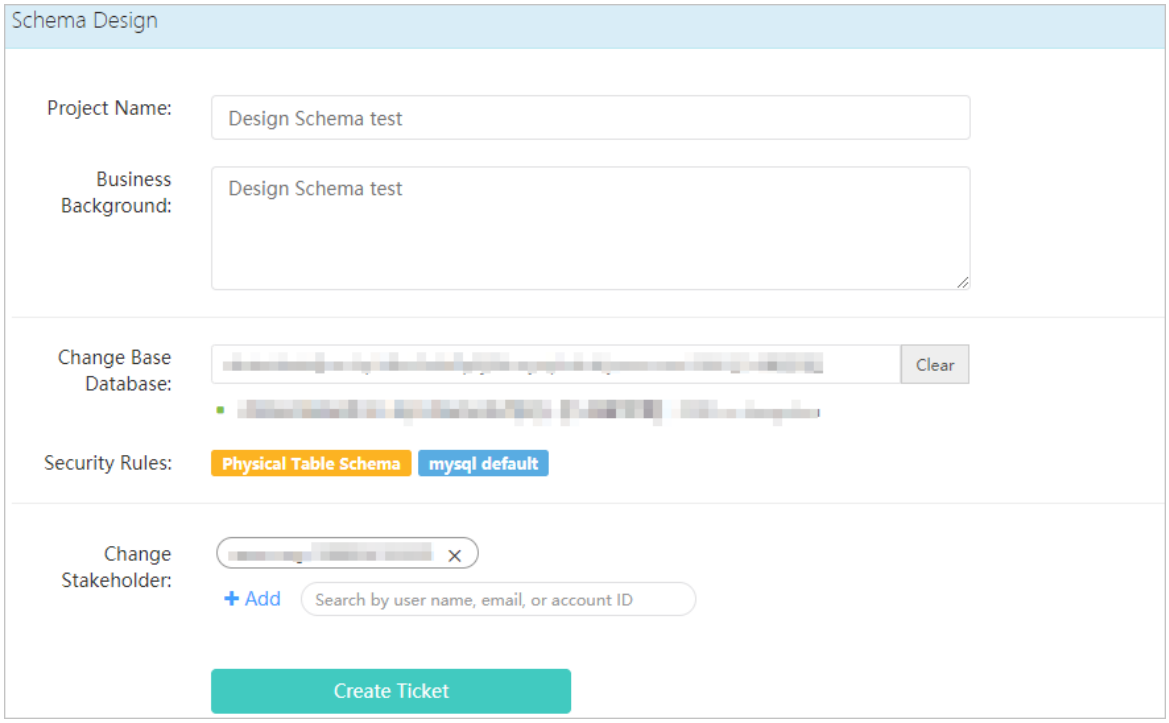
- This feature allows multiple users to simultaneously change a schema in the DMS console at the same time.
- This feature allows you to send verified scripts to other environments. This ensures consistency between schemas in different environments.

Precautions

When you submit a schema design ticket to delete a table, make sure that the table is created by using a schema design ticket.

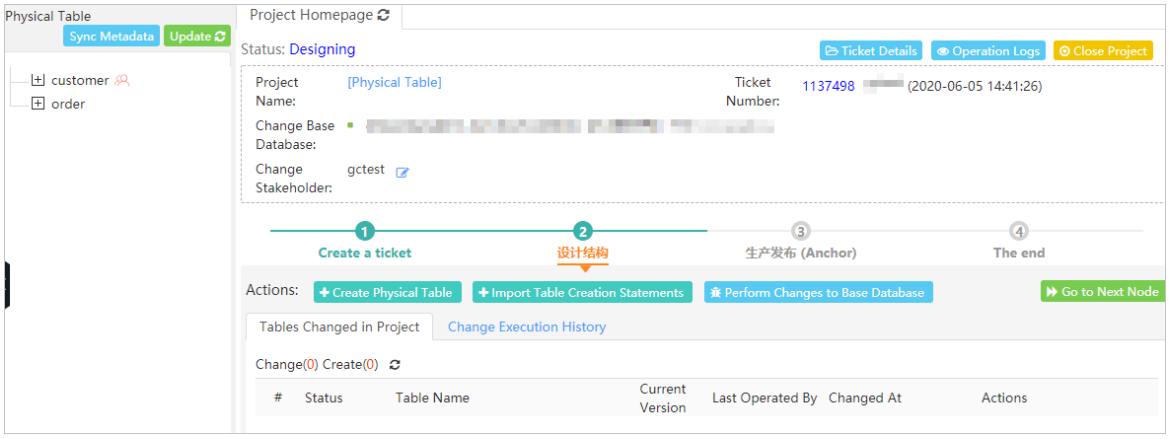
Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **More > Schemas > Schema Design**.
3. In the upper-right corner of the Schema Design tab, click **Schema Design**.
4. On the Schema Design tab, specify the required parameters for a schema design ticket.



Parameter	Description
Project Name	The name of the project. Specify a name that can help you identify the project.
Project description	The background information about the project, such as the purpose or objective of the project. The description is used to reduce communication costs.
Change Base Database	The database whose schema you want to change. You can search for databases by keyword. Prefix match is applied. Only databases on which you have permissions in test or development environments are displayed. **Note** You must have at least the query, export, or change permissions on the database that you select.
Security Rules	No configurations are required. The default setting is specified.
Change Stakeholder	The stakeholders of the changes. The specified stakeholders can view the ticket details and are included in the approval process. Unauthorized users, except for administrators and database administrators (DBAs), cannot view the ticket details.

5. Click **Create Ticket**.
6. Change a schema based on your business requirements.



- Create a table:
  a. Click **Create Physical Table**.

     > **Note** If the destination database is a logical database, click **Create Logical Table**.

  b. On the Create Physical Table tab, set the required parameters. The parameters include the table name, character set, fields, and indexes.
  c. Click **Save**.

     > **Note** After you click Save, DMS verifies the specified information based on design specifications. If the information does not comply with the design specifications, a message appears.

  d. After the information passes the precheck, click **Confirm Changes and Submit to Save**.
- Change the schema of a table:
  a. In the left-side table list, right-click the name of the required table.
  b. On the menu that appears, select **Design Table**.
  c. Change the schema as required and click **Save**.

     > **Note** After you click Save, DMS verifies the specified information based on design specifications. If the information does not comply with the design specifications, a message appears.

  d. After the specified information passes the verification, click **Confirm Changes and Submit to Save**.

7. After the schema is changed, click **Perform Changes to Base Database**.
8. In the Perform Changes to Base Database dialog box, set the Execution Strategy parameter to **Execute Now** or **Schedule**.
9. Click **Submit for Execution** and wait until the ticket is approved.
10. After the ticket is approved, click **Go to Next Node**.

> Note
>
> - After the ticket is approved, DMS applies the changes at the specified point in time. If you do not specify the execution time, the changes are automatically applied after the ticket is approved at the last approval node. You can view the execution status and operation logs. After all changes are applied, you can repeat the preceding procedure to change the schema again. If no additional changes are required for the schema, click Go to Next Node.
> - After the ticket is submitted to the next node, whether you can go back to the previous node is subject to the predefined design specifications.

11. In the Go to Next Node message, click **Go to Next Node**.
12. On the Project Homepage tab, click **Perform Changes to Target Database**.
13. In the Perform Changes to Target Database dialog box, set the Target Database and Execution Strategy parameters and click **Submit for Execution**.

> Note The required database must reside in a production environment.

14. Wait until the ticket is approved and the changes are applied.
15. Click **Go to Next Node**.
    The schema design process ends and the ticket is closed.

5.1.2. Schema synchronization

Data Management (DMS) provides the schema synchronization feature. You can use this feature to compare the schemas of two databases, generate a script to synchronize schemas, and then run the script on the destination database. This topic describes the schema synchronization feature and how to synchronize schemas.

Prerequisites

The source databases and destination databases are ApsaraDB for OceanBase or MySQL databases.

Precautions

- You cannot synchronize schemas to a destination database that resides in a production environment.
- The empty database initialization feature allows you to synchronize some or all tables from a physical or logical database.

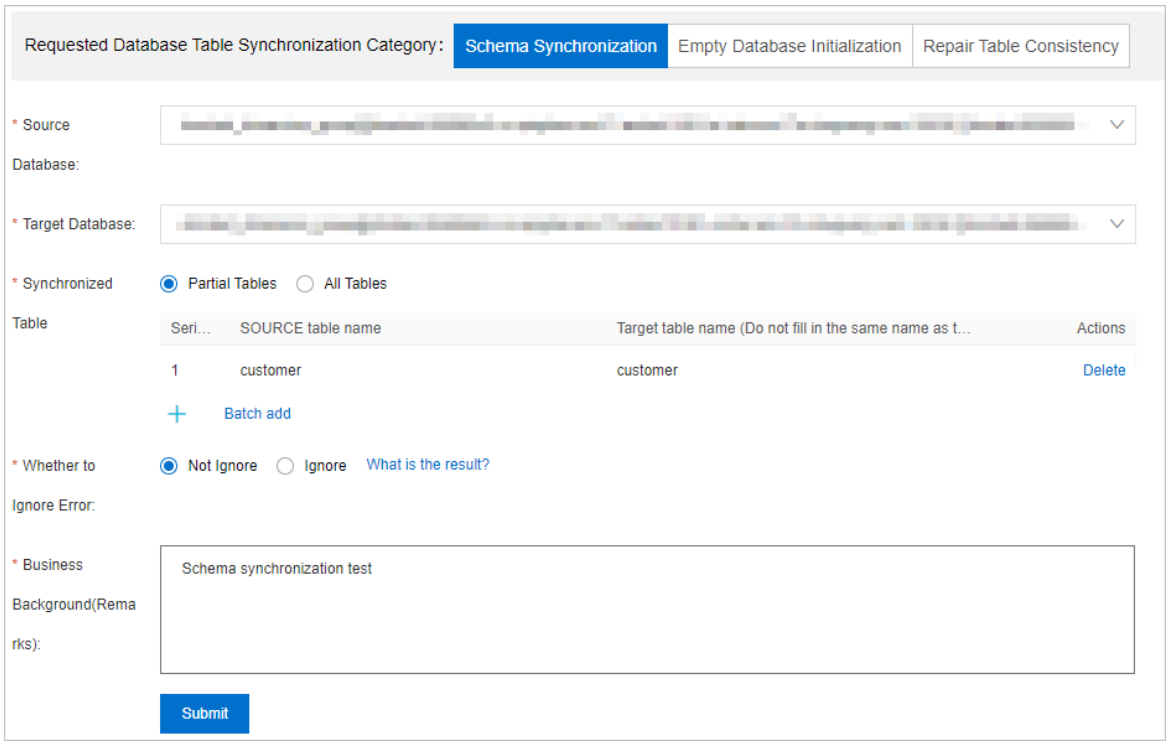
Scenarios

You can use the schema synchronization feature to synchronize schemas and ensure schema consistency in the following scenarios:

- Synchronize data between a database in a production environment and a database in a test environment.
- Synchronize data between different databases that are deployed in a test environment.
- Synchronize data between different databases that are deployed in a production environment.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **More > Schemas > Schema Synchronization**.
3. Specify the required parameters for a schema synchronization ticket.



Parameter	Description
Source Database	The name of the source database from which you want to synchronize schemas. You must have the read permissions on the source database.
Target Database	The name of the destination database to which you want to synchronize schemas. You must have the change permissions on the destination database.
Synchronized Table	The tables that you want to synchronize. Valid values: ○ **Partial Tables**: Synchronize one or more tables in the source database. You can click **Batch Add** to add multiple tables. **Note** If you do not set this parameter, the names of the destination tables are the same as the names of the source tables. ○ **All Tables**: Synchronize all tables in the source database.
Whether to Ignore Error	○ **Not Ignore**: If an error occurs when SQL scripts are executed in serial mode, the system immediately stops executing the current and remaining SQL scripts. ○ **Ignore**: If an error occurs when SQL scripts are executed, DMS stops executing the current SQL script and continues to execute the next statement until all remaining SQL scripts are executed.

4. Click **Submit**. DMS starts to analyze the schemas.
5. Check the comparison results.

   > **Note** If the schemas are changed when the system analyzes the schemas, click Re-analyze in the Schema Analysis step.

6. Verify the script that is used to synchronize schemas and click **Submit and Synchronize to Target Database**.

   > **Note** If the schemas of the source database and destination database are the same, you do not need to submit the script, and the schema synchronization ticket is closed.

5.1.3. Synchronize shadow tables

Data Management (DMS) provides the shadow table synchronization feature to automatically create a shadow table based on the schema of a source table. DMS generates the name of the shadow table by attaching a prefix or suffix to the name of the source table. You can use this feature for end-to-end stress testing.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, move the pointer over the **More** icon and choose **Schemas > Shadow Table Synchronize**.
3. On the **Table/Database Synchronization Application** page, set the parameters that are described in the following table.

Parameter	Description
Source Database	The database whose data is to be synchronized.
Prefix / Suffix	The prefix or suffix that is used to create a shadow table name. The name can be in the **Prefix** + Source table name format or Source table name + **Suffix** format. You can use a custom prefix or suffix as needed. By default, the Prefix + Source table name format is used. Default shadow table name: `__test_Source table name`.
Synchronized Table	The tables whose schemas you want to synchronize. Valid values: ○ **Partial Tables** ○ **All Tables**

Parameter	Description
Synchronization Policy	The policy that is used for shadow table synchronization. Valid values: ○ **Synchronize Now**: DMS immediately synchronizes the tables after you submit the ticket. In this case, the tables are synchronized only once. ○ **Scheduled Synchronization**: DMS synchronizes the tables at the specified time on a regular basis. You can use a crontab expression to schedule synchronization based on your requirements. The minimum interval for synchronization is 1 hour. By default, the shadow tables start to be synchronized at 02:00 every day. For more information, see the Crontab expressions section of this topic.
Whether to Ignore Error	Specifies whether to skip errors that occur when SQL statements are being executed. Valid values: ○ **Not Ignore**: If an error occurs when SQL statements are being executed, DMS stops executing the current and subsequent SQL statements. ○ **Ignore**: If an error occurs when SQL statements are being executed, DMS skips the current SQL statement and continues to execute subsequent SQL statements until all remaining statements are executed.
Business Background(Remarks)	The business background of the project, such as the purposes and objectives of the project.

4. Click **Submit**. DMS starts to analyze the schemas.
5. Check the analysis results.

> **Note** If the schemas are changed during schema analysis, click Re-analyze in the Schema Analysis step.

6. Verify the script that is used to synchronize schemas and click **Submit and Synchronize to Target Database**.

> **Note** If the schemas of the source database and destination database are the same, you do not need to submit the script. The schema synchronization ticket is closed.

Crontab expressions

If you need to schedule the synchronization task to be run in a more precise manner, you can use a crontab expression. The interval for running the task can be specified by using a combination of minutes, hours, days, weeks, or months.

A crontab expression consists of five fields of the NUMERIC type. Valid values of each field:

- **Minutes**: `0 to 59`.
- **Hours**: `0 to 23`. A value of 0 indicates the midnight.
- **Days**: `1 to 31`. A value of this field indicates a specific day of a month.
- **Months**: `1 to 12`. A value of 1 indicates January, and a value of 2 indicates February. Similarly, the specific month that is indicated by a specific value can be obtained.

- **Weeks**: `1 to 7`. A value of 1 indicates Sunday, and a value of 2 indicates Monday. In other words, the seven week days from Sunday to Saturday are indicated by values 1 to 7.

Usage notes

- Specify the time for running a stress testing task by the day or week. You cannot specify the day and week at the same time. After you specify one of the preceding two values, you must set the other value to `?`. A value of ? indicates an unspecified value. For example, if you schedule the task to be run on the first and second days of each month, the **Weeks** field must be set to `?`.
- Limit the characters in a crontab expression to English special characters. The special characters can be wildcards such as asterisks (*) and question marks (?).
- Separate multiple values with commas (,).
- Use a hyphen (-) to indicate a value range. For example, if you set the **Days** field to `1-5`, the task is scheduled to be run on the first to fifth days of a month.
- Use a forward slash (/) to indicate an interval for running the task. For example, if you set the Days field to `*/2`, the task is scheduled to be run every two days.

Crontab expression examples

- To schedule the task to be run at 23:00 every Saturday and Sunday, use the following crontab expression: 0 23 ? * 7,1.
- To schedule the task to be run at 09:30 on the fifth, fifteenth, and twenty-fifth days of each month, use the following crontab expression: 30 9 5,15,25 * ?.
- To schedule the task to be run at 00:00 every two days, use the following crontab expression: 0 0 */2 * ?.

5.1.4. Initialize empty databases

DMS provides the empty database initialization feature. You can use this feature to compare the schemas of two databases, generate a script for schema synchronization, and then run the script to synchronize the schema from the source database to the destination database. To use this feature, make sure that the destination database is empty. This topic describes how to initialize an empty database.

Prerequisites

- The source and destination databases are MySQL or ApsaraDB for OceanBase databases.
- The destination database is an empty database that does not contain tables.

Usage notes

The empty database initialization feature allows you to synchronize some or all tables from a physical or logical database.

Scenarios

You can use this feature to synchronize the schemas of databases that are deployed across multiple regions and units. For example, you can use this feature in the following scenarios:

- Synchronize schema between a database in an online environment and a database in an offline environment.
- Synchronize schema between different databases that are deployed in offline environments.
- Synchronize schema between different databases that are deployed in online environments.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **Schema Change > Empty Database Initialization**.
3. On the Ticket Application page, set the parameters for the empty database initialization ticket. The following table describes the parameters.

Parameter	Description
Source Database	The name of the source database from which you want to synchronize schemas. You must have the query permissions on the source database.
Target Database	The name of the destination database to which you want to synchronize schemas. You must have the change permissions on the destination database. **Note** The type of the destination database must be the same as the type of the source database.
Initialized Table	The one or more tables whose schemas you want to synchronize. Valid values: ○ **Partial Tables**: synchronizes the schemas of some of the tables in the source database. You can also click the + icon to add more tables. ○ **All Tables**: synchronizes the schemas of all tables in the source database.
Whether to Ignore Error	○ **Not Ignore**: If an error occurs when SQL statements are being executed in serial mode, DMS stops executing the current and remaining SQL statements. ○ **Ignore**: If an error occurs when SQL statements are being executed, DMS stops executing the current SQL statement and continues to execute the next statement until all remaining SQL statements are executed.
Business Background(Remarks)	The purpose or objective of the initialization operation.

4. Click **Submit**. DMS starts to analyze the schemas.
5. Check the analysis results.

   Note If the schemas are changed during schema analysis, click Re-analyze in the Schema Analysis step.

6. Verify the script that is used to synchronize schemas and click **Submit and Synchronize to Target Database**.

> **Note** If the schemas of the source database and destination database are the same, you do not need to submit the script. The schema synchronization ticket is closed.

5.1.5. Repair table consistency

DMS provides the table consistency repairing feature. This feature is used to compare schemas between tables in databases that are deployed in different environments, provides an efficient way to identify schema differences, and execute SQL statements that are specific to the required environment. This ensures schema consistency between different environments.

Prerequisites

The source databases and destination databases are MySQL or ApsaraDB for OceanBase databases.

Scenarios

- Ensure the schema consistency between physical tables that are deployed in the test environment and the production environment.
- Ensure the schema consistency between physical tables and logical tables in a physical database or a logical database.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **More > Schemas > Table Consistency Repairing**.
3. On the Table/Database Synchronization Application tab, set the required parameters to create a Repair Table Consistency ticket.

Parameter	Description
Base Database(Physical Database)	The source database based on which schema consistency is to be repaired. You must have the query permissions on the source database.
Target Database	The destination database whose data is to be modified. You must have the change permissions on the destination database.
Repaired Table	The tables between which schema consistency is to be repaired. To add tables, click the + icon and specify the required table names. **Note** If you do not specify the destination table name, the system names the destination table after the name of the specified source table.

Parameter	Description
Whether to Ignore Error	○ **Not Ignore**: If an error occurs when SQL statements are being executed in serial mode, the system immediately stops executing the current and remaining SQL statements. ○ **Ignore**: If an error occurs when DMS is executing an SQL statement, DMS stops executing the current SQL statement and continues to execute the remaining SQL statement.
Business Background	The business background of the ticket. This parameter reduces communication costs.

4. Click **Submit**. DMS starts to analyze the schemas.
5. Check the analysis results.

   > **Note** If the schemas are changed during schema analysis, click Re-analyze in the Schema Analysis step.

6. Verify the script that is used to synchronize schemas and click **Submit and Synchronize to Target Database**.

   > **Note** If the schemas of the source database and destination database are the same, you do not need to submit the script. The schema synchronization ticket is closed.

5.2. Change data

DMS provides some data change features that allow you to change data. This topic describes how to use the data change features to change data.

Context

DMS allows you to submit data change tickets to initialize data for a newly published project, clear historical data, fix bugs, or run a test. The operations that you can perform to change data include insert, update, delete, and truncate operations.

Data change features

Feature	Description

Feature	Description
Normal Data Modify	This feature allows you to perform the following types of data changes: • Regular data changes. • Lock-free schema changes. You can perform this type of operations to change character sets and collations for tables, adjust time zones, and change column data types. Compared with regular data change operations, lock-free schema changes can be performed to achieve the following benefits: ◦ Prevents table locking caused by database schema changes to ensure business continuity. ◦ Prevents latency caused by native online DDL operations that are performed on databases to ensure consistent synchronization between the primary and secondary databases. ◦ Reclaims tablespaces and reduces fragmentation rates by performing lock-free schema changes instead of executing the OPTIMIZE TABLE statement that results in table locking. **Note** You can use this feature only for MySQL databases. Before you use this feature, you must set the Lock-free Schema Change parameter to Open (DMS OnlineDDL first) in the **Advanced information** section when you register or edit a database instance. For more information, see Register database instances with DMS.
Lockless change	This feature allows you to change a large amount of data. For example, you can use this feature to delete historical data and update all fields in a table. Multiple SQL statements for data changes are divided and executed at the same time based on the primary key or unique key. This way, impacts on database performance and space are reduced. **Note** You can use this feature only for MySQL databases.
History Data Clean	This feature allows you to regularly clean historical data. This way, the stability of the online environment is not affected when you obtain historical data. **Note** You can use this feature only for MySQL databases.
Large Data Import	This feature allows you to quickly import a large amount of data to databases. This reduces the costs of labor and material resources. **Note** You can use this feature for the following types of databases: • Self-managed MySQL databases and ApsaraDB RDS for MySQL databases • PolarDB-X databases

Feature	Description
Programmable Object	Databases provide programmable objects such as stored functions and stored procedures. This feature allows you to use programmable objects to standardize management processes and provides audit records.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, click **Data Change** and select the data change feature that you want to use.
3. Set the parameters for the data change ticket.

   The following table describes the parameters for a **Normal Data Modify** ticket.

Parameter	Description
Database	The database on which you want to perform the change operation. You must select one or more databases on which you have the change permissions. If you have only the query permissions on a database or the change permissions on the tables in the database, you cannot submit a data change ticket for the database.
Reason Category	The reason for the change operation. This helps you find the ticket in subsequent operations.
Business Background	The purposes or objectives of the change operation. This reduces unnecessary communication.
Execution Method	The method that is used to execute the ticket. Set this parameter based on your business requirements.
Affected Rows	The estimated number of data rows that are affected by the change operation. To obtain the actual number of affected rows, you can write an SQL statement that includes the COUNT function on the SQLConsole tab.
SQL Statements for Change	The executable SQL statements that are used to perform the change operation. You can write the SQL statements in the SQL Text field or upload a file to provide the SQL statements. DMS verifies the syntax of the SQL statements when you submit the ticket. If the syntax is invalid, you cannot submit the ticket.
SQL Statements for Rollback	The executable SQL statements for rolling back the change operation. You can write the SQL statements in the SQL Text field or upload a file to provide the SQL statements.
Change Stakeholder	The stakeholders involved in the change operation. All specified stakeholders can view the ticket details and take part in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Attachments	The images or files that provide more information about the change operation.

4. After you set the parameters, click **Submit**.
5. After the ticket passes the precheck, click **Submit for Approval**. In the message that appears, click **OK**.
6. After the ticket is approved, click **Execute Change**.
7. Set the Execute Immediately parameter and click **Confirm Execution**.

   > **Note** By default, the Execute Immediately switch is turned on. You can turn off the **Execute Immediately** switch and specify a point in time to execute the ticket. DMS automatically executes the ticket at the specified point in time.

   Wait until the execution is complete.

5.3. Export data

DMS provides the data export feature. You can use this feature to export a database or SQL result sets. Then, you can extract the required data for data analysis.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, click **Data Export** and click **SQL Result Set Export** or **Database Export**.
3. On the Ticket Application page, set the parameters for the SQL Result Set Export ticket or Database Export ticket.
   - The following table describes the parameters for a SQL Result Set Export ticket.

Parameter	Description
Database Name	The database from which you want to export an SQL result set. You must select a database on which you have the export permissions.
Reason Category	The reason for this export operation. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of this export operation. This reduces unnecessary communication.
Affected Rows	The estimated number of data rows that are affected by this export operation. To obtain the actual number of affected rows, use the `COUNT` function in SQL statements on the SQLConsole tab.
Skip Validation	Specifies whether to skip validation. If you select **Skip Validation**, you must enter a reason in the field next to the check box. **Warning** After you select **Skip Validation**, DMS does not check the number of rows that may be affected by this export operation. If the amount of data to be exported is large, your business may be affected. Proceed with caution.

Parameter	Description
Stakeholder	The stakeholders involved in this export operation. All specified stakeholders can view the ticket details and take part in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Export Statement	The executable SQL statement that is used to export data. Example: `select * from testtable`. DMS verifies the syntax of the SQL statement when you submit the ticket. If the syntax is invalid, you cannot submit the ticket.
Attachments	The images or files that provide more information about this export operation.

- The following table describes the parameters for a Database Export ticket.

Parameter	Description
Database Name	The database that you want to export. You must select a database on which you have the export permissions. After you select the database, you can select the tables from which you want to export data and configure filter conditions in the Tables & Filters section.
Exported table	■ Specifies whether to export specific tables or all tables in the database. If you select Partial Tables, you must select one or more tables in the Tables & Filters section. ■ If you select All Tables, you export all tables in the database.
Reason Category	The reason for this export operation. This helps you find the ticket in subsequent operations.
Business Background	The purpose or objective of this export operation. This reduces unnecessary communication.
Stakeholder	The stakeholders involved in this export operation. All specified stakeholders can view the ticket details and take part in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.
Export content	The type of data that you want to export. Valid values: **Data**, **Structure**, and **Data & Structure**.
File Format	The format of the exported file. Valid values: SQL, CSV, and EXCEL.
Exported Structure Type	The type of schema that you want to export.
More Options	The other objects that you want to export. Click **Big data type export options** or **SQL script other options** and select the options as required.
Attachments	The images or files that provide more information about this export operation.

4. After you set the parameters, click **Submit**.

   > **Note** If you export an SQL result set, DMS prechecks the SQL statements. After the SQL statements pass the precheck, click **Submit for Approval**. In the message that appears, click **OK**.

5. After your ticket is approved, go to the **Home** page and click **Submitted Tickets**.
6. Find the data export ticket that is submitted and click the ticket number.
7. In the **Download** section, click **Download Exported File**.

5.4. Perform SQL reviews

DMS provides the SQL review feature that allows you to remove SQL statements that do not use indexes or do not conform to database development standards. This way, the risk of SQL injection attacks is reduced.

Prerequisites

SQL reviews are performed before the related code is published to an online environment. Therefore, you must set the environment type of the relevant database instance to **Test** in the DMS console.

Context

When you develop a project, you must execute SQL statements on databases to add, delete, modify, and query data so that you can implement business logic and visualize data. Before the project is published, you must review all SQL statements that you want to execute. You must ensure that all SQL statements conform to database development standards to ensure business continuity.

If all SQL statements require to be manually reviewed by DBAs in sequence, substantial human resources are consumed, and the efficiency of R&D is reduced. The SQL review feature of DMS can quickly review SQL statements and provide optimization suggestions.

Usage notes

- Only XML or TXT files can be uploaded.
- Tables that are specified in SQL statements must exist in the specified database. Otherwise, DMS cannot review these SQL statements.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **SQL Review > SQL Audit Ticket**.
3. In the Application step, set the parameters for the SQL review ticket. The following table describes the parameters.

Parameter	Description
Project Name	Enter a project name based on your business requirements so that the ticket can be distinguished from other tickets in subsequent operations.

Parameter	Description
Database	Select a database in the test environment for the project as the destination database. You must have the change permissions on the database.
Business Background	Enter the business background of the project to help relevant users obtain the details of the project.
Change Stakeholder	Select one or more stakeholders involved in the ticket.
Upload a file	◦ If you want to upload a file, click **Upload** to upload the file. Note ■ iBATIS and MyBatis files in the XML format can be uploaded. ■ SQL statements must be saved as TXT files. Separate multiple SQL statements in a file with semicolons (;). ■ To remove an added file, move the pointer over Delete in the Operation column and click **Confirm**. ◦ If you want to enter text, click **Enter text**, write SQL statements or enter texts in the XML format, and then click **Save**.

4. Click **Submit**.
5. View the result of the SQL review.

   Note
   - If the SQL statements conform to database development standards and use indexes, the result indicates that the SQL review succeeds and no recommended indexes are provided.
   - If the SQL statements conform to database development standards but do not use indexes, the result indicates that the SQL review succeeds and recommended indexes are provided.
   - If the SQL statements do not conform to database development standards, the result indicates that the SQL review fails.

6. If an SQL statement fails the SQL review, click **View reason** to check the reason. You can also find the review record in the Check Result step and click **Details**, **Adjust SQL**, or **More** in the **Operation** column to perform the required operations.

   Note After you modify the SQL statements, click **Confirm** to allow the system to review the SQL statements again. For dynamic SQL statements in XML files, you must enumerate each combination of SQL statements.

7. After all SQL statements pass the SQL review, click **Submit for Approval** in the **Approval** step.

> **Note** The approval process of the ticket varies based on the security rules that are configured for the current database instance.

5.5. Clone databases

The database clone feature allows you to replicate data at the database level. This topic describes how to use the database clone feature.

Prerequisites

- The source and destination databases are MySQL databases.
- The database instances to which the source and destination databases belong are managed in Flexible Management mode. You have logged on to the database instances in the DMS console.

Scenarios

- Create a full database backup.
- Initialize databases that are deployed in different environments, such as development and test environments.
- Copy data from a database in an online environment to a database in an offline environment for data processing and analysis.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **Environment Construction > Database Clone**.
3. In the upper-right corner, click **Database Clone**.
4. In the Apply step, set the parameters for the Database Clone ticket. The following table describes the parameters.

Parameter	Description
Task Name	The name of the task. Enter a name that can help you identify and manage the task in subsequent operations.
Source database	The source database whose data you want to clone. You can enter a keyword to search for a database and select the database from the matched results.
Target database	The destination database to which you want to write the cloned data. You can enter a keyword to search for a database and select the database from the matched results. **Note** The destination database must be different from the source database.

Parameter	Description
Select source table	The one or more tables that you want to clone from the source database. You can enter a keyword to search for a table and select the table from the matched results. **Note** To clone all tables, set this parameter to **All Tables**.
Duplicate objects	The method that is used to handle object name conflicts. Set this parameter based on your business requirements. Valid values: ○ **Skip duplicate name object**: If two objects have the same name, DMS does not clone the object in the source database. ○ **Overwrite duplicate name object**: If two objects have the same name, the schema and data of the object in the destination database are overwritten by the schema and data of the object in the source database.
Migration Objects	The objects that you want to clone. In addition to tables, you can simultaneously clone other objects from the source database to the destination database. These objects include views, stored procedures, functions, triggers, and events.
Time options	The time when you want to run the database cloning task. Valid values: **Running immediately** and **Specified time**. If you set this parameter to **Specified time**, you must specify a date and time to run the task. ○ **Running immediately**: The task is run immediately after the ticket is approved. ○ **Specified time**: DMS automatically runs the task to clone data at a specified point in time.

5. After you set the parameters, click **Submit**.
6. After the ticket is approved, the task is automatically run at a specified point in time.

5.6. Generate test data

DMS provides the test data generation feature that allows you to quickly generate data. You can generate test data for functional or performance tests.

Prerequisites

- A relational database, such as a self-managed MySQL database, an ApsaraDB RDS for MySQL database, an AnalyticDB for MySQL database, or a PolarDB-X database is used.
- A table is created. You can use the schema design feature to create a table. For more information, see Design a schema.

Context

Functional tests or performance tests often require test data. You can use one of the following methods to generate test data:

- Write test data: This method is low in efficiency and is not applicable to scenarios in which a large

amount of test data is required.

- Use scripts: This method requires high costs and the data that is generated by using this method cannot meet discreteness requirements.
- Export data from a production environment as test data: This method is not secure and may cause data leaks.

DMS provides the test data generation feature that allows you to generate test data in a quick, efficient, and secure manner. You can use this feature to control the discreteness of the data that is generated.

Usage notes

- You can use this feature to generate test data in only one table at a time. To generate test data in multiple tables, submit a ticket for each table.
- To prevent database overload due to the instantaneous generation of excessive data, DMS allows you to perform traffic throttling. Check the following examples for your reference.
  - About 1 minute is required to generate one million rows of data in a table that has four fields.
  - About 2 to 3 minutes are required to generate one million rows of data in a table that has 40 fields.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **Environment Construction > Test Data Generation**.
3. In the upper-right corner, click **Test Data Generation**.
4. In the Application step, set the parameters for the test data generation ticket. The following table describes the parameters.

Parameter	Description
Task Name	The name of the task. Enter a name that can help you identify and manage the task in subsequent operations.
Database Name	The database to which the table in which you want to generate test data belongs.
Table Name	The name of the table in which you want to generate test data. You can enter a keyword to search for a table and select the table from the matched results. After you select a table, the **Configure the algorithm** parameter appears and displays the field information of the table. **Note** You can use this feature to generate test data in only one table at a time. To generate test data in multiple tables, submit a ticket for each table.

Parameter	Description
Configure the algorithm	The algorithms that are used to generate test data. To configure the algorithm for a field, you can click the value in the Generation mode column that corresponds to the field. Then, set the parameters in the Generation mode dialog box based on your business requirements. **Note** For example, you can use the random, customize, or enumeration algorithm to generate test data of the STRING type. The customize algorithm can be used to generate multiple industry-standard types of data.
Number of rows generated	The number of data rows that you want to generate.
Conflict Handling	The method that is used to handle data conflicts. Set this parameter based on your business requirements.
Change Stakeholder	The stakeholders involved in the ticket. All specified stakeholders can view the ticket details and assist in the approval process. Irrelevant users other than DMS administrators and DBAs are not allowed to view the ticket details.

5. After you set the parameters, click **Submit**.
6. After the ticket is approved, DMS automatically generates test data as specified.

5.7. DevOps

5.7.1. Manage iterations

DMS provides the DevOps iteration feature that allows you to advance an R&D process stage by stage. You can create specific types of tickets in each iterative stage. This facilitates collaborative development and improves the efficiency of R&D processes.

Create an iteration

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **R&D Space > DevOps**.
3. Click the **Iteration** tab.
4. Click **Create Iteration**.
5. In the **Create Iteration** dialog box, set the parameters and click **OK**. The following table describes the parameters.

Parameter	Description
Iteration Name	The name of the iteration.
Project	The project to which the iteration belongs. For more information, see Manage projects.

Parameter	Description
Iteration Template	The iteration template that you want to use. ○ The iteration template contains all the stages of an R&D process. ○ The iteration template specifies the requirements that must be met in each iterative stage before the iteration is advanced to the next stage. ○ The iteration template specifies various types of tickets that can be created in each iterative stage. **Note** ○ Each iteration can be associated with only one iteration template. ○ You cannot change the associated iteration template after an iteration is created. If you need to use another iteration template, create another iteration. For more information, see Manage iteration templates.
Participants	The one or more participants of the iteration. An iteration participant can perform the following operations: ○ Create a ticket in each iterative stage. ○ Advance the iteration to the next stage. **Note** Only the creator of an iteration, project administrators, DBAs, and DMS administrators can add or remove an iteration participant.
Available Databases	The one or more databases that you want to use during the iteration. **Note** Only the databases that are available in the selected project can be selected. If no database is selected when you create the project, all the databases of the current tenant can be selected.
Description	The description of the iteration. Enter an informative description that can help you manage the iteration.

6. Click the name of the iteration that you create. The **Iteration Details** page appears.
7. Advance the iteration.

i. Move the pointer over **Create Ticket** and select a ticket type. The available types of tickets in each iterative stage are specified by the iteration template.

> **Note** When you create a ticket, the Database drop-down list displays only the databases that meet the following conditions:
>
> - The database is available for the iteration.
> - The type of the environment to which the database belongs meets the requirements of the iteration template.

ii. After all required tickets of an iterative stage are executed, click **Enter Next Stage**. In the message that appears, click **OK** to advance the iteration to the next stage.

> **Note**
>
> - If a ticket fails the status check, the iteration cannot be advanced to the next stage. The conditions that must be met before an iteration can be advanced to the next stage are specified by the iteration template.
> - After the iteration is advanced to the next stage, you cannot create tickets in the previous stage.

8. Repeat Step 5 until the iteration is complete or closed.

5.7.2. Manage projects

DMS provides the DevOps feature. This feature allows you to specify participants, databases, and iteration templates for a project. This can efficiently ensure a smooth R&D process, reduce accidental operations, and ensure data security.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **R&D Space > DevOps**.
3. Click the **Project** tab.
4. Click **Create Project**.
5. In the **Create Project** dialog box, set the parameters and click **OK**. The following table describes the parameters.

Parameter	Description
Project Name	The name of the project.
Project administrator	The administrator of the project. You can select one or more administrators. By default, the project creator is selected. A project administrator can perform the following operations: ○ View and modify the basic information about the project. ○ Be assigned to an approval node of an approval process in an iterative stage.

Parameter	Description
Project tester	The tester of the project. You can select one or more testers. A tester can be assigned to an approval node of an approval process in an iterative stage.
Participants	The participant of the project. You can select one or more participants. A project participant can perform the following operations: ○ View the basic information about the project. ○ Create an iteration in the project. **Note** Only project administrators, DBAs, and DMS administrators can add or remove a project participant.
Available Iteration Templates	The iteration template that is available in the project. You can select one or more iteration templates. ○ If one or more iteration templates are selected, only the selected iteration templates are available when you create an iteration in the project. ○ If no iteration template is selected, all the iteration templates in the Available Iteration Templates drop-down list are available when you create an iteration in the project. **Note** After an iteration template is associated with a project, the configurations of existing iterative stages can be modified. However, you cannot create or delete an iterative stage. For more information, see Manage iteration templates.
Available Databases	The database that is available in the project. You can select one or more databases. ○ If one or more databases are selected, only the selected databases are available when you create an iteration in the project. ○ If no database is selected, all the databases of the current tenant are available by default when you create an iteration in the project.
Project Description	The description of the project. Enter an informative description that can help you manage the project.

6. (Optional) Create an iteration

   Click the name of the project that you create. On the project details page, click **Create Iteration** to create an iteration. For more information, see Manage iterations.

7. (Optional) Manage the project
   - Modify the basic information about the project.
     - To modify the name of the project, click the icon next to the project name.
     - To modify the description of the project, click the icon next to **Project Description**.

- To add or remove databases that are available in the project, click **View** next to **Available Databases**.
  - To add available databases to the project, select the databases that you want to add in the **All Databases** section and click the icon. Then, the databases are displayed in the **Available Databases** section.
  - To remove available databases from the project, select the databases that you want to remove in the **Available Databases** section and click the icon. The databases are removed from the **Available Databases** section.
- To add or remove iteration templates that are available in the project, click **View** next to **Available Iteration Templates**.
  - To add available iteration templates to the project, select the iteration templates that you want to add in the **All Iteration Templates** section and click the icon. Then, the iteration templates are displayed in the **Available Iteration Templates** section.
  - To remove available iteration templates from the project, select the iteration templates that you want to remove in the **Available Iteration Templates** section and click the icon. The iteration templates are removed from the **Available Iteration Templates** section.

- Manage the members of the project.
  - To add an administrator to the project or remove an administrator from the project, click **Change** next to **Administrator** in the **Project Member Management** section.
  - To add a tester to the project or remove a tester from the project, click **Change** next to **Testers** in the **Project Member Management** section.
  - To add a participant to the project or remove a participant from the project, click **Change** next to **Participants** in the **Project Member Management** section.
- Manage the project.
  - To create a similar project, click **Create As** in the upper-right corner. In the **Replicate Project** dialog box, set the **Name** and **Description** parameters. Then, click **OK**.
  - To view the operation records of the project, click **Operation History** in the upper-right corner. You can view the operation time, operator, and logs of each operation performed on the project.
  - To close the project, click **Close Project** in the upper-right corner. In the message that appears, click **OK**.

    Notice After the project is closed, you cannot open the project again. Proceed with caution.

5.7.3. Manage iteration templates

The iteration template feature of DMS allows you to customize R&D processes and manage the quality of R&D processes. This feature can be used in combination with various types of tickets such as data change, schema design, and SQL review. This topic describes how to create and configure an iteration template.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Database Development**. In the left-side navigation pane, choose **R&D Space > DevOps**.
3. Click the **Iteration Template** tab.
4. Click **Create Iteration Template**.
5. In the **Create Iteration Template** dialog box, set the parameters and click **OK**. The following table describes the parameters.

Parameter	Description
Template Name	The name of the iteration template.
Usage Scope	The scope within which the iteration template is available. Valid values: ○ **Available for All Users**: This iteration template can be used by all the users of the current tenant. ○ **Specific Users**: This iteration template can be used by only specific users.
Description	The description of the iteration template. Enter an informative description that can help you manage the iteration template.

6. Click the name of the iteration template that you create. The **Template Details** page appears.
7. Manage iterative stages based on your development requirements.
   - To create an iterative stage, click **Add** in the left-side pane, enter a name in the field, and then click a blank area on the page.
   - To modify the name of an iterative stage, click the icon next to the name of the iterative stage that you want to modify.

     Note By default, a new iteration template contains two iterative stages: **Dev** and **Product**. These two stages can be modified and deleted.

   - To delete an iterative stage, click the icon next to the name of the iterative stage that you want to delete. In the message that appears, click **OK**.
8. Click an iterative stage and configure rules for the stage. Repeat this step for each stage.

i. Click the **Rule base configuration** tab and configure the rules that are described in the following table.

Rule	Description	Operation
Database environment type	Specifies one or more types of environments to which databases belong. You can select the databases in the selected environments when you create tickets in the iterative stage.	Click the value in the Parameter column that corresponds to the **Database environment type** rule. From the drop-down list, select one or more environment types. You can also remove the selected environment types.
Stage promotion personnel authority	Specifies one or more types of users who have the permissions to advance the iteration to the next stage.	Click the value in the Parameter column that corresponds to the **Stage promotion personnel authority** rule. From the drop-down list, select one or more user types. You can also remove the selected user types. **Note** Only the creator and participants of an iteration can advance the iteration to the next stage.
Fallback management	Specifies whether the iteration can be rolled back after it is advanced to the next stage.	Click the value in the Parameter column that corresponds to the **Fallback management** rule. From the drop-down list, select **Allow Rollback** or **Rollback not allowed**. **Note** The Fallback management rule is not displayed for the first iterative stage of the iteration template.

ii. Click the **Stage Work Orders and Checkpoints** tab and configure rules for tickets.

- To create a ticket rule, click **Create Ticket Rule**.
  - Select a ticket type from the **Ticket Type** drop-down list. Valid values:
    - **Data change**: data change tickets, including the Normal Data Modify, Lockless change, Data Import, and Programmable Object tickets.
    - **Structural design**: schema design tickets.
    - **SQL audit**: SQL review tickets.
    - **Library table synchronization**: database and table synchronization tickets, including the Schema Synchronization and Empty Database Initialization tickets.

Note You can specify the types of tickets that can be created in the iterative stage.

- Select **Yes** or **No** from the **Required** drop-down list.
  - **Yes**: You must create a ticket of the specific type in the iterative stage. Otherwise, the iteration cannot be advanced to the next stage.
  - **No**: The iteration can be advanced to the next stage regardless of whether you create a ticket of the specific type in the iterative stage.
- From the **Stage Status (Click Value to Modify)** drop-down list, specify whether to check the status of a ticket and select the ticket state that allows the iteration to be advanced to the next stage.

Ticket type	Operation
Data change	■ **Do not check**: does not check the status of the ticket and allows the iteration to be advanced to the next stage. ■ **Successful execution**: allows the iteration to be advanced to the next stage if the data change is performed. ■ **Closed**: allows the iteration to be advanced to the next stage if the ticket is closed.
Structural design	■ **Do not check**: does not check the status of the ticket and allows the iteration to be advanced to the next stage. ■ **Design node completed**: allows the iteration to be advanced to the next stage after the design node of the schema design ticket is complete. ■ The nth node has been completed: allows the iteration to be advanced to the next stage after the nth node of the schema design ticket is complete. **Note** ■ Valid values of n: 1 to 7. ■ To view the configuration of each node in a schema design ticket, go to the Details page of security rule set for the database that you want to manage and click the **Schema Design** tab. On this tab, find the **R & D process** rule under the **Basic Configuration Item** checkpoint and click Edit in the Actions column. ■ **Published**: allows the iteration to be advanced to the next stage if the schema design is published. ■ **Work order closed**: allows the iteration to be advanced to the next stage if the schema design is complete. ■ **Closed**: allows the iteration to be advanced to the next stage if the ticket is closed.

Ticket type	Operation
SQL audit	■ **Do not check**: does not check the status of the ticket and allows the iteration to be advanced to the next stage. ■ **Audit successful**: allows the iteration to be advanced to the next stage if the SQL review is complete.
Library table synchronization	■ **Do not check**: does not check the status of the ticket and allows the iteration to be advanced to the next stage. ■ **Successful execution**: allows the iteration to be advanced to the next stage if the synchronization is performed. ■ **Closed**: allows the iteration to be advanced to the next stage if the ticket is closed.

Note You can select multiple ticket states that allow the iteration to be advanced to the next stage based on your requirements.

- To delete a ticket rule, click **Delete** in the Actions column.

iii. Click the **Stage to advance the approval process** tab and configure rules for approval processes.

Rule name	Description	Operation
Non-compliance status check	The action to take if the status of the ticket does not meet the requirements of the iteration template.	Click the value in the Parameter column that corresponds to the **Non-compliance status check** rule. Select an action to take from the drop-down list. Valid values: ■ **No propulsion allowed**: does not allow the iteration to be advanced to the next stage. ■ **Can be promoted and requires approval**: allows the iteration to be advanced to the next stage after approval.
Compliance check	The action to take if the status of the ticket meets the requirements of the iteration template.	Click the value in the Parameter column that corresponds to the **Compliance check** rule. Select an action to take from the drop-down list. Valid values: ■ **Can be promoted without approval**: allows the iteration to be advanced to the next stage without approval. ■ **Can be promoted and requires approval**: allows the iteration to be advanced to the next stage after approval.

6.Data factory

6.1. Task orchestration (new)

6.1.1. Orchestrate tasks

Data Management (DMS) provides the task orchestration feature. You can use this feature to orchestrate different types of tasks, and then schedule and run the tasks. You can create a task flow that consists of one or more task nodes. This allows you to schedule tasks in complex scenarios and improves efficiency of data development.

Prerequisites

The following types of databases are supported:

- MySQL: ApsaraDB RDS for MySQL, PolarDB-X, AnalyticDB for MySQL V3.0, and MySQL databases from other sources
- SQL Server: ApsaraDB RDS for SQL Server and SQL Server databases from other sources
- PostgreSQL: ApsaraDB RDS for PostgreSQL, AnalyticDB for PostgreSQL, and PostgreSQL databases from other sources
- Oracle
- PolarDB for Oracle
- DamengDB
- OceanBase

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Task Orchestration**.
3. Create a task flow.
   i. Click **Create Task Flow**.
   ii. In the **Create Task Flow** dialog box, enter a name and a description for the task flow.
   iii. Click **OK**.
4. In the **Task Type** list on the left side of the canvas, drag task nodes to the blank area on the canvas.
5. Click the task node on the canvas.
6. Configure the task nodes.

   Configure the task node. In the following example, a Single Instance SQL node is configured.

Parameter	Description

Parameter	Description
Database	i. Click the **Node Information** tab. ii. Select a database that you want to manage from the drop-down list below **Node Information**. **Note** You can view the schemas of tables in the database on the **Metadata** tab. iii. Enter the SQL statements to be executed in the SQL editor. iv. Click **Save**.
Variable Setting	○ Click the **Variable Setting** tab, click **Node Variable**, and then configure node variables. For more information, see Configure a time variable. ○ Click **Task Flow Variable** and configure task flow variables. For more information, see Configure a time variable.
Task rerun config	Click the **Advanced Settings** tab. You can turn off or turn on **Enable re-run of failed task**.

7. (Optional)Connect the node to its upstream and downstream nodes.

   Move the pointer over the upstream node, click the hollow dot on the right side of the upstream node, and then drag the connection line to the node that you are configuring.

   In the following example, a **Table status check** node is connected to the **Single Instance SQL** node. Move the pointer over the **Table status check** node, click the hollow dot on the right side of the **Table status check** node, and then drag the connection line to the **Single Instance SQL** node.

   Note In the task flow shown in the preceding figure, the Table status check node is executed before the Single Instance SQL node.

8. (Optional)Click the blank area of the canvas to configure the task flow.
   i. Configure basic properties. Click the **Task Flow Information** tab. In the **Properties** section, modify the task flow name, owner, and stakeholders, turn on or turn off the **Enable message notification** switch, and select an error handling policy and concurrency control policy.
   ii. In the **Scheduling Settings** section, turn on **Enable Scheduling** to configure the scheduling cycle for the task flow.

   ## Set the parameters as required. The following table describes the parameters.

Parameter	Description

Parameter	Description
Scheduling Type	The scheduling type of the task flow. Valid values: ■ **Cyclic scheduling**: The task flow is periodically scheduled. For example, the task flow is run once a week. ■ **Schedule once**: The task flow is run once at a specific point in time. You need to specify only the point in time when the task flow is run.
Effective Time	The period during which the scheduling properties take effect. The default time period is from January 1, 1970 to January 1, 9999, which indicates that the scheduling properties permanently take effect.
Scheduling Cycle	The scheduling cycle of the task flow. Valid values: ■ **Hour**: The task flow is run within the hours that you specify. If you select this value, you must set the Timed Scheduling parameter based on your business requirements. ■ **Day**: The task flow is run at the specified point in time every day. If you select this value, you must set the Specific Point in Time parameter. ■ **Week**: The task flow is run at the specified point in time on the days that you select every week. If you select this value, you must set the Specified Time and Specific Point in Time parameters. ■ **Month**: The task flow is run at the specified point in time on the days that you select every month. If you select this value, you must set the Specified Time and Specific Point in Time parameters.
Timed Scheduling	The method for scheduling the task flow to run. DMS provides the following scheduling methods: ■ Scheduling at a specific interval: ■ **Starting Time**: the beginning of the time range within which DMS runs the task flow. ■ **Intervals**: the interval at which DMS runs the task flow within the specified time range. Unit: hours. ■ **End Time**: the end of the time range within which DMS runs the task flow. For example, if you set the Starting Time parameter to 00:00, the Intervals parameter to 6, and the End Time parameter to 20:59, DMS runs the task flow at 00:00, 06:00, 12:00, and 18:00. ■ Scheduling at the specified point in time: You must set the **Specified Time** parameter. For example, if you select 0Hour and 5Hour, DMS runs the task flow at 00:00 and 05:00.
Specified Time	■ If you set the Scheduling Cycle parameter to Week, you can select one or more days of a week from the drop-down list. ■ If you set the Scheduling Cycle parameter to Month, you can select one or more days of a month from the drop-down list.

Parameter	Description
Specific Point in Time	Specifies the point in time of the specified days at which the task flow is run. For example, if you set this parameter to 02:55, DMS runs the task flow at 02:55 on the specified days.
Cron Expression	The CRON expression is automatically generated based on the specified scheduling cycle and time settings.

9. Publish the task flow. For more information, see Publish task flows.

6.1.2. Batch processing

The batch processing feature of Data Management (DMS) provides a low-code tool that you can use to develop data processing tasks. You can combine a variety of task nodes to create a data flow and configure periodic scheduling to process or synchronize data.

Supported database types

- MySQL: ApsaraDB RDS for MySQL, PolarDB-X, AnalyticDB for MySQL V3.0, and MySQL databases from other sources
- SQL Server: ApsaraDB RDS for SQL Server and SQL Server databases from other sources
- PostgreSQL: ApsaraDB RDS for PostgreSQL, AnalyticDB for PostgreSQL, and PostgreSQL databases from other sources
- Oracle

Scenarios

The batch processing feature supports the batch processing of data. You can use the feature in the following scenarios:

- You can construct an offline data warehouse by using this low-code tool in a visualized way. Then, you can use this data warehouse to perform ad hoc query, data analysis from multiple dimensions, data mining, and offline computing.
- You can process a large amount of complex big data in scenarios such as refined enterprise operations, digital marketing, and intelligent recommendation.
- You can use the batch processing feature that is developed based on Spark SQL to significantly improve the efficiency of Spark SQL nodes on a Hadoop-based platform.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Data processing**.
3. Click the **Batch Processing** tab.
4. Click **Create Data Flow**.
5. In the **Create Data Flow** dialog box, set the **Processing Method**, **Data Flow Name**, and **Description** parameters, and click **OK**.

6. On the details page of the data flow, create nodes for the data flow. For more information, see Create a data flow.
7. On the details page, click the blank area on the canvas to configure the data flow.
   i. Click the **Data Flow Information** tab. In the **Properties** section, set parameters such as **Data Flow Name**, **Owner**, and **Stakeholders**.
   ii. In the **Scheduling Settings** section, turn on **Enable Scheduling** to schedule the data flow based on your needs.
   iii. Click the **Advanced Settings** tab and configure variables. For more information, see Configure a time variable.
8. Publish the data flow. For more information, see Publish task flows.

6.1.3. Configure a data flow

Data Management (DMS) provides the batch processing feature that allows you to combine various task nodes to form a data flow and configure periodic scheduling to process or synchronize data. This topic describes how to configure a data flow.

Limits

The following types of databases are supported:

- MySQL: ApsaraDB RDS for MySQL, PolarDB-X, AnalyticDB for MySQL V3.0, and MySQL databases from other sources
- SQL Server: ApsaraDB RDS for SQL Server and SQL Server databases from other sources
- PostgreSQL: ApsaraDB RDS for PostgreSQL, AnalyticDB for PostgreSQL, and PostgreSQL databases from other sources
- Oracle

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Data processing**.
3. Click the **Batch Processing** tab.
4. Click the name of the data flow that you want to configure to go to the details page of the data flow.
5. Configure a Data Import node.

   > **Note** The first node of the data flow must be a **Data Import** node, which specifies the source table from which the data flow reads data.

   i. In the **Data Processing Type** list on the left side of the canvas, drag the **Data Import** node to the blank area on the canvas.

ii. Click the **Data Import** node that you created on the canvas. On the **Source of data** tab in the lower part, configure the data source.

Parameter	Description
Database Type	The type of the database from which the data flow reads data.
Database	a. The name of the source database. Enter a keyword to search for databases and select the source database from the drop-down list. b. If you have not logged on to the selected database, the **Login Instance** dialog box appears. In the dialog box, set the **Database Account** and **Database password** parameters.
Table	The name of the table. Select the table from which the data flow reads data.

6. Configure a data processing node. In this example, a Data Filtering node is configured to filter data in the data source.

> **Note** All types of nodes other than **Data Import** and **Data Output** can be configured as data processing nodes.

i. In the **Task Type** list on the left side of the canvas, drag the **Data Filtering** node to the blank area on the canvas.

ii. Move the pointer over the **Data Import** node, click the hollow circle on the left side of the Data Import node, and then drag the connection line to the **Data Filtering** node.

iii. Select the **Data Filtering** node that you created on the canvas. On the **Data Filtering** tab in the lower part, specify filter conditions for the data source.

For example, you can enter `name='Jack'` in the field as a filter condition.

> **Note** You can also double-click a function on the right side of the Data Filtering tab to specify filter conditions.

7. Configure a Data Output node.

> **Note** The last node of the data flow must be a **Data Output** node, which specifies the destination table to which the processed data is written.

i. In the **Task Type** list on the left side of the canvas, drag the **Data Output** node to the blank area on the canvas.

ii. Click the **Data Output** node that you created on the canvas. On the **Data Output** tab in the lower part, configure the data output.

Parameter	Description
Database Type	The type of the database in which the destination table resides.
Database	The name of the destination database. Enter a keyword to search for databases and select the destination database from the drop-down list. **Note** If you have not logged on to the selected database, enter the database account and password in the **Login Instance** dialog box that appears.
Table name	The destination table to which the data flow writes the processed data. Enter the name of an existing table or a new table.
SQL Statements Executed Before Writing	The SQL statements to be executed before the data is written.
SQL Statements Executed After Writing	The SQL statements to be executed after the data is written.
Automatic Table Creation	Specifies whether to automatically create a table as the destination table if the specified destination table does not exist. You can turn on or off **Automatic Table Creation**. ■ Turned off: does not automatically create a table as the destination table. In this case, the data flow fails to run. ■ Turned on: automatically creates a table as the destination table. In this case, the data flow continues to run.

iii. Move the pointer over the **Data Filtering** node, click the hollow circle on the left side of the Data Import node, and then drag the connection line to the **Data Output** node.

Then, the icon on the right side of the nodes automatically disappears, which means the dependencies of the nodes in the data flow are all configured.

6.1.4. Configure variables

This topic describes the variables that are used in the task orchestration feature and how to configure time variables.

Overview

When you configure a task node in a task flow, you can configure some or all of the following types of variables on the **Variable Setting** tab:

- Node variables are time variables that can be used only on the current node.
- Task flow variables are time variables that can be used on all nodes of the current task flow.

  > **Note** You can configure task flow variables on one of the task nodes in a task flow. Then, the configuration of task flow variables is synchronized to other task nodes in the task flow.

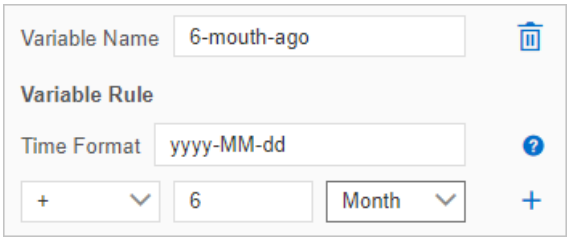
- Input variables are automatically obtained by Data Management (DMS). You can use ${var_name} to reference an input variable in SQL statements for the current node. You can also use an input variable as a filter condition for a conditional branch node. The following variables are considered as input variables:
  - Upstream variables: the output variables from upstream nodes.
  - Status variables: For more information, see the Status variables section of this topic.
  - System variables: For more information, see the System variables section of this topic.
- Output variables: the variables that are defined and have their values assigned on the current node. Output variables can be accessed and referenced by the downstream nodes of the current node.

  For example, if you configure an output variable on a Script node, the output variable can be referenced in the SQL statements of a downstream node.
- Script output variables: If the last line of the script file is a JSON string in the format of `{ key1: value1, key2: value2, … }` for a Script node, the script task parses the key-value pairs in the JSON string to obtain output variables. For each variable, the name is *key* and value is the value of the key parameter. You can reference a script output variable in the format of `${key}` in the SQL statements of a downstream node.

  For example, if the last line of the script file is `echo {"hello": "world"}`, the script task parses the key-value pair to obtain an output variable whose name is *hello* and value is world.

Configure time variables

Parameter	Description
Variable Name	The name of the custom time variable. > **Note** To delete a configured variable, click the icon.

Parameter	Description
Variable Rule	The time format and time offset configurations of the custom time variable. • **Time Format**: the required time format of the time variable. For more information about time formats, see the Time formats section of this topic. • Time offset: A time variable is defined based on the value of the bizdate variable that indicates one day before the current date. For example, you have created a variable named `6_month_ago` in the yyyy-MM-dd format and set the offset to "- 6 Month". In this case, if the current date is August 12, 2021, the value of the `${6_month_ago}` variable is 2021-02-11, which indicates February 11, 2021.  **Note** After you configure a time variable, you can reference the variable in the `${Variable name}` format in the SQL statement that you enter in the SQL editor to the right of the Variable Setting tab. You can also click **SQLPreview** to view the value of the time variable.

Time formats

The following table describes the time formats that variables support.

Time variable	Description	Sample format	Sample value
Anno Domini (AD)	G indicates AD.	Gyyyy	AD 2021
Year	• y or yyyy: the year of the current day. • yy: the last two digits of the year. • Y: the year of the last day in the current week. The last day of the week is Sunday.	yyyy	2021
Month	M: the month of the current year. Valid values of M: [1,12]. Valid values of MM: [01,12].，MMM将返回一月至十二月	MM	08
Week	• w: the week of the current year. Valid values of w: [1,52]. Valid values of ww: [01,52]. • W: the week of the current month. Valid values: [1,5].	ww	13

Time variable	Description	Sample format	Sample value
Day	• D: the day of the current year. Valid values of D: [1,365]. Valid values of DD: [01,365]. Valid values of DDD: [001,365]. • d: the day of the current month. Valid values of d: [1,31]. Valid values of dd: [01,31].	D	360
Day of the week	• E: the day of the week. Valid values: Monday to Sunday. • e: the number that indicates the day of the week. Valid values: [1,7]. A value of 1 indicates Monday. Note To count Sunday as the first day of the week, you can set the offset to "+ 1 Day".	e	1
Ante meridiem (AM) or post meridiem (PM)	a: indicates whether the point in time is before or after the midday. If the point in time is in the range of 00:00 to 11:59, the return value is AM. If the point in time is in the range of 12:00 to 23:59, the return value is PM.	a	AM
Hours	• H: the hour of the current day. A value of 0 indicates the first hour of the day. Valid values of H: [0,23]. Valid values of HH: [00,23]. • h: the hour of the half day. A value of 1 indicates the first hour of the half day. Valid values of h: [1,12]. Valid values of hh: [01,12]. • K: the hour of the half day. A value of 0 indicates the first hour of the half day. Valid values of K: [0,11]. Valid values of KK: [00,11]. • k: the hour of the current day. A value of 1 indicates the first hour of the day. Valid values of k: [1,24]. Valid values of kk: [01,24].	HH	10
Minute	m: the minute of the hour. Valid values of m: [0,59]. Valid values of mm: [00,59].	m	27
Seconds	• s: the second of the minute. • S: the millisecond of the minute.	ss	08
Time zone	z: the time zone.	z	UTC+08:00

The following table describes the sample time formats that use multiple time variables.

Sample format	Sample value

Sample format	Sample value
`yyyy-MM-dd`	2021-08-12
`yyyyMM01`	20210801
`HH:mm:ss`	11:05:21
`yyyyMMdd HH:mm:ss`	20210812 11:05:21

Status variables

Status variable	Description
all_success	All of the tasks are run.
all_failed	All of the tasks fail to be run.
one_success	A task of the current task flow is run.
one_failed	A task of the current task flow fails to be run.

Note You can use status variables on a conditional branch node to make a conditional evaluation in a task flow. The subsequent tasks in the task flow can be run only if the conditional branch node meets the specified conditions.

System variables

System variable	Description	Sample value
sys.flow.start.timestamp	The timestamp generated when the task is run.	2021-05-24T11:20:07.562+08:00
sys.flow.start.year	The year when the task is run.	2021
sys.flow.start.month	The month of the year when the task is run.	5
sys.flow.start.day	The day of the month when the task is run.	24
sys.flow.start.hour	The hour of the day when the task is run.	11
sys.flow.start.minute	The minute of the hour when the task is run.	20
sys.flow.start.second	The second of the minute when the task is run.	7

System variable	Description	Sample value
sys.flow.start.milliseconds	The millisecond of the second when the task is run.	562
sys.flow.start.timezone	The time zone.	Asia/Shanghai
sys.flow.biztime	The data timestamp. By default, the data timestamp is one day before the day when the task is run.	1621740007562
sys.flow.name	The name of the task flow.	dwd_activityDailyPV
sys.node.name	The name of the node in a task flow.	Single Instance SQL-1

6.1.5. Publish a task flow

After you configure or modify a task flow, you must publish the latest task flow. This prevents the modified task flow from being published before the modifications to the task flow are confirmed.

1. Log on to the DMS console.
2. Go to the Task Orchestration page or Batch Processing tab.
   - To go to the Task Orchestration page, click **DTS** in the top navigation bar. Then, click **Task Orchestration** in the left-side navigation pane.
   - To go to the Batch Processing tab, click **DTS** in the top navigation bar. In the left-side navigation pane, click **Data processing**. On the Data processing page, click the **Batch Processing** tab.
3. Click the name of a task flow that you want to manage and go to the details page of the task flow.
4. (Optional)Test the running of the task flow.
   i. Click **Try Run** in the upper-left corner of the canvas.
   ii. In the **Alert** message, click **OK**.
   iii. In the lower part of the canvas, click the **Execution Logs** tab and check whether the task flow is run.
      - If `status SUCCEEDED` appears in the last line of the logs, the task flow is run.
      - If `status FAILED` appears in the last line of the logs, the task flow fails to be run.

      > **Note** If the task flow fails to be run, view the node on which the failure occurs and the reason for the failure in the logs. Then, modify the configuration of the node and try again.
5. Publish the task flow.
   i. Click **Publish** in the upper-left corner of the canvas.
   ii. In the **Publish** dialog box, enter text in the **Remarks** field and click **OK**.
6. (Optional)View the status of the task flow.
   i. In the upper-right corner of the canvas, click **Go to O&M**.

ii. On the right side of the page, check whether the task flow is published.

- **Published**: The task flow is published.
- **Not published**: The task flow is not published.

6.1.6. Create an ETL task flow

The stream processing feature allows you to perform extract, transform, and load (ETL) processing on streaming data. This way, you can accurately and efficiently obtain the data that you need. This topic describes how to create an ETL task flow.

Prerequisites

- A source MySQL database is created.
- A destination MySQL or AnalyticDB for MySQL 3.0 database is created.
- The source and destination databases are in the same region.

Background information

An ETL task flow allows you to add transformation components between the source and destination databases. This way, you can transform data in a variety of ways and write the processed data to the destination database in real time.

For example, you can add a field to the source table and configure a function to assign values to the field. Then, you can write the field to the destination database.

Key nodes

- **Input / Dimension Table**: the source database of the ETL task flow.
- **Output**: the destination database of the ETL task flow.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Data processing**.
3. Click the **Stream Processing** tab.
4. Create a data flow.
   i. On the Stream Processing tab, click **Create Data Flow**.
   ii. In the Create Data Flow dialog box, enter a name for the data flow.
   iii. Select **FlinkSQL** as Development Method.
   iv. Enter a description for the data flow.
   v. Click **OK**.
   vi. In the Create Task panel, set the Source Database Type, Source Region, Destination Database Type, Destination Region, and Purchase Quantity parameters.
   vii. Click **Purchase**.
5. In the data flow list, find the data flow you want to configure and click the name of the data flow.
6. On the details page of the data flow, configure the data flow.

Task	Description
Configure source databases	i. In the **Data Processing Type** list on the left side of the canvas, drag the **Input/Dimension Table** node to the blank area on the canvas. ii. Click the Input/Dimension Table node and configure the node information and output fields. **Note** You can select one or more source databases. You can select the same type of source database multiple times.
Configure transformation components	i. In the **Data Processing Type** list on the left side of the canvas, drag the transformation component node to the blank area on the canvas. ii. Move the pointer over the **Input/Dimension Table** node, click the hollow circle on the right side of the **Input/Dimension Table**, and then drag the connection line to the transformation component node. iii. Click the transformation component node and configure the node in the lower part of the page. **Note** You can select one or more transformation components. You can select the same transformation component multiple times.
Configure the destination database	i. In the **Data Processing Type** list on the left side of the canvas, drag the **Output** node to the blank area on the canvas. ii. Drag the connection line from the transformation component node to the **Output** node. iii. Click the Output node and configure the node information and field mapping information. **Note** You can select only one destination database.

7. Publish the data flow.
   i. In the upper-right corner of the page, click **Generate Flink SQL Validation**.
   ii. In the upper-right corner of the data flow configuration tab, click **View ETL Validation Details** to view the generated Flink SQL statements.

   Note If the validation fails, you can fix the error as prompted. After the error is fixed, click Generate Flink SQL validation again to generate Flink SQL statements.

iii. You can perform one of the following operations:

- Publish: In the upper-right corner of the data flow configuration tab, click **Publish** to publish the data flow.
- Publish and perform a precheck: In the upper-right corner of the data flow configuration tab, choose **Publish > Publish and perform precheck** . After the data flow is published, the system automatically performs a precheck.

> **Note** If the precheck fails, you can modify the data flow as prompted. After the data flow is modified, you can publish the data flow and perform a precheck again.

6.2. Data migration, synchronization, and change tracking

This topic describes the data migration, synchronization, and change tracking features of DTS integrated with DMS.

Features

DMS integrates with the data migration, synchronization, and change tracking features of DTS.

- Data migration

  You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature is suitable for the following scenarios: data migration to Alibaba Cloud, data migration between instances in Alibaba Cloud, and database splitting and scale-out.

- Data synchronization

  You can use DTS to synchronize data between data sources in real time. This feature is suitable for the following scenarios: active geo-redundancy, geo-disaster recovery, zone-disaster recovery, cross-border data synchronization, cloud-based business intelligence (BI) systems, and real-time data warehousing.

- Change tracking

  You can use DTS to track incremental data from ApsaraDB RDS for MySQL instances, PolarDB for MySQL instances, PolarDB-X instances, self-managed MySQL databases, and self-managed Oracle databases in real time. Then, you can consume the tracked data as needed. This feature is suitable for the following scenarios: cache updates, business decoupling and asynchronous data processing, real-time data synchronization between heterogeneous databases, and real-time data synchronization that involves extract, transform, load (ETL) operations.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**.
3. Create a data migration, data synchronization, and change tracking task.
   - Data migration
     a. In the left-side navigation pane, click **Data Migration**.
     b. On the Data Migration page of the DTS console, create a data migration task. For more information, see *Data migration* in the *DTS documentation*.

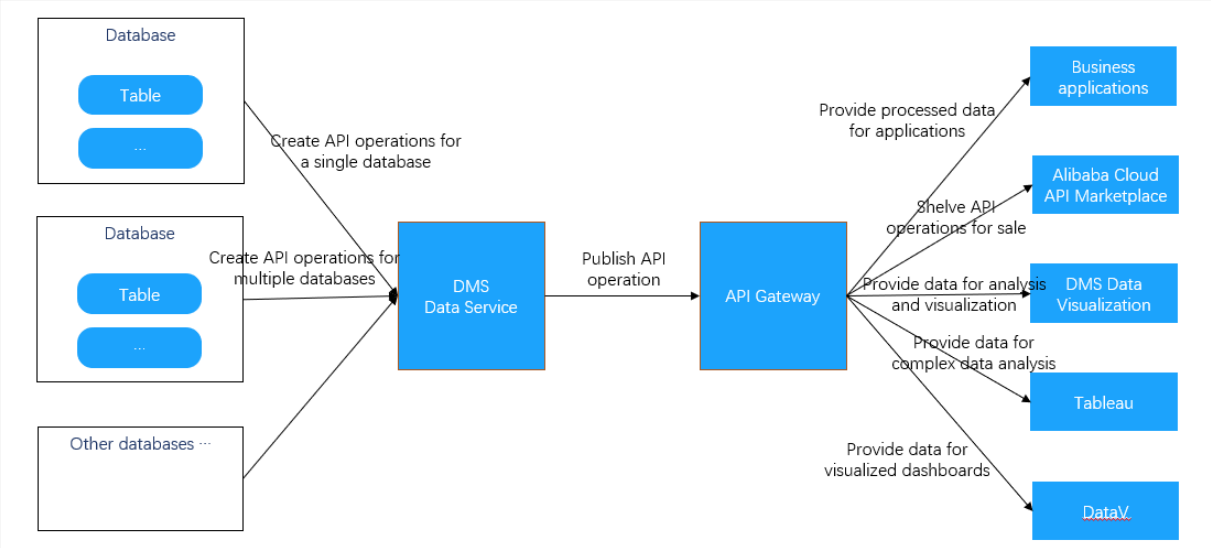
- Data synchronization
  a. In the left-side navigation pane, click **Data Synchronization**.
  b. On the Data Synchronization page of the DTS console, create a data synchronization task. For more information, see *Data synchronization* in the *DTS documentation*.
- Change tracking
  a. In the left-side navigation pane, click **Data Subscription**.
  b. On the Change Tracking page of the DTS console, create a change tracking task. For more information, see *Change tracking* in the *DTS documentation*.

6.3. Data service

6.3.1. Overview

Data Management (DMS) provides the data service feature, which allows you to export the data that is managed by DMS. This feature is applicable to scenarios where you need to export data at the column or row level, display data in a visualized manner, or perform complex analysis.

Features



- You can use the data service feature to create APIs that can be called to access the data that is managed by DMS. When you create the APIs, you can apply the security control features that are used for SQL execution in the SQLConsole, such as permission control and data de-identification.
- The data service feature works based on a serverless architecture. This feature frees you from the concern about the infrastructure of the runtime environment, such as servers and networks. You need to focus only on how to create APIs and design data query logic. This avoids operations and maintenance (O&M) overheads that are generated by using traditional architectures.
- The data service feature is fully integrated with API Gateway. You can use this feature to publish APIs to API Gateway. This way, you can use all the features that are provided by API Gateway, such as API permission control, IP address-based access control, throttling, metering and billing, and SDKs.

Scenarios

Scenario	Description
Minimize data exposure	Assume that you need to export the data that is managed by DMS to an external environment. In this case, APIs can be called to export the data of specific rows or columns to the external environment. To export the data of specific rows, specify a filter condition in the SQL statement. To export the data of specific columns, specify the columns in the SQL statement. Compared with data export of a whole table, this minimizes data exposure and ensures data security.
Connect visualization tools to databases	Most visualization tools can connect to databases by calling APIs. You can connect a visualization tool to your database by calling an API, instead of by using a username and a password. This method is easy to implement and avoids account exposure.
Sell APIs in the Alibaba Cloud Marketplace	If you want to provide paid or free data for other users, publish an API to the Alibaba Cloud Marketplace.
Provide processed data for applications	After data is processed and summarized by using the data warehouse development feature of DMS, APIs can be created and provided for applications to read the processed data from DMS to meet business needs. To modify the logic of data reading, you need only to modify the query logic of the required API without the need to republish the application.

6.3.2. Develop an API

The data service feature of Data Management (DMS) allows you to develop APIs with ease. This topic describes how to create and manage APIs.

Context

The data service feature allows you to export the data that is managed by DMS. This feature is applicable to various scenarios, such as data export at the column or row level, data visualization, or complex data analysis. For more information, see Overview.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, choose Data Application > **Data Service**.
3. In the left-side navigation pane, click the **API Development** tab.
4. On the APIManagement tab, click **New API** in the upper-right corner.
5. Set the parameters for the new API.

i. Set the parameters on the AttributeConfiguration tab.

Parameter	Description
APIName	The name of the API. ■ The name can contain letters, digits, and underscores (_). ■ The name must start with a letter. ■ The name must be 4 to 100 characters in length.
Description	Optional. The description of the API. Enter an informative description, for example, a description of the data that you want the API to return or the scenarios in which the API can be called.
Path	The path of the API. The path must start with a forward slash (/) and can contain letters, digits, underscores (-), and hyphens (-). The specified path forms a part of the URL that is used to call the API. A URL that is used to call an API must be in the `https://{Domain name}{Path}` format. For example, if the domain name is `xxxx-cn-hangzhou.alicloudapi.com` and the path is `/item/monthly_data`, the URL that is used to call the API is `https://xxxx-cn-hangzhou.alicloudapi.com/item/monthly_data`.
ReturnFormat	The format in which you want the API to return data. Valid value: **JSON**.
RequestMode	The request method. Valid values: **POST** and **GET**.
TimeOut (MS)	The maximum period of time that the system can wait until an API request expires. Unit: milliseconds. If the execution time of an API exceeds the specified timeout period, the system returns a timeout error. Maximum value: 30000, which indicates 30 seconds.
Returns the maximum number of records	The maximum number of entries that can be returned for an API request. This parameter limits the number of entries that can be returned for each API request. **Note** If the database instance is managed in Security Collaboration mode, the value of this parameter must be smaller than the maximum number of entries that is specified in the security rules.
Return Field Metadata	Specifies whether to return the field metadata.
Return Execution Details	Specifies whether to return the execution details.

ii. Click the **ExecuteConfiguration** tab and set the parameters.

Parameter	Description
Instance query type	■ **Single InstanceQuery**: You can call the API to read data from only one database instance. ■ **Cross-instanceQuery**: You can write dynamic SQL statements for the API to query data across multiple database instances. **Note** If you select **Cross-instanceQuery**, you need to only enter dynamic SQL statements in the **QuerySQL** field.
Data source	The database that is queried by the API. You can search for databases on which you have query permissions by keyword and then select a database.
ConfigurationMode	■ **Table boot mode**: Configure a data query by selecting a table and fields. ■ **Script mode**: Configure a data query by specifying variables and writing SQL statements. **Note** If you select **Script mode**, you need to only enter SQL statements in the **QuerySQL** field.
SelectTable	The table to be queried. You can search for tables by keyword.
FieldList	The fields in the selected table. You can specify the required fields as request parameters or response parameters.
Script mode	The mode in which an SQL script is written to define the data query logic.
QuerySQL	The SQL statement that is used to query the data in the table. After you enter an SQL statement, click **ParsingScript** to verify the syntax and to parse the request parameters and response parameters. **Note** ■ Custom variables are supported. Custom variables can be mapped as request parameters in API requests. The variables in the SQL statement must be defined in the `${Variable name}` format. For example, you can define the `${category}` variable and use it in the following SQL statement: `select item_id, item_name from ex_item where category =${category}`. ■ If you select **Cross-instanceQuery** as **Instance query type**, you must use the syntax of cross-database query SQL statements. For more information, see Cross-database query.

iii. Click the **Request Parameters** tab and set the parameters.

Parameter	Description
ParametersName	The name of the request parameter. ■ The name can contain letters, digits, hyphens (-), and underscores (_). ■ The name must start with a letter or an underscore (_). ■ The name must be 1 to 50 characters in length.
VariableName	The name of the field that is specified by the request parameter. The field name is specified on the **ExecuteConfiguration** tab and cannot be changed.
Cannot be empty	Specifies whether the request parameter is required.
Description	The description of the request parameter.
Data type	The data type of the request parameter. The data type is used to check whether the value of the request parameter in an API request is valid. Valid values: String, Integer, and Floating point. Default value: String. **Note** This parameter affects the SQL statement that is executed when the API is called.
Example value	The sample value of the request parameter. You can use the sample values that are provided in SDKs and documentation as references when you call API operations.
Default value	The default value of the request parameter. If the request parameter is optional and not specified in the API request, the default value is used.

iv. Click the **Return parameter** tab and set the parameters.

Parameter	Description
ParametersName	The name of the response parameter. ■ The name can contain letters, digits, hyphens (-), and underscores (_). ■ The name must start with a letter or an underscore (_). ■ The name must be 1 to 50 characters in length.
VariableName	The name of the field that is returned. The name cannot be changed.
Description	The description of the response parameter.
Data type	The data type of the response parameter. Valid values: String, Integer, and Floating point. Default value: String. This parameter is used by DMS to convert the type of the data in API responses. This parameter affects the JSON data that is returned.
Example value	The sample value of the response parameter. You can use the sample values that are provided in SDKs and documentation as references to help you understand API responses.

6. Click **Save**.
7. In the left-side navigation pane, click the **API Development** tab.
8. Perform the following operations to manage the API based on your business requirements:
   - Publish the API

     On the APIManagement tab, find the API and click **Publish** in the **Operation** column of the API. In the message that appears, click **OK**.
   - Modify the API

     On the APIManagement tab, find the API and click **Modify** in the **Operation** column of the API. Modify the configurations of the API based on the descriptions in Step 5 and click **Save**.
   - Delete the API

     On the APIManagement tab, find the API and click **Delete** in the **Operation** column of the API. In the message that appears, click **OK**.

6.3.3. Unpublish or test an API

This topic describes how to unpublish or test an API that has been published.

Prerequisites

An API is created. For more information, see Develop an API.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, choose Data Application > **Data Service**.

3. Click the **API Publish** tab on the left side.
   The APIPublishList tab displays all the published APIs.
4. Find the API that you want to manage and perform the following operations based on your business requirements:
   - Unpublish the API:

     Click **Offline** in the **Operation** column. In the message that appears, click **OK**.
   - Test the API:

     Click **Test** in the **Operation** column. For more information, see Test an API.

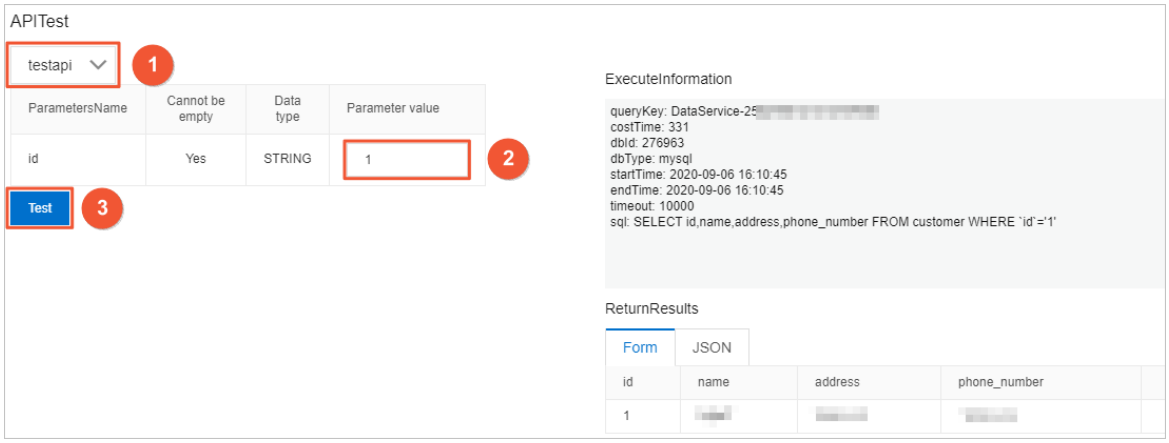
6.3.4. Test an API

After you create an API, you can test the API to verify whether the API meets your business requirements.

Prerequisites

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, choose Data Application > **Data Service**.
3. Click the **API Test** tab on the left side.
4. On the APITest tab, test an API.



   i. Select the API that you want to test from the drop-down list.

   ii. Enter values in the Parameter value column.

   iii. Click **Test**.
   After the test is complete, the execution information and return results appear on the right side. You can evaluate whether the API meets your business requirements based on the information.

   > **Note** You can click the **JSON** tab in the **ReturnResults** section so that the return results are displayed in the JSON format.

6.3.5. Call an API

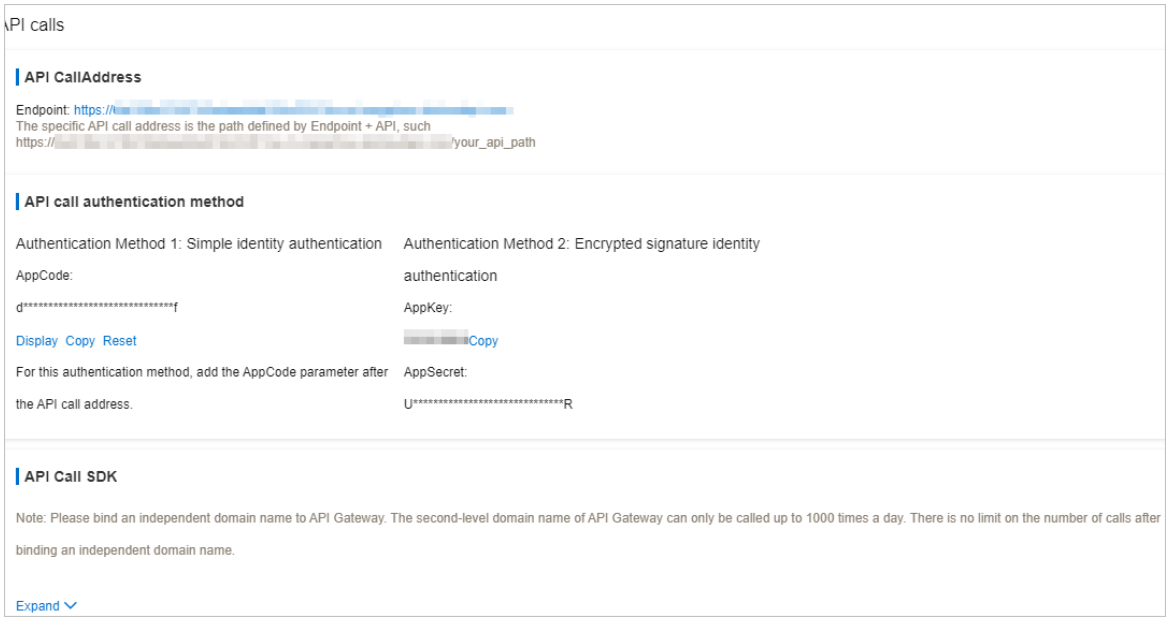
After you create, publish, and test an API, you can call the API in an application by using an SDK.

Prerequisites

- An API is created and published. For more information, see Develop an API.
- API Gateway is activated. For more information, see the documentation of *API Gateway*.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**. In the left-side navigation pane, choose Data Application > **Data Service**.
3. Click the **API Call** tab on the left side.
4. View the API call address and the authentication information.



   - **Simple identity authentication**: requires only an AppCode. This authentication method is suitable for calling APIs by using URLs. This authentication method has a low security level and is generally used in scenarios in which data visualization is involved, such as calling APIs in DataV.
   - **Encrypted signature identity authentication**: requires an AppKey and an AppSecret, which are used to dynamically generate an encrypted signature for calling an API. This authentication method has a high security level.
5. Call the API in an application by using an SDK.

> **Note** For more information about how to call an API in an application by using an SDK, see the documentation of *API Gateway*.

6.4. Data visualization

6.4.1. Overview

This topic introduces the basic concepts, design philosophy, and scenarios of the data visualization feature of Data Management (DMS).

Background information

DMS allows you to manage databases and query data in the SQLConsole where results are returned in the form of a table. However, if you want to analyze business characteristics in scenarios such as trend analysis and growth comparison, tables cannot meet the requirements, and data visualization is required. To resolve this issue, DMS provides the data visualization feature. You can use this feature to gain insights into your business and make better business decisions.

Basic concepts

The data visualization feature provides a three-layer model for you to visualize data in various forms, including datasets, charts, and dashboards or big screens. You can execute SQL statements in the SQLConsole to obtain datasets and convert the datasets to common charts such as line charts, pie charts, column charts, circular charts, table charts, dual Y-axis charts, and funnel charts. Then, on a dashboard or big screen, you can freely combine and lay out these charts based on your analysis logic or methodology to visually present your business data.

> **Note** For example, you can use indicator cards to display the overall metrics of your business, such as a transaction volume and unique visitors (UVs). Then, you can use a line chart to present the growth trend of the transaction volume and a column chart to compare transactions in all regions. Finally, you can use a table chart with a filter to query region-specific data.

Design philosophy

Two core concepts of data visualization are datasets and charts. Datasets are also called data views, and charts are also called visualization components.

- Datasets represent the structured form of data. Data logic, permissions, and services are all based on this form.
- Charts represent the visual form of data. Data presentation, interaction, and guidance are all based on this form.

  > **Note** Datasets and charts complement each other to provide the same data in two different forms and help you better understand data.

- Dashboards or big screens are used for quick data analysis and custom data visualization. You can combine charts on dashboards or big screens as needed. This can satisfy the data visualization needs of most users.

Scenarios

- Analyze data in a secure and custom manner

  The data visualization feature is based on the security control feature in DMS. This ensures that data is authorized before it is visualized.

- You can set the configurations only once to implement the advanced filtering, advanced control, interaction, drilling, download, and sharing of visual components. This facilitates data analysis and decision-making. For example, you can use this feature to compare data and analyze the geographic information of data, data distribution, data trends, and data clusters with ease.
- Dashboards use automatic layouts. They can be used for most visual reports that require simple configuration and need to be viewed and shared with ease.
- Big screens use custom layouts. They can be used for specific visual reports that require additional modifier elements and need to be retained for a long period of time. Time and efforts are required to configure a big screen in these scenarios, such as a big screen for massive online promotions.

- Monitor operations in real time

  In the data factory of DMS, you can synchronize your business data in real time to AnalyticDB or ApsaraDB RDS where data can be analyzed. Then, you can visualize data that is analyzed in AnalyticDB or ApsaraDB RDS. This way, you can monitor database performance in real time and make sure that data flows are seamlessly connected. In addition, you can compare data to detect anomalies and handle key-link issues. Pivot-driven mode and chart-driven mode are provided for chart configuration. You can apply these modes to different scenarios based on your business requirements.

6.4.2. Terms

This topic describes the terms related to the data visualization feature of DMS.

Term	Description
dimension	A dimension represents an attribute of business data, such as time, region, gender, and category. A dimension contains a collection of discrete values, based on which a measure is obtained.
measure	A measure is a statistical value that is obtained from an aggregation operation. For example, unique visitor (UV) and transaction volume are both measures.
dataset	A dataset is a collection of data in the form of a two-dimensional table. The data is generated after an SQL statement is executed to query a database. Therefore, you must prepare a database and an SQL statement to obtain a dataset.
chart	A chart is a graph that visualizes data to present a data feature. For example, a line chart shows a data trend, a table chart displays detailed data, a bar chart compares data, and a pie chart highlights percentages. Different charts may need different numbers of dimensions and measures. For example, a line chart requires one dimension and one or more measures.
dashboard	A dashboard is a visualization tool where multiple charts are combined to present business data in a comprehensive way. Charts can be laid out more flexibly on dashboards than in traditional visual reports. You can divide a dashboard into sections and adjust the size and position of the chart in each section. This can optimize the dashboard layout and provide user-friendly interactions. You can also configure a global filter that allows you to filter data across charts on a dashboard and display data that is queried.

Term	Description
dashboard collection	A dashboard collection is used to manage a group of dashboards that are related to each other. For example, you can create a dashboard collection that is specific to products and put the Products Sold dashboard, the Products Added to Shopping Cart dashboard, and the Products Returned dashboard to this dashboard collection. This provides convenient and unified management. In a dashboard collection, you can create recursive directories to classify the dashboards.
big screen	On a big screen, you can combine multiple charts as you can on a dashboard. You can also use auxiliary graphics, such as images and rectangles, on a big screen. This allows you to create layouts in a more flexible way. Different from dashboards that you can divide into sections, big screens adopt an absolute positioning layout. You can freely drag charts and auxiliary graphics on a big screen. This meets the requirement for more flexible data visualization.

6.4.3. Go to the Data Visualization tab

This topic describes the methods that you can use to go to the Data Visualization tab.

Go to the Data Visualization tab in the Data Management (DMS) console of the new version

1. Log on to the DMS console.
2. In the top navigation bar, click **DTS**.
3. In the left-side navigation pane, click **Data Visualization**.

Go to the Data Visualization tab from the SQLConsole tab

1. Go to the **SQLConsole** tab. For more information, see SQLConsole.
2. In the upper part of the **SQLConsole** tab, click **Data Visualization**.

6.4.4. Manage datasets

A dataset is a collection of data in the form of a two-dimensional table. The data is generated after an SQL statement is executed to query a database. This topic describes how to manage datasets.

Create a dataset

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. On the **Dataset management** tab, click the + icon.
4. In the **Writing SQL** step, set the parameters and click **Execute**.

Parameter	Description
Name	The name of the dataset. The dataset name must be unique for each Data Management (DMS) user.

Parameter	Description
Description	The description of the dataset.
Select an existing database	The database to be queried. You must have permissions to query the database.
Search table/field name	The name of the table or field that you want to search for and select.

5. After the SQL statement is executed, click **Next Step**.
6. In the **Edit dataset model** step, set the **Data type** and **Visualization type** parameters for each field based on your requirements.

   In this step, you must specify each queried field as a dimension or measure and specify a visualization type for each field.

Parameter	Description
Data type	Valid values: ○ **Dimension**: the scope, aspect, or angle of measures. ○ **Measure**: the statistical value that is obtained after an aggregation operation. To show how transaction volume changes over time, you can set the Data type parameter for the time field to **Dimension** and that for the transaction volume field to **Measure**. For more information, see Terms.
Visualization type	Valid values: ○ **Digital** ○ **String** ○ **Date** ○ **Geography: Country** ○ **Geography: Provinces** ○ **Geography: City** To show how transaction volume changes over time, you can set the Visualization type parameter for the time field to **Date** and that for the transaction volume field to **Digital**.

7. Click **Save**. The dataset is created, and you are navigated to the Dataset management tab.

Modify a dataset

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. On the **Dataset management** tab, find the dataset that you want to modify and click the icon in the **Operation** column.

4. In the **Writing SQL** step, configure the SQL query statement and click **Execute**.

   For more information about how to set the other parameters in the **Writing SQL** step, see Create a dataset.

5. After the SQL statement is executed, click **Next Step**.
6. In the **Edit dataset model** step, set the **Data type** and **Visualization type** parameters for each field based on your requirements.

   For more information about the **Data type** and **Visualization type** parameters, see Create a dataset.

7. Click **Save**. The dataset is modified, and you are navigated to the Dataset management tab.

Delete a dataset

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. On the **Dataset management** tab, find the dataset that you want to delete and click the icon in the **Operation** column.
4. In the message that appears, click **OK**.

> **Note** Before a dataset is deleted, DMS checks whether the dataset is referenced by a chart. If the dataset is referenced, you cannot delete the dataset. To delete the dataset, you must go to the chart editing page to cancel the reference or delete the chart that references the dataset.

Grant permissions on a dataset

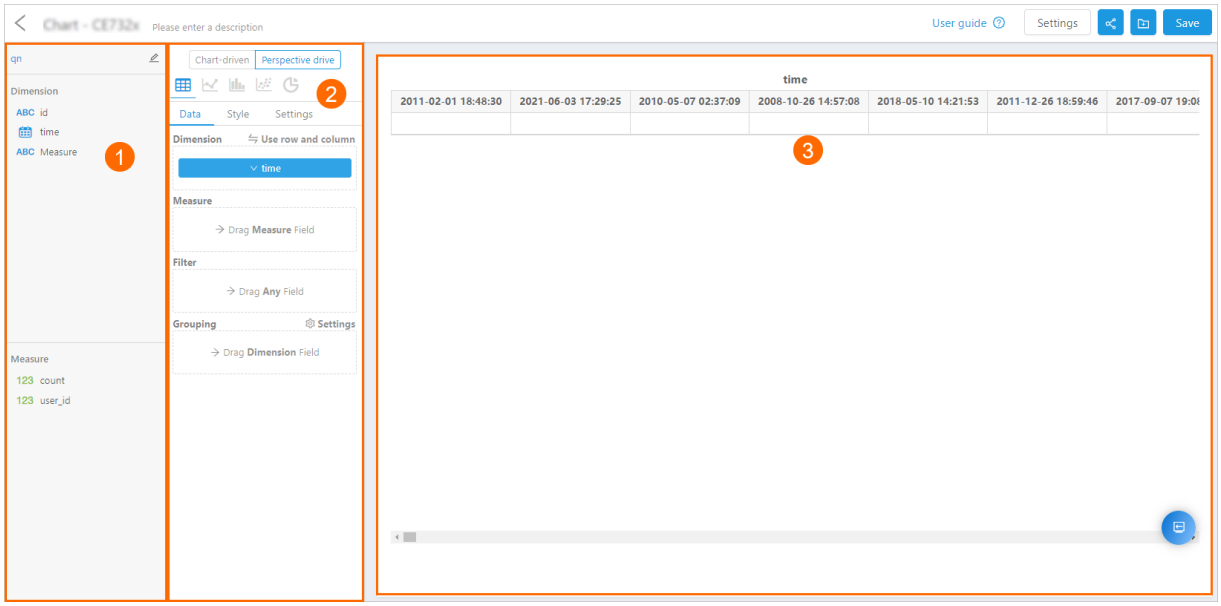
1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. On the **Dataset management** tab, find the dataset on which you want to grant permissions and click the icon in the **Operation** column.
4. In the **Share** dialog box, specify the permissions to be granted on the dataset and the users to whom you want to grant the permissions. Then, click **Authorize**.

6.4.5. Manage charts

A chart is the smallest unit for data visualization. A chart is generated by aggregating and grouping SQL query results based on the dataset model provided by a dataset, and then visualizing the processed data by using code.

Each chart must be associated with a dataset. In the chart editor, the original SQL statement and the dataset model in a dataset are used to generate a new SQL statement. Then, the new SQL statement is executed to query data that is to be visually presented in a chart. This topic describes features of charts and how to use charts.

Chart editor



The following table describes the three sections of the chart editor.

No.	Section	Description
①	Dataset model display section	After you select a dataset in the upper-left corner of this section, its dataset model will be automatically displayed below, including dimensions and measures. You can drag fields in this section to corresponding areas on the **Data** tab in the chart configuration section.
②	Chart configuration section	At the top of this section, you can select a configuration mode and a chart type. Move the pointer over a chart type icon to view the numbers of dimensions and measures required for this chart type. Below the chart type icons are three tabs. • On the Data tab, you can specify which fields in the dataset model are needed for the chart. You can drag fields from the dataset model display section to corresponding areas on this tab. • On the Style tab, you can set the display style of the chart. • On the Settings tab, you can configure functional modules for the chart, such as filters and cache.
③	Chart display section	This section displays the chart based on the dataset model and your configurations in the chart configuration section.

Configuration modes

The chart configuration section provides you with the following two configuration modes that are based on different visual presentation logic: including the **pivot-driven mode** and **chart-driven mode**. This meets different requirements in various scenarios.

Configuration mode	Description	Scenario

Configuration mode	Description	Scenario
Chart-driven mode	The chart-driven mode represents the general visual presentation logic that is based on chart classification. Various charts can be configured by using the chart-driven mode. In this mode, dimensions and measures can be regarded as fixed configuration items, along with style settings, for configuring a chart.	The chart-driven mode is applicable to most data visualization scenarios.
Pivot-driven mode	The pivot-driven mode represents the visual presentation logic that is based on pivot tables. A chart configured by using the pivot-driven mode can be regarded as the visualization of a pivot table through coding. Dimensions and measures in the pivot table are converted to axes in the chart for graphic display. In this mode, you can configure different graphic coding for each measure. The lowest-level dimension can be used as a common dimension axis.	The pivot-driven mode is applicable to scenarios in which a small amount of data needs to be freely analyzed on clients.

Data configuration

To complete data configuration, you must drag fields from the dataset model display section to corresponding areas on the **Data** tab in the chart configuration section. When you are dragging a field, the areas where the field can be placed are highlighted on the Data tab.

Dimension

Only categorical fields can be placed in the Dimension area. Values of each field in this area will be grouped in the new SQL statement to be generated.

Measure

- Only fields of the NUMERIC type can be placed in the Measure area. Values of each field in this area will be aggregated in the new SQL statement to be generated. To specify how a field will be aggregated, you can click the field name and select an aggregate function. The following six aggregate functions are supported:
  - sum
  - avg
  - count
  - count_distinct
  - max
  - min
- You can specify the following data formats for fields in the Measure area:
  - Default: the default format.
  - Numeric value: You can specify a unit, the number of decimal places to which numbers are rounded, and whether to use the thousands separator.
  - Currency: You can specify a unit, the number of decimal places to which numbers are rounded, whether to use the thousands separator. You can also prefix and suffix texts.
  - Percentage: You can specify the number of decimal places to which numbers are rounded.
  - Scientific type: You can specify the number of decimal places to which numbers are rounded.

Filter

Both dimension and measure fields can be placed in the Filter area. Values of fields in this area will be used to specify filter conditions in the SQL statement to be generated. Data can be filtered by values, conditions, and dates.

> **Note** Filter methods applicable to each field in this area correspond to the visualization type that you specified for the field when you configure the dataset model.

Filter method	Description	Visualization type
By conditions	Filtering by conditions is the most flexible filter method. In the Filter configuration dialog box, you can specify filter conditions for the selected field as needed. You can also use the logical AND and logical OR operators to specify multiple conditions.	• Numeric value • String • Geographical value
By values	In the Filter configuration dialog box, a set of distinct values of the selected field are listed in the left-side section. You can select values from this section and add them to the right-side section to filter data.	• String • Geographical value
By dates	You can query the selected field in a specified period of time. The period of time can be fixed or dynamic.	Date

Grouping

You can place only one dimension field in the Grouping area. Values of the field in this area will be grouped in the new SQL statement to be generated. In the chart to be generated, values of the field will be grouped. Each group has a distinct color. The legend of the chart will help you differentiate groups with different colors. If you want measures of a dimension to be displayed as several groups in a chart, you can use color settings to differentiate the groups.

For example, a dataset model contains the trade_date, zone, and price fields, respectively representing the transaction date, transaction region, and transaction volume. You want a bar chart to show aggregated transaction volumes of each region by day. In this case, you can place the trade_date field in the Dimension area, the zone field in the Grouping area, and the price field in the Measure area.

> **Note** When you generate a pie chart, you require a dimension field whose values are to be grouped. In this case, you must place the dimension field in the Grouping area instead of the Dimension area.

General settings for fields

- Field alias

  Click a field name and select **Field settings**. In the **Field settings** dialog box, specify a field alias. You can specify the following two types of field aliases:

  - Permanent alias.

  - Dynamic alias: Dynamic aliases are generated by writing code in JavaScript and can be used with variables. DMS provides the Moment.js library that can be used with variables to dynamically generate aliases for fields of the DATE type. Dynamic aliases can only be generated for fields in table charts.

- Field description

  Click a field name and select **Field settings**. In the **Field settings** dialog box, enter a description. You can add field descriptions only for charts that are configured by using the **chart-driven mode**.

- Field sorting

  Click a field name, select **Sort Type** or Sort, and then select a sorting method. The following table describes the available sorting methods.

Sorting method	Description
Default	Do not sort the field values.
Ascending order	Sort the field values in ascending order in the SQL statement to be generated.
Descending order	Sort the field values in descending order in the SQL statement to be generated.
Customize	You can drag field values to sort them. In the chart to be generated, the field values will be displayed in the specified order. Only categorical fields support this method.

Other configurations

- Size: If you are configuring a scatter chart, you must place a field of the NUMERIC type in the Size area. The field is used to code points in terms of size. Values of the field will be aggregated in the SQL statement to be generated.
- Prompt information: You can set prompts only for certain Cartesian charts. Only fields of the NUMERIC type can be placed in the Prompt information area. Values of each field in this area will be aggregated in the SQL statement to be generated.
- When you configure a chart by using the pivot-driven mode, you can place fields of any type in the Label area. Values of each categorical field in this area will be grouped and values of each field of the NUMERIC type will be aggregated in the SQL statement to be generated.
- When you configure a scatter chart by using the pivot-driven mode, you must place a field of the NUMERIC type in the x data axis area. The field is specified as the measure of the X-axis. Values of the field will be aggregated in the SQL statement to be generated.
- When you configure a dual Y-axis chart by using the chart-driven mode, you can specify a measure for the left Y-axis and a measure for the right Y-axis.

Chart configuration

After you configure fields, you can select a chart type by clicking the corresponding icon at the top of the chart configuration section.

> **Note** You can move the pointer over an icon to check the prerequisites of the chart type. If your data configuration does not meet all the prerequisites, the icon is dimmed. Only after all the prerequisites are met does the icon become highlighted. Click the icon and a chart appears in the chart display section.

On the **Style** tab of the chart configuration section, you can customize the chart display style.

Functionality settings

On the **Settings** tab of the chart configuration section, you can configure a filter and cache and specify whether to automatically load data.

- Filter

  You can create a filter or edit an existing filter by clicking **Settings** next to Filter.

  > **Note** To configure a filter, you must have defined **variables** when you configure the dataset.

  After you add a chart with a filter to a dashboard, you can click the button in the upper-left corner of the dashboard card. In the filter panel that appears, specify filter conditions and then click **Query** in the lower-right corner of the panel. Conditions that you specified in the chart filter and conditions that you specified in the global filter of the dashboard take effect at the same time.

- Cache

  You can enable or disable caching for a chart. You can also specify a validity period for the cached data.

  > **Note** If you enable caching for a chart, the result of the first query for the chart on a dashboard or big screen will be stored in the cache. The SQL query statement will be used as a key. Within the validity period, DMS returns the cached result for the same SQL query statement with no need to access the data source.

- Automatically load data

  In scenarios where data query is frequent, you may not want data to be loaded immediately when you open a dashboard. In this case, you can set **Automatically load data** to **No**. This parameter is set to **Yes** by default.

Create a chart

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Chart**.
4. On the Chart management page, click the + icon.
5. Configure the dataset and the chart.
6. In the upper-right corner, click **Save**.

Modify a chart

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Chart**.
4. On the **Chart management** page, find the chart that you want to modify and click the icon in the **Operation** column.
5. After the configurations of the dataset and the chart are modified, click **Save**.

Delete a chart

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Chart**.
4. On the **Chart management** page, find the chart that you want to delete and click the icon in the **Operation** column.
5. In the message that appears, click **OK**.

Duplicate a chart

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Chart**.
4. On the **Chart management** page, find the chart that you want to duplicate and click the icon in the **Operation** column.
5. In the **Copy chart** dialog box, set the **Chart name** and **Description** parameters.
6. Click **OK**.

Grant permissions on a chart

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Chart**.
4. On the **Chart management** page, find the chart on which the permissions you want to grant and click the icon in the **Operations** column.
5. In the dialog box that appears, select the user to which you want to grant permissions and the permissions to be granted.
6. Click the button in the dialog box.

6.4.6. Manage dashboards

Dashboards are the other type of visualization applications that are provided by DMS. Dashboards support automatic layout and provide interactive capabilities to help you create visual reports. This topic describes features of dashboards and how to use dashboards.

Create a dashboard collection

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Dashboard** section of the **Resource management** page, click **New Dashboard collection**.
5. In the **New Dashboard collection** dialog box, set the **Name** and **Description** parameters.
6. Click **Save**.

Add a chart to a dashboard

1. In a dashboard, click the icon in the upper-right corner. The New Chart dialog box appears, which contains the list of available charts.
2. Select the chart that you want to add to the dashboard and click **Next Step**.
3. Select an option from the **Data Refresh mode** drop-down list. The following modes are supported:
   - Manual refresh: You must click the **Synchronize data** icon in the upper-right corner of the chart to refresh data.
   - Scheduled refresh: The system automatically refreshes data in the chart at the specified interval in the unit of seconds.
4. Click **Save**. The chart is added to the dashboard.

Configure a chart

- To refresh data in a chart, click the icon in the upper-right corner of the chart to trigger a query and synchronize data.

  > **Note** If cache is enabled for the chart, the cached content is also refreshed when you click this icon.

- To edit a chart, click the icon in the upper-right corner of the chart.
- To view a chart in full screen, click the icon in the upper-right corner of the chart.
- To change the data refresh mode of a chart or delete a chart, click the icon in the upper-right corner of the chart and select Basic information or Delete as required.

Use the auto layout feature

- You can drag the lower-right corner of a chart to adjust its size.
- You can drag the top of a chart to adjust its position.
- Dashboards adopt a fluid layout. If the width of the display window is equal to or greater than 768 pixels, charts on a dashboard are displayed based on specified percentages. If the width of the display window is less than 768 pixels, charts are displayed in the mode for mobile devices.
- When you adjust the size or position of a chart, other charts in the same dashboard automatically

adapt to the change, as designed in a fluid layout.

Configure filter interactions

You can configure filter interactions between charts in the same dashboard. Click the icon in the upper-right corner of a dashboard. The Linkage relationship settings dialog box appears.

- You can configure multiple filter interactions for a dashboard. For each filter interaction, you need to specify a trigger, an associated field, and a mapping relationship between the trigger and the associated field.
  - A trigger is a field that triggers a filter interaction. This field can be a dimension or an aggregated measure. Only fields that are used in the selected chart, instead of all fields in the dataset model that corresponds to the selected chart, can be specified as triggers.
  - An associated field can be any field or variable in the dataset model that corresponds to the selected chart. The data type of a trigger must be the same as the data type of its associated field.
- You can configure multiple filter interactions that have the same trigger. This way, one trigger is associated with multiple charts. A relationship diagram is displayed on the right of the Linkage relationship settings dialog box, showing the filter interactions between charts.

  > **Note** You can also configure multiple filter interactions whose associated fields belong to the same chart. All filter conditions, which are the mapping relationships defined in these filter interactions, take effect at the same time.
- After you configure filter interactions, each chart to which a trigger belongs has an icon in the upper-left corner. Move the pointer over the icon and an action prompt appears.

Configure global filters

You can configure global filters for a dashboard. Global filters allow you to filter data, within one or across multiple charts, by defining filter conditions or replacing variables. Click the icon in the upper-right corner of a dashboard. The Global filter settings dialog box appears.

- **Basic settings**

  The Global filter settings dialog box is divided into three sections: **Filter list**, **Associated chart** and **Category**, and **Filter configuration**.
- **Filter list**

  In the **Global filter settings** dialog box, click the plus sign (+) to the right of Filter list. A global filter is created with the default name: New Filter. To rename or delete the filter, move the pointer over the filter name and click the corresponding icon that appears to the right of the filter name.
- **Associated chart** and **Category**

  In the **Associated chart** section, select the charts that you want to associate with the global filter. Then, in the **Category** section, select at least one associated **field** or **variable** that you want to associate with the global filter. The input of the global filter is used as the value of the corresponding field to form a filter condition or used to replace the corresponding variable in the SQL query statement.
- **Filter configuration**

You can configure the following types of filters:

- **Drop-down list**

  Drop-down lists can be set only for global filters whose associated fields are dimensions. The options of a drop-down list are a set of distinct values of the associated field.

  - If the associated charts of a global filter are based on different datasets, the options of the drop-down list include values of all the associated fields in all the datasets.
  - If you set the Type parameter to Drop-down menu, the system automatically executes an SQL statement to query associated fields in the datasets and sets the values of these fields as options of the drop-down list. Therefore, you can enable cache and set a validity period.
  - If you do not want to use the values of associated fields as options of the drop-down list, select Custom options and click the plus sign (+) that appears. In the Edit custom options dialog box, enter one pair of option text and option value per line. Separate each option text and option value with a space. If the option text and option value are the same, you need only to enter one of them.

  If you select associated variables in the Category section, you still can use field values as options of the drop-down list.

  In the Filter configuration section, you can also enable the Multiple choice feature.

  > **Note** Assume that you select associated variables in the Category section and select Multiple choice for the drop-down list. If you select multiple options from the drop-down list, the selected options are converted to `'Option 1','Option 2','Option 3'` to replace the corresponding variable. Therefore, you need to include `in ()` in the SQL statement to ensure correct execution.

- **Date selection**

  Date selection can be set only for global filters whose associated fields are dimensions. The selected date can be converted to one of the following formats:

  - Date. Example: 2019-01-01.
  - Date and time, accurate to seconds. Example: 2019-01-01 12:00:00.
  - Date and time, accurate to minutes. Example: 2019-01-01 12:00.
  - Month. Example: 2019-01.
  - Week. Example: 2019-5th week.
  - Year. Example: 2019.

  You can set a default value for a date filter. The default value can be a specified date, whether the date is fixed on the timeline or moves on the timeline.

  > **Note** You can set default values only for date filters.

  If you select Multiple choice for a date filter, you can set the Date format parameter only to Date, Month, or Year. To specify multiple months or years as default values, you must select dates in the corresponding months or years.

  > **Note** Similar to drop-down list filters, if you select associated variables in the Category section and select Multiple choice for a date filter, you also need to modify the SQL statement to ensure correct execution.

- **Date range**

  Date ranges can be set only for global filters whose associated fields are dimensions. The formats to which a selected date range can be converted are the same as those for date filters. When you associate a date range filter with variables, you must select two variables, one as the start time and the other as the end time.

- **Text input box**

  Text input boxes can be set only for global filters whose associated fields are dimensions. If you set the query mode of a text input box filter to Auto query, after you enter a value in the input box, you still need to press the Enter key to trigger a query.

- **Number range input box**

  Number range input boxes can be set only for global filters whose associated fields are metrics. If you set the query mode of a number range input box filter to Auto query, after you enter a value in the input box, you still need to press the Enter key to trigger a query. When you associate a number range input box filter with variables, you must select two variables, one as the start value and the other as the end value.

- **Filter hierarchy**

  In the tree menu section, you can drag and drop filters to set a filter hierarchy. A parent filter is used as a drop-down list to filter options of its child filter.

- **Query mode**

  In the lower-right corner of the Global filter settings dialog box, you can set the query mode to Auto query or Manual query.

- **Auto query**: Changes to the value of the filter immediately trigger a query. For text input box filters and number range input box filters, even if you set the query mode to **Auto query**, you still need to press the Enter key to trigger a query after you enter a value.
- **Manual query**: If you set the query mode to Manual query, a **Query** button and a **Reset** button appear on the right of the global filter bar. To trigger a query, select or enter a value on the left of the global filter bar and click **Query**.

Perform drilling in a chart

To perform drilling in a chart, select the chart elements that you want to drill on, right-click, and then select the dimension that you want to drill to.

> **Note** If a chart contains a trigger configured for filter interactions, you cannot perform drilling in the chart.

- Each chart that allows drilling has an icon in the upper-left corner. If you move the pointer over the icon, an action prompt appears.
- If a chart allows drilling, its drilling path is displayed in the lower-left corner. You can drill to a level by clicking the level name in the drilling path.

  > **Note** For example, you can select the Shanghai, Shenzhen and Guangzhou chart elements to drill to the `education` field. This way, the chart displays the education level and salary distribution in Shanghai, Shenzhen, and Guangzhou.

- The drilling feature is implemented differently in charts configured by using the pivot-driven mode and in charts configured by using the chart-driven mode.
  - Pivot-driven mode

    In charts configured by using the pivot-driven mode, drilling is implemented by adding dimensions to or removing dimensions from filter conditions.

    > **Note** Pivot tables support roll-up or drill-down operations. A roll-up operation is to remove a dimension from a filter condition. A drill-down operation is to add a dimension to a filter condition. Pivot tables also allow you to drill down to a row or a column.

- Chart-driven mode
  - In table charts configured by using the chart-driven mode, drilling is implemented in a similar way it is implemented in pivot tables.
  - In other charts configured by using the chart-driven mode, a drilling operation replaces an existing dimension in a filter condition with the dimension you want to drill to. This way, data changes are presented from another perspective.

  You can perform drilling in the following types of charts configured by using the chart-driven mode:

  - Table chart
  - Column chart
  - Line chart
  - Scatter chart
  - Pie chart
  - Funnel chart
  - Secondary Y-axis chart

Edit a dashboard collection

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Dashboard** section of the **Resource management** page, click the Edit icon in the upper-right corner of the dashboard collection that you want to edit.

Delete a dashboard collection

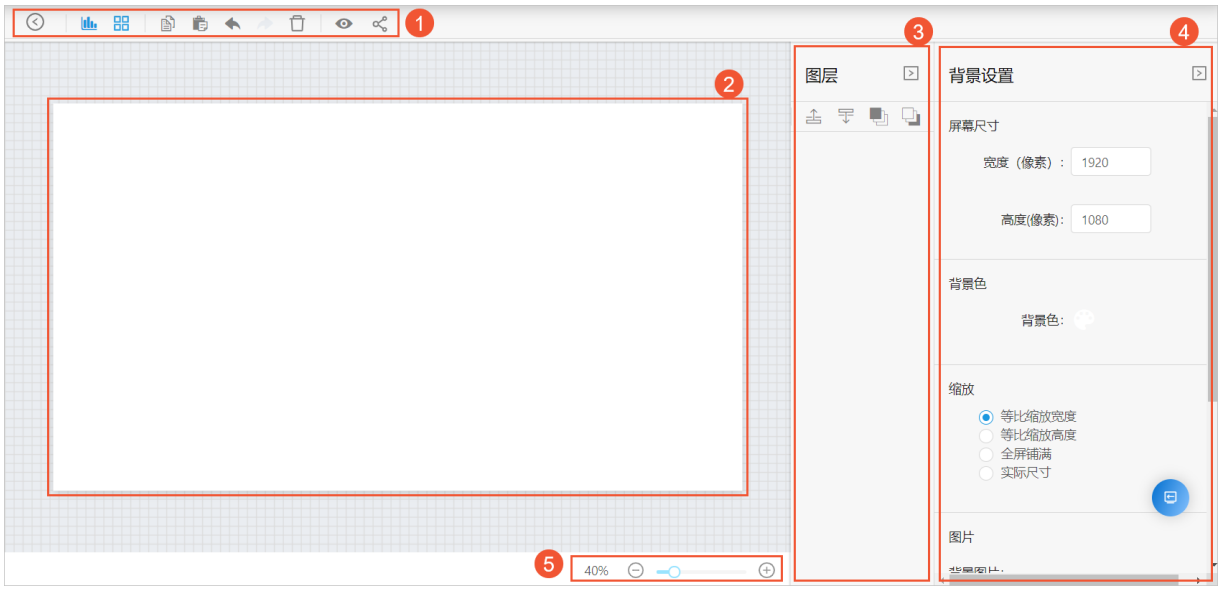
1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Dashboard** section of the **Resource management** page, find the dashboard collection that you want to delete.
5. Click the  icon.
6. In the message that appears, click **OK**.

6.4.7. Manage big screens

In addition to dashboards, Data Management (DMS) provides another type of visualization applications: big screens. Big screens provide customizable layouts and styles. You can combine visualization components with bountiful built-in auxiliary graphics to create visualized big screens of various visual styles.

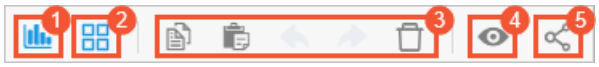
Dashboards and big screens are designed for different purposes. Dashboards are used to generate visual reports. Big screens are widely used for data viewing and presentation in a static or scrolling way. Therefore, it usually takes more time to create a big screen than to create a dashboard. This topic describes how to stylize a big screen.

Big screen editor



No.	GUI element
①	Toolbar
②	Canvas
③	Layer panel
④	Parameter settings panel
⑤	Zoom tool

Toolbar



No.	GUI element	Description
①	**Chart**	You can click this icon to add charts to a big screen.
②	**Auxiliary graphics**	You can click this icon to add the following four types of auxiliary graphics: rectangle, label, video, and timer.
③	**Operation icons**	You can click the operation icons to manage layers on the canvas. The following operation icons are provided: **Copy**, **Paste**, **Undo**, **Redo**, and **Delete**.
④	**Preview**	You can click this icon to open a new web page to view the display effect of the big screen that you are editing.
⑤	Share	You can click this icon to share visualized data.

- **Chart**

You can click the Chart icon to add charts. To add a chart, perform the following steps:

i. Click the **Chart** icon.

ii. In the **Add Chart** dialog box, select a chart and click **Next Step**.

iii. Select a mode from the **Data Refresh mode** drop-down list and click **Save**.

DMS provides the following two methods to update data:

- **Manual refresh**: If you select this mode, you must click **Synchronize Data** in the upper-right corner of the chart on the big screen each time you want to update data.
- **Scheduled refresh**: The system automatically updates data in the chart at the specified interval in the unit of seconds.

iv. Click **Save**. The chart is added to the canvas.

You can drag a chart to adjust its position. You can also drag the chart in the lower-right corner to resize the chart.

- **Auxiliary graphics**

DMS provides four types of auxiliary graphics. You can click the **Auxiliary graphics** icon and select an auxiliary graphic from the list.

Auxiliary graphic	Description
Rectangle	This auxiliary graphic is generally used to decorate the background of charts. You can set the background color, background picture, and borders for a rectangle.
Label	This auxiliary graphic is generally used to display texts. You can set the text font, margin, background color, and borders for a label.
Video	This auxiliary graphic is used to play online videos.
Time	This auxiliary graphic is used as a ticking clock.

- **Operation icons**

Operation icon	Description	Shortcut key in Mac OS	Shortcut key in Windows
Copy	You can click this icon to copy a chart or an auxiliary graphic.	Cmd+C	Ctrl+C
Paste	You can click this icon to paste a chart or an auxiliary graphic.	Cmd+V	Ctrl+V
Undo	When you edit the canvas, you can click the Undo icon to roll back an operation or click the Redo icon to restore an operation.	N/A	N/A
Redo			
Delete	You can click this icon to delete a chart or an auxiliary graphic.	Delete	Backspace or Delete

> Note
>
> - You can click the **Copy**, **Paste**, or **Delete** icon to copy, paste, or delete multiple layers at a time.
> - To select multiple layers, press and hold the Cmd or Alt key on the keyboard and click the layers.

- **Preview**

  After you create a big screen, you can click the **Preview** icon in the toolbar to preview the big screen on a new web page.

Canvas

Each chart or auxiliary graphic that is added to the canvas can be viewed as a **layer**. On the canvas, you can perform the following operations:

- You can drag a layer to adjust its position. You can also select a layer and use the arrow keys on the keyboard to fine-tune the position of the layer.

  > **Note** When you are dragging a layer, a prompt appears to the right of the layer to show the current position of the layer in the form of coordinates. Guides are also displayed on the canvas to help you align the layer.

- When you edit a part of the canvas, you can adjust the slider of the zoom tool to zoom in or zoom out on the canvas.
- You can press and hold the Cmd or Alt key on the keyboard, click multiple layers to select them, and then copy, delete, or align the layers at a time.
- To deselect one or more selected layers, click the blank area on the canvas.
- To edit a chart, click the **Edit** icon in the upper-right corner of the chart.

Layer panel

In the Layer panel, you can perform the following operations:

- Select a layer: You can select a layer on the canvas by selecting the corresponding item in the layer list.
- Select multiple layers: You can select multiple layers on the canvas by pressing and holding the Cmd or Alt key on the keyboard and selecting the corresponding items in the layer list.
- Adjust the position of a layer on the Z axis: After you select a layer, you can click the icons above the layer list to adjust the position of the layer on the Z axis. The following icons are provided: **Move up**, **Move down**, **Top**, and **Bottom**.

Parameter settings panel

The content that is displayed in the parameter settings panel varies with the layer that you select.

- By default, the **Background settings** panel is displayed as the parameter settings panel. The following table describes the parameters in the Background settings panel.

Parameter	Description

Parameter	Description
Screen size	You can adjust the size of the canvas based on the display terminal of the big screen.
Background color	You can specify a background color for the big screen.
Zoom	○ **Proportional scaling width**: The width of the canvas is the same as the width of the display terminal. The height of the canvas is proportionally scaled. ○ **Proportional scaling height**: The height of the canvas is the same as the height of the display terminal. The width of the canvas is proportionally scaled. ○ **Full screen**: Both the height and width of the canvas are the same as the height and width of the display terminal. In this mode, the canvas may be deformed on the display terminal. ○ **Actual Size**
Picture	You can upload a background picture for the big screen.

- If a single layer is selected, the **Chart** panel is displayed as the parameter settings panel. You can set the following parameters in the Chart panel:
  - **Chart size**
  - **Chart location**
  - **Background**
  - **Border**
  - **Data Refresh mode**
- If multiple layers are selected, the **Layer alignment** panel is displayed as the parameter settings panel. You can click the following icons to operate the layers: **Top alignment**, **Left alignment**, **Center horizontally**, **Vertically centered**, **Right alignment**, and **Bottom alignment**.

Create a big screen

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Big Screen** section of the **Resource management** page, click **New big screen**.
5. In the **New Big Screen** dialog box, set the **Name** and **Description** parameters.
6. Click **Save**. You can view the new big screen in the **Big Screen** section.

Edit a big screen

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Big Screen** section of the **Resource management** page, find the big screen that you want to edit.

5. On the page that appears, edit the big screen based on your requirements. For more information about how to edit a big screen, see Big screen editor.

Delete a big screen

1. Log on to the DMS console.
2. Go to the Data Visualization tab.
3. In the left-side navigation pane, click **Data Display**.
4. In the **Big Screen** section of the **Resource management** page, find the big screen that you want to delete.
5. Click the icon.
6. In the message that appears, click **OK**.

6.5. Heterogeneous database migration

6.5.1. Overview

Advanced Database & Application Migration (ADAM) can help you migrate IT systems in your enterprise to the cloud in an easy and reliable manner.

Context

ADAM is a digital migration solution that can help you rearchitect your traditional databases and applications in the cloud. ADAM is developed based on years of experience of Oracle database and application architecture analysis, architecture selection, and business transformation in Alibaba Group. ADAM provides professional data and application transformation services for enterprises, backed by the comprehensive partner ecosystem. ADAM also provides a whole set of product-based services, such as database and application architecture analysis, architecture selection, business transformation, data migration, and O&M optimization.

Features

- Database evaluation

  This feature provides intelligent evaluation and analysis based on the collected source database data. Then, the feature generates a report that contains multiple factors, such as the database migration solution, compatibility of the destination database, migration risks, application transformation suggestions, and migration costs. To use the feature, log on to the Data Management (DMS) console. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Database Evaluation**.

- Database transformation and migration

  This feature provides intelligent tools for database transformation and migration. This facilitates source database comparison, schema migration, and schema revision based on database evaluation results. To use the feature, log on to the DMS console. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Database Transformation and Migration**.

- Application evaluation and transformation

This feature provides application transformation items such as call stacks and SQL statements, and provides the architecture migration guide to sort the architecture of large-scale cluster migration. To use the feature, log on to the DMS console. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Application Evaluation and Transformation**.

- Migration lab

  This feature provides an SQL statement comparison test platform, an SQL statement automatic transformation platform, and tools for periodic data collection. To use the feature, log on to the DMS console. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **Migration Lab**.

- SQL conversion

  This feature provides tools to transform statements in Oracle, Teradata, or Db2 databases to statements that can be executed in MySQL databases, PolarDB for Oracle clusters, or AnalyticDB for PostgreSQL databases. To use the feature, log on to the DMS console. In the top navigation bar, click **DTS**. In the left-side navigation pane, click **SQL Conversion**.

6.5.2. Database evaluation

6.5.2.1. Collect database information

ADAM provides two database information collection methods: collect database information online and collect database information by using Database Collector.

Context

- Collect database information online: If you want to collect the information of a source database online, you must ensure that the source database can be accessed over the Internet. In addition, you must add the ADAM servers to the whitelist of the source database.
- Collect database information by using Database Collector: If the source database is inaccessible over the Internet, you can download the Database Collector client to collect the information of the database.

Collect the information of a database online

1. Log on to the DMS console.
2. In the top navigation bar, choose **DTS**.
3. In the left-side navigation pane, choose Heterogeneous Database Migration(ADAM) > Database Evaluation. In the **Collect DB Information** wizard, click **Online Database Collection**.
4. Click **Create Collection Task** to create a collection task.
5. Log on to the source database, and then create and authorize an account to collect the information of the source database.
   - Information of databases in Oracle 10g, 11g, or 12c can be collected by ADAM. If the source database is an Oracle 10g, 11g, or 12c database, create a local user in a non-CDB architecture.
     a. Create a user named eoa_user and set the password to eoaPASSW0RD****.

     ```
     create user eoa_user identified by eoaPASSW0RD**** default tablespace users;
     ```

b. Grant the permissions to query data from the source database to the eoa_user user.

```
grant connect,resource,select_catalog_role,select any dictionary to eoa_user;
```

c. Grant the permissions on the DBMS_LOGMNR package to the eoa_user user.

> **Note** If the source database is an Oracle 10g database, run the following command before you grant permissions on the DBMS_LOGMNR package to the eoa_user user:
>
> ```
> create or replace public synonym DBMS_LOGMNR for sys.dbms_logmnr;
> ```

```
grant execute on DBMS_LOGMNR to eoa_user;
```

d. Grant the permissions on the DBMS_METADATA package to the eoa_user user to query DDL statements for objects.

```
grant execute on DBMS_METADATA to eoa_user;
```

e. Grant the permissions to query transactions to the eoa_user user.

```
grant select any transaction to eoa_user;
```

f. Grant the permissions to query tables to the eoa_user user.

```
grant select any table to eoa_user;
```

g. Grant the permissions to analyze tables to the eoa_user user.

```
grant analyze any to eoa_user;
```

h. Grant the permissions to generate random numbers to the eoa_user user.

```
grant execute on dbms_random to eoa_user;
```

- If the source database is an Oracle 12c database, connect to an Oracle 12c container database (CDB) to create a common user.

```
create user c##eoa_user identified by "eoaPASSW0RD****" default tablespace users;
grant connect,resource,select_catalog_role,select any dictionary to c##eoa_user conta
iner=all;
grant execute on DBMS_LOGMNR to c##eoa_user container=all;
grant execute on dbms_metadata to c##eoa_user container=all;
grant select any table to c##eoa_user container=all;
grant select any transaction to c##eoa_user container=all;
grant analyze any to c##eoa_user container=all;
grant execute on dbms_random to c##eoa_user container=all;
alter user c##eoa_user set container_data=all container=current;
```

- Information of databases in Teradata 13, 14, and 15 can be collected by ADAM.

  If the source database runs Teradata 13, 14, or 15, grant the permissions on the DBC of the database to the user that you want to use to collect the information of the source database.

```
grant select,show on dbc to (username);
```

- Information of Db2 for LUW databases can be collected by ADAM.

  If the source database is a Db2 for LUW database, the user that you want to use to collect the information of the source database must be granted the permissions to run the db2look command. You must assign the DBA role to the user.

6. After you configure the source database account, click **Next** to go to the **Test Connectivity and Start Collection** wizard.
   i. **Collection Task Name**: Enter the name of the collection task.
   ii. **Source Database Type**:
   - Select **ORACLE** as the source database type.

     Database Name /Service Name /SID: Specify the database name, service name, or system ID (SID) of the source Oracle database.
   - Select **TERADATA** or **Db2_LUW** as the source database type.

     Database Name: Enter the name of the source database.

   iii. Set the **Source Database Network Type** parameter to **Database with Public IP Address**.
   iv. Enter additional information of the source database.
   - **Host IP**: Enter the IP address of the source database.

     > **Note** Add the IP addresses of the ADAM servers to the whitelist of the source database to ensure that ADAM can collect data from the source database online.
   - **Port Number**: Enter the port that is used to connect to the source database.
   - **Username**: Enter the username of the user account that you created to collect database information.
   - **Password**: Enter the password of the preceding user account.
   - (Optional)**Advanced Settings**: Specify the encoding method of the source database in the Included Schemas and Excluded Schemas fields.
7. Click **Test Connectivity** to perform a test. After the test succeeds, click **Start Collection**.
8. After the collection task is complete, select the collection task and click **Next: Create Profile** to create a database profile.

Collect the information of a database by using Database Collector

If you cannot access the source database over the Internet or from Alibaba Cloud, you can use Database Collector to collect information of the source database offline.

1. Log on to the DMS console.
2. In the top navigation bar, choose **DTS**.
3. In the left-side navigation pane, choose Heterogeneous Database Migration(ADAM) > Migration Lab. In the **SQL Periodic Collection** section, click **Download** to download the Database Collector client.
4. In the Download Database Collector panel, download the Database Collector client based on the operating system of your on-premises machine. Then, decompress the package and install the Database Collector client.

> **Note** We recommend that you do not install the Database Collector client on the server of the source database. The on-premises machine on which you want to install the Database Collector client must meet the following requirements:
>
> - Network: accessible to the source database
> - CPU: 2 cores
> - Memory: 8 GB
> - Hard disk: 100 GB free space

5. Create and authorize an account to collect database information.
   i. Log on to the source database.
   ii. Create a temporary user by using the credentials of a user that is granted the SYSDBA permission, and then grant the required permissions to the temporary user. If an existing user of the source database is granted the required permissions, you can use the credentials of the existing user to collect information of the source database.

- If the source database is an Oracle 10g, 11g, or 12c database, create a local user in a non-CDB architecture.
  a. Create a user named eoa_user and set the password to eoaPASSW0RD****.

```
create user eoa_user identified by eoaPASSW0RD**** default tablespace users;
```

  b. Grant the permissions to query data from the source database to the eoa_user user.

```
grant connect,resource,select_catalog_role,select any dictionary to eoa_user;
```

  c. Grant the permissions on the DBMS_LOGMNR package to the eoa_user user.

> **Note** If the source database is an Oracle 10g database, run the following command before you grant permissions on the DBMS_LOGMNR package to the eoa_user user:
>
> ```
> create or replace public synonym DBMS_LOGMNR for sys.dbms_logmnr;
> ```

```
grant execute on DBMS_LOGMNR to eoa_user;
```

  d. Grant the permissions on the DBMS_METADATA package to the eoa_user user to query DDL statements for objects.

```
grant execute on dbms_metadata to eoa_user;
```

  e. Grant the permissions to query transactions to the eoa_user user.

```
grant select any transaction to eoa_user;
```

  f. Grant the permissions to query tables to the eoa_user user.

```
grant select any table to eoa_user;
```

  g. Grant the permissions to analyze tables to the eoa_user user.

```
grant analyze any to eoa_user;
```

  h. Grant the permissions to generate random numbers to the eoa_user user.

```
grant execute on dbms_random to eoa_user;
```

- If the source database is an Oracle 12c database, connect to an Oracle 12c CDB to create a common user.

```
create user c##eoa_user identified by "eoaPASSW0RD****" default tablespace users;
grant connect,resource,select_catalog_role,select any dictionary to c##eoa_user c
ontainer=all;
grant execute on DBMS_LOGMNR to c##eoa_user container=all;
grant execute on dbms_metadata to c##eoa_user container=all;
grant select any table to c##eoa_user container=all;
grant select any transaction to c##eoa_user container=all;
grant analyze any to c##eoa_user container=all;
grant execute on dbms_random to c##eoa_user container=all;
alter user c##eoa_user set container_data=all container=current;
```

- Teradate 13 / 14 / 15

  If the source database runs Teradata 13, 14, or 15, grant the permissions on the DBC of the database to the user that you want to use to collect the information of the source database.

  ```
  grant select,show on dbc to (username);
  ```

- Db2 for LUW

  If the source database is a Db2 for LUW database, assign the DBA role to the user that you want to use to collect information of the source database.

iii. Collect structured data to generate feasibility reports and compatibility reports.

Database Collector can collect the information of Oracle 10g, 11g, and 12c databases, Teradata 13, 14, and 15 databases, and Db2 for LUW databases. If issues occur when you collect database information, submit a ticket and attach the log files that are stored in the logs directory.

a. Run the collect command to collect the information of the source database. If the Database Collector client runs on Windows, run the .bat command. If the Database Collector client runs on Linux, run the .sh command.

   a. Oracle 10g

      ```
      collect_10g[.sh|.bat] -h -u -p -d <service_name>;
      ```

   b. Oracle 11g

      - Oracle 11g R1

        ```
        collect_11gR1[.sh|.bat] -h -u -p -d <service_name>;
        ```

      - Oracle 11g R2

        ```
        collect_11gR2[.sh|.bat] -h -u -p -d <service_name>;
        ```

   c. Oracle 12c

      > **Note** You can use the `collect_11gR2` script to collect the information of a pluggable database (PDB) in Oracle 12c. collect_12c[.sh|.bat] -h <host> -u <username> -p <password> -P <port> -d <service_name> -s <sid>

   d. Teradate 13 / 14 / 15

      ```
      collect_td[.sh] -h ip -p password -u username;
      ```

   e. DB2_LUW

      ```
      collect_db2_luw[.sh] -h ip -u username -p password -d databasename  -P port
      ;
      ```

      -h: the IP address of the source database. - -u: the username of the eoa_user user. The value is eoa_user. - -p: the password of the eoa_user user. The password is eoaPASSW0RD. -P: the port that you want to use to collect the information of the source database. Example: 1521. -d: the service name of the source database. If the source database is an Oracle 12c database, specify the service name of the specified PDB. -s: the name of the Oracle database instance. This option is required only for Oracle 12c.

b. Export the collection results.

   After the collection task is complete, Database Collector generates a packet and returns the path of the packet. If the execution of the collection task is successful, the following log entries are displayed:

   ```
   [***] ************************************************************
   [***] *     Collect Successfully!
   [***] *
   [***] * Complete the file packaging, the package result path is:
   [***] *        ~rainmeter/out/data.zip ***************************************
   *************************
   ```

c. After the collection task is complete, you can delete the temporary user.

Use the credentials of a user that is granted the SYSDBA permission to connect Database Collector to the source database and execute the drop statement to delete the temporary user.

a. Oracle 10g, 11g, or 12c (non-CDB)

```
drop user eoa_user cascade;
```

b. Oracle 12c (CDB)

```
drop user c##eoa_user cascade;
```

What to do next

For more information about how to create a profile, see Manage a database profile.

6.5.2.2. Manage a database profile

A database profile is used to evaluate a source database. This helps you monitor the status of your source database. When you migrate or transform a source database, you can search for the database based on the database profile.

Create a profile

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Evaluation.
3. In the DB Evaluation Process wizard, click the **Source DB Profile** step.
4. On the page that appears, click **Create Profile**.
5. In the **Create Profile** panel, set the **Profile Name**, Type, and Report Language Type parameters, and then click **Upload** to upload a file of data collected from the source database. After the file is uploaded, click **Create**.

> **Note** The database profile can be created in 1 to 30 minutes. The duration depends on the size of the data collected from the source database.

View the profile

After the state of the database profile changes to **Complete**, click **Details** in the **Actions** column to view the profile.

- Basic information statistics

> **Note** The basic information statistics feature is available only for the profile of an Oracle database.

On the **Overview** tab, view the basic information of the source database.

- Session: the connection status of the database. The higher the value is, the more sessions are established to the database.
- Complexity: the complexity of the database evaluated based on the collected information about the source database, such as scenarios and features.

- Risk: the risk level of the database. The higher the value is, the more likely the database is exposed to performance risks related to SQL statements or objects.
- Hotspot: indicates whether the database contains objects that are frequently accessed. The higher the value is, the more objects are frequently accessed.
- Scale: the scale of database resources. The higher the value, the larger the database scale.
- Load: the performance of the database. The higher the value, the larger the load of the database.

- Distribution of main objects

  On the **Overview** tab, view the distribution of object types in the source database.

- Details of profile analysis results

  On the **Details** tab, view the multi-dimensional analysis results of the source database.

  - Performance

    This metric shows the transactions per second (TPS), queries per second (QPS), CPU utilization, and load of the database.

  - Capacity

    This metric shows the capacity rankings of database schemas and the capacity proportions of different object types such as table, index, and large object (LOB).

  - Oracle Features

    This metric shows features used by your source database in a tree diagram and a table. The tree diagram provides two levels to list all the features collected from the database. If you click a feature, the objects that use the feature are listed in a table. You can search for features by entering a condition in the search box.

  - External Dependency

    This metric shows database links and link details. You must revise schemas when you transform databases that contain external dependencies.

  - Other Dimensions

    This metric shows special tables and SQL statements, such as tables without primary keys, high growth tables, and SQL statements with aggregate functions. You can click **View** in the Actions column to view details.

  - Object Details

    This metric shows object information from different dimensions, such as object feature tags, relationships such as association and dependency, and features contained in objects. This metric allows you to view the object information without the need to query the source database. You can click **View Object Details** in the **Actions** column to view the details of an object.

    In the Basic Information section of the View Object Details dialog box, you can view the basic information and DDL statements of the object. In the Details section, you can view the analysis data provided by Advanced Database & Application Migration (ADAM), such as the referenced objects, dependent objects, and features of the object.

- Object Search

  The Object Search feature allows you to search for objects when you transform databases and applications. Exact matches, fuzzy matches, and type-based matches are supported. You can combine different filter conditions to search for objects in the database profile. You can search for objects by schema, DDL, object type, or tag. You can also view the dependencies and tags of objects.

Supported operations

Operation	Description	Procedure
Update the profile	You can manage multiple versions for the profile of the source database. If the data in the source database changes after the profile is created, you can append a new data file to the original profile.	In the **Source DB Profile** step, find the database profile that you want to manage and click **Append** in the **Actions** column. In the Append Profile panel, click **Upload** to upload a data file and click **Create**. ADAM then generates a new version of the profile based on all the data files.
Grant permissions on the profile	You can authorize other users to access the profile. You can also revoke permissions on the profile from other users. **Notice** Use the authorization feature with caution. You are liable for data disclosures caused by improper authorization.	In the **Source DB Profile** step, find the database profile that you want to manage and click **Authorize** in the **Actions** column. In the Authorize panel, select **Confirm**, enter an Apsara Stack tenant account ID, and then click OK. The validity period of the authorization is one year. Authorized users can view the profile and create new analysis projects based on the profile.
Delete the profile	If you delete a database profile, all analysis data of the profile is deleted.	In the **Source DB Profile** step, choose **...** > **Delete**.

What to do next

Select a database profile and click **Next: Select Destination Database(1)** to go to the **Select Destination DB** step.

6.5.2.3. Select a destination database

A variety of types of databases are available on Apsara Stack. Advanced Database & Application Migration (ADAM) analyzes the profile of the source database and the compatibility between the source database and various destination databases. ADAM provides you with the basis for selecting a destination database based on destination database compatibility, which regular users are most concerned about.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Evaluation.

3. Click the **Select Destination DB** tab. In the profile list, select the database profile that you want to evaluate.
4. Click **Next: Select Destination Database**.
5. Click the **Compatibility** tab.
   - The DB Object Compatibility table includes the compatibility, the number of compatible objects, and the number of incompatible objects.
     - **Compatibility (%)**: The compatibility is equal to the number of compatible objects divided by the total number of objects.
     - **Compatible with ADAM**: the objects that are compatible with the destination database, and the objects that ADAM can transform. ADAM automatically transforms incompatible objects to compatible objects.
     - **Incompatible**: the objects that are incompatible and must be manually transformed. ADAM provides comprehensive suggestions on how to transform these objects.
   - The SQL Compatibility table includes the compatibility, the number of compatible SQL statements, the number of incompatible SQL statements that can be transformed, and the number of incompatible SQL statements.
     - **Compatibility**: The compatibility is equal to the total number of directly compatible SQL statements and SQL statements that are compatible after transformation divided by the total number of objects.
     - **Compatible**: the number of objects that are directly compatible or compatible after transformation.
     - **Compatible After Conversion**: ADAM provides SQL statements that can be executed in the destination database, but you must modify the corresponding SQL statements in the application.
     - **Incompatible**: incompatible objects.
6. (Optional)ADAM allows you to obtain suggestions only on the database schemas that you want to migrate. You can select the schemas from the **Select Schema** drop-down list.

What to do next

Click **Next: Create Destination Database Evaluation**.

6.5.2.4. Evaluate a database

Advanced Database & Application Migration (ADAM) allows you to evaluate the compatibility and specifications of destination databases, and migration risks. This helps you understand the feasibility and transformation costs of database migration.

Create a database evaluation project

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Evaluation.
3. On the **Source DB Profile** tab, select a database profile, and click **Next: Select Destination Database**.
4. On the **Select Destination DB** tab, click **Next: Select Destination Database** and then click **Create Project**.

> **Note** You can also click **Create Project** on the **Evaluate Dest. DB** tab to create an evaluation project.

5. Create a database evaluation project. Set the parameters that are described in the following list:
   - Project Name: the name of the project. This parameter is required.
   - Source Database Profile: the database profile.
   - Project Type: the type of the destination database.
   - Destination Database Version: the version of the destination database. This parameter is required.
   - Are you sure that you want to evaluate invalid objects?
     - Yes: evaluates invalid objects in the source database profile.
     - No: does not evaluate invalid objects in the source database profile.
   - Kernel Version: If you set Project Type to PolarDB-O Engine, this parameter is required.
   - Report Language: the language of the evaluation reports.
6. Click **Create**. The evaluation project is created and automatically performed.

Evaluate a database

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Evaluation.
3. Click the **Evaluate Dest. DB** tab.
4. In the project list, find a project and click **Details** in the **Actions** column to go to evaluation details page of the destination database.

> **Note** You can view the evaluation status and process of projects in the project list. You can go to the evaluation details page only if the evaluation status of the project is **Complete**.

5. View the evaluation overview.
   - **Compatibility**: the compatibility between the source and destination databases. The higher the compatibility, the fewer the SQL statements and objects that need to be transformed.
   - **Transformation**: the focus areas of the destination database transformation.
     - **Object Transformation**: the database objects that are transformed by ADAM and do not need to be modified.
     - **Application Transformation**: the focus areas of application transformation that are obtained by analyzing SQL statements collected from the database during preliminary evaluation.
   - **Specifications**: the required specifications of the destination database and migration cost calculated by ADAM based on collected data. The evaluation of the specifications is subject to the data collection environment. Purchase databases based on your business requirements.
     - **Destination DB Specifications**: the specifications of the recommended destination database.
     - **Estimated Cost**: the estimated cost of buying the recommended destination database.

- **Risks**: the risks of database migration and transformation.
  - **Source DB Risk**: the existing risks of the source database.
  - **Migration Risk**: the risks that may arise when the database is migrated.
- **Overall Compatibility**: the overall compatibility between the source and destination databases.

6. View the evaluation details.
   - **Object Compatibility**

     The compatibility of all objects in the source database.
   - **SQL Compatibility**

     The SQL Compatibility panel shows the evaluation results of SQL statements collected from the source database.
   - **Project Transformation**

     The Project Transformation panel lists the focus areas of database transformation. You can transform your database objects based on these focus areas. You can also apply to use the ADAM database transformation feature to transform database objects. Some objects may need to be manually modified.
   - **Destination Database Solution**

     The Destination Database Solution panel provides specifications of the recommended destination database and migration plan guidelines to help you migrate databases to Alibaba Cloud. The specifications of the recommended destination database are generated based on the configurations, performance, SQL, and external dependencies of the source database, and the comprehensive analysis of the source and destination databases.
   - **Migration Risk**

     Migration risks include risks from source and destination databases.
   - **Project Dependency (Schema)**

     The Project Dependency (Schema) panel shows dependent and referenced objects, and provides suggestions on these objects.
7. Download the database evaluation report.

   In the lower part of the **Evaluate Dest. DB** tab, click **Download Simple Report** or **Download All Report** to obtain the database evaluation report generated by ADAM.

6.5.3. Database transformation and migration

6.5.3.1. Overview

Advanced Database & Application Migration (ADAM) provides the intelligent database transformation and migration feature to facilitate source database comparison, schema migration, schema revision, and data migration based on database evaluation results.

Supported database types

ADAM allows you to migrate data from on-premises databases to ApsaraDB. The destination databases must be of the following types:

- PolarDB for Oracle
- ApsaraDB RDS for MySQL
- ApsaraDB RDS for PostgreSQL
- PolarDB-X
- AnalyticDB for PostgreSQL

> **Note** You must configure an IP address whitelist when you create a destination database. For more information, see Configure an IP address whitelist.

Procedure

1. Create a migration project: Create a migration project based on database evaluation results.
2. Run a precheck: Check the account permissions, plug-ins, version, character sets, clocks, and resources of the destination database to ensure a smooth migration.
3. Optional. Verify the source database: Analyze the source database, track database changes, and update the migration project based on the database changes. To verify a migration project, make sure that the ADAM server can access the source database. If you skip this step, the data to be migrated to ApsaraDB may not be the latest data. In this case, you must collect and evaluate the database data again.
4. Migrate and revise schemas: Migrate objects to the destination database as possible as you can. ADAM evaluates and verifies the objects and provides solutions for incompatible objects. You can troubleshoot issues based on the error messages and try again.
5. Optional. Track incremental data by performing data comparison: Track the major changes to the data and schemas of your source database. This improves the migration efficiency of changed DDL statements.
6. Migrate data: Use Data Transmission Service (DTS) to migrate data.

6.5.3.2. Configure an IP address whitelist

To make sure that Advanced Database & Application Migration (ADAM) can connect to and use the destination database, you must configure an IP address whitelist when you create the destination database.

Configure an IP address whitelist based on the database type

- For more information about how to configure an IP address whitelist for a PolarDB instance, see *Configure an IP address whitelist* in *User Guide* of *PolarDB*.
- For more information about how to configure an IP address whitelist for an ApsaraDB RDS for MySQL instance, see *Configure an IP address whitelist for an ApsaraDB RDS for MySQL instance* in *User Guide* of *ApsaraDB RDS for MySQL*.
- For more information about how to configure an IP address whitelist for an ApsaraDB RDS for PostgreSQL instance, see *Configure an IP address whitelist for an ApsaraDB RDS for PostgreSQL instance* in *User Guide* of *ApsaraDB RDS for PostgreSQL*.
- For more information about how to configure an IP address whitelist for a PolarDB-X instance, see *Set an IP address whitelist* in *User Guide* of *PolarDB-X*.
- For more information about how to configure an IP address whitelist for an AnalyticDB for PostgreSQL instance, see *Configure an IP address whitelist* in *User Guide* of *AnalyticDB for PostgreSQL*.

Usage notes

If ADAM and the destination database are not in the same region, add the IP address of the machine on which ADAM is deployed to the IP address whitelist of the destination database.

6.5.3.3. Create a migration project

You can create a migration project in the Advanced Database & Application Migration (ADAM) console. Then, you can migrate data from a source database to a destination database based on database evaluation results. The migration project maximizes the compatibility between the source and destination databases by converting data types. ADAM also provides a schema migration tool that automatically converts data types based on the migration project.

Prerequisites

- Evaluate a database is complete.
- A destination database instance is created.
- If you migrate data to a PolarDB-X instance, you must first create a database and bind it to an ApsaraDB RDS instance. The database must have the same name as a schema on the PolarDB-X instance.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. On the **Transform and Migrate DB** page, click **Create Migration Project**.
4. In the **Create Migration Project** panel, set the parameters.

Parameter	Description
Project Name	The descriptive name of the migration project.
Evaluate DB	○ Select Database Evaluation: In the database evaluation list, select a database that is evaluated. ○ Upload Database Evaluation: Select a migration project type from the Project Type list and click **Upload File**.
IP Address	The IP address of the destination database.
Database Name	The name of the destination database.
Encoding Method	The encoding mode for the destination database.
Port Number	The public port number of the destination database.
Username	The username used to log on to the destination database.
Database Password	The password used to log on to the destination database.

5. Click **Test Connectivity** to perform a test. After the test is complete, click **Create**.

After the status of the migration project changes to `ACTIVE`, the project is created.

What to do next

After the migration project is created, ADAM automatically performs a precheck. For more information, see Run a precheck.

6.5.3.4. Run a precheck

Advanced Database & Application Migration (ADAM) runs a precheck to check the permissions, plug-ins, and version of the destination database. This ensures smooth migration.

Context

After you create a migration project, ADAM automatically runs a precheck. A precheck involves the following items:

- Permissions granted to the destination database account:
  - Check whether the account of the destination database has the permissions to create users.
  - Check whether the account of the destination database has the permissions to grant permissions to users.
  - Check whether the account of the destination database has the permissions to create and delete schemas.
  - Check whether the account of the destination database has the permissions to create and delete DDL statements.
- Plug-ins installed on the destination database:

  Check whether the required plug-ins are installed on the destination database.
- The version of the destination database:

  Check whether the version of the evaluated destination database is the same as the version of the destination database that is used to migrate schemas.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. After the status of the project changes to `ACTIVE`, click **Details**.

   > **Note** If the status of the project is ACTIVE and the font color is yellow, you can click Details and fix the issue based on the precheck result. If the font color of ACTIVE remains yellow after you fix the issue, ignore the issue.

4. Click **precheck** to view the precheck result.

What to do next

On the **precheck** tab, click **Next: Verify Source Database**. For more information, see Verify the source database.

6.5.3.5. Verify the source database

You can verify whether major changes are performed on the data and schemas of the source database since the last data collection. This helps you migrate databases.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. Find your migration project and click **Details** in the **Actions** column.
4. Click the **Verify Source DB** wizard to view the changes in the source database.
   - Init: the initial number of objects in the project.
   - Unchanged: the number of objects that remain unchanged in the project.
   - Changed: the number of objects that are changed in the project.
   - Deleted: the number of objects that are deleted from the project.
   - New: the number of new objects in the project.

What to do next

Click **Next: Schema Migration** to migrate the schemas of the source database. For more information, see Migrate and revise schemas.

6.5.3.6. Migrate and revise schemas

ADAM allows you to manage the permissions of users to migrate schemas by configuring a migration rule. ADAM revises the DDL statements of an object that fails to be migrated based on the information provided by the schema revision feature and executes the revised DDL statements in the destination database.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. Find the migration project and click **Details** in the **Actions** column.
4. Click the **Schema Migration/Revision** wizard.
5. (Optional)Configure users to migrate schemas. On the **Rule Configuration** tab, specify **User-specific Schema Migration** based on your needs.
   - Unified User: All schemas belong to a user. This is the default value.
   - Separate Users: Each schema belongs to a user.
     a. Enter a password.
        - If you have created a user, enter the password of the account.
        - If you have not created a user, ADAM automatically creates a user in the destination database. Then, you must specify a password. Otherwise, the password used to log on to the destination database is specified by default.

b. Add or remove schemas of users as needed.

c. Click **Save**.

d. In the message that appears, click **OK**.

> **Note** If you configure separate users, you must make sure that these users are consistent with the users of the source database and specify corresponding schemas for the users.

6. Click the **Schema Migration** tab to migrate schemas.

i. Click **Start Schema Migration**. In the message that appears, click **OK**.

> **Notice**
>
> - If you start schema migration, all schema objects in the destination database are deleted. Make sure that no important schema objects exist in the destination database in advance.
> - If you migrate schemas to a PolarDB-X database, the migration of FOREIGN KEY constraints may require a long period of time.
> - If the schema migration fails, you can download the details to view the information about the schema migration failure.

ii. (Optional)Perform other operations as required.

- Click **Remigrate All Schemas** to re-migrate DDL statements. All the DDL statements will be re-migrated.
- Click **Remigrate Failed Schemas** to migrate the DDL statements that fail to be migrated.
- Click **Stop Migration** to stop migrating the current DDL statements.
- Click **Custom Schema Migration** to customize the schema migration operation. To meet different schema migration requirements, ADAM allows you to customize schema migration operations. You can migrate schemas by specifying the Object Type, Schema, or Status parameter.

7. Click the **Schema Revision** tab. On the page that appears, click **Revise** in the **Actions** column.

- **Transformation**: View the current DDL statements that ADAM transforms. You must manually transform unsupported DDL statements.
- **Dependent Objects**: View the dependent objects of the current object.
- **Referenced Objects**: View the referenced objects of the current object.
- **Object Features**: View the features of the current object.

What to do next

- If a large number of data changes occurred in the source database, you can click the **Compare Increments (Optional)** step to check the incremental data. For more information, see Track incremental data by performing data comparison.
- If the data in the source database slightly changed, you can click the **Migrate Data** tab to directly migrate data. For more information, see Migrate data.

6.5.3.7. Track incremental data by performing data comparison

If the data or schemas of your source database have undergone major changes, you can perform data comparison to track the changes. This feature facilitates the migration of changed DDL statements.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. Find your migration project and click **Details** in the **Actions** column.
4. Click the **Compare Increments (Optional)** step.
5. Click **Start** to perform data comparison and track the incremental data.
6. After data comparison is complete, click **Incremental Data Comparison History** to view the records.

   If you find that schema changes exist after data comparison, you can return to the previous step and click **Start** to synchronize the schemas. For more information, see Migrate and revise schemas.

6.5.3.8. Migrate data

This topic describes how to migrate data.

Prerequisites

The schema migration and revision are complete. For more information, see Migrate and revise schemas.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Database Transformation and Migration.
3. Find your migration project and click **Details** in the **Actions** column.
4. Click the **Data Migration** step.
5. Migrate data.
   i. In the **Data Migration** drop-down list, click **Create Migration Project**.
   ii. In the **Create Migration Project** panel, set the parameters.
   iii. In the lower part of the page, click **Create**.

   After the configuration is complete, the migration task automatically starts. However, errors may occur if a large amount of data is migrated at a time. Therefore, Advanced Database & Application Migration (ADAM) migrates data in batches.

iv. (Optional) Click **View Details** to go to the DTS console. On the Overview page, you can check the details of the migration task.

> **Note** If the data migration instance cannot meet your requirements for data volume, you can click **Upgrade** in the Actions column to upgrade the specifications of the data migration instance.

6.5.4. Application evaluation and transformation

6.5.4.1. Overview

After your source database is transformed and migrated, you can transform your applications. However, applications are much more complex than databases, and the code involved may be developed by different developers. Application transformation has become a pain point for data migration to the cloud. Advanced Database & Application Migration (ADAM) provides the application evaluation and transformation feature to help you transform your applications.

Core features

- Provides application transformation items such as call stacks and SQL statements.
- Analyzes application usage information, such as framework and performance.
- Sorts out architectures to migrate large-scale clusters.

Procedure

1. Collect application information
2. Deploy a data collection environment
3. Create an application profile
4. Evaluate applications
5. Perform static application transformation

> **Note** If you perform dynamic application evaluation and transformation, complete Steps 1, 2, 3, and 4. If you perform static application evaluation and transformation, skip to Step 5.

6.5.4.2. Collect application information

Advanced Database & Application Migration (ADAM) allows you to collect data from Java applications that use JDK 1.6 or later and helps you evaluate and analyze the features that need to be transformed. Non-Java applications do not support data collection.

Context

The client for collecting application information consists of the following modules:

- Agent for dynamic data collection: collects basic information of the database that is requested by the application when the application is run. The basic information includes the SQL statements, schemas, and call stacks of requests. System information, performance information, and SQL hotspots are also collected.

- Collector for centralized data collection: collects, desensitizes, and processes data from the Agent.

Background information

- Collected SQL statements are masked. The request parameters and returned values of SQL statements are not collected.
- Data collection is read-only to prevent intrusions to applications.
- Data collection is automatically suspended during peak hours to restrict the memory usage within a specified range.
- Dynamic data collection of Java applications in Tomcat, JBoss, and Oracle WebLogic containers that use JDK 1.6 or later is supported.

Download Application Collector

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Application Evaluation and Transformation.
3. In the **Collect Application Data** step, click **Download Database Collector** in the lower-left corner.
4. In the **Download Application Collector** panel, click **Download Application Collector**.

Usage notes

- Basic technical knowledge is required before you deploy the client. Make sure that the client is deployed by Java developers.
- SUN JDK, Oracle JDK, and OpenJDK 1.6 or later are supported. IBM JDK is not supported.
- The decompressed client package contains the collector and javaagent directories. The Collector is deployed independently on a server that does not have online applications to prevent online applications from being affected during data processing. The javaagent directory is copied to the application server to be monitored and deployed with applications to collect data.
- Make sure that the Collector and Agent have system operation permissions.

  > **Note** If you are using the Unix or Linux operating system, you must run the `chmod -R 775 collector/` command to add system operation permissions to the directories.

- The Collector acts as a server and can be deployed with 1 to 20 Agents. An Agent is deployed on a single application server.

  > **Note** You can deploy Agents on a few machines of a distributed application based on load balancing.

- Deploy the Collector, and then deploy the Agent. The application and the Collector must be interconnected for centralized data masking. The machine where the Collector is deployed must use JDK 1.6 or later and have a Java Virtual Machine (JVM) memory size of more than 4 GB. The disk volume is determined by the number of monitored applications, monitoring duration, activities, and the number and sizes of SQL statements. If no explosive growth of data happens, you can estimate the data volume based on the data collected within half a day. In most cases, the data volume of a monitored application is less than 1 GB within seven days.
- The application must be deployed on a server that uses JDK 1.6 or later. The JVM heap size of the

application must be at least 300 MB. Supported containers include Tomcat, JBoss, and WebLogic. Docker images can be deployed to a Container Service for Kubernetes (ACK) cluster.

- The Agent monitors the SQL statements and call stacks that are used to access Oracle databases. Make sure that all the operations are monitored during the monitoring period of the Agent. The application must be monitored when recurring tasks are running. Otherwise, the collected data is incomplete.

What to do next

Deploy a data collection environment.

6.5.4.3. Deploy a data collection environment

This topic describes how to deploy a data collection environment to collect data.

Run a precheck

- Make sure that the collector is deployed on a standalone server that has no online applications.
- Make sure that JAVA_HOME and JDK 1.6 or later have been configured.

Start the collector

- If you use a UNIX-based server, run the following command in the directory in which the collector is installed:

```
./run.sh
```

- If you use a Windows-based server, run the following command in the directory in which the collector is installed:

```
start /b java -jar javaagent-collector.jar
```

View the log file in the `collector/logs/collector.log` directory. If the log file contains a startup or success message, the collector is deployed.

Configure and start the Java agent

Configurations before the startup

- Make sure that the JAVA_HOME environment variable has been configured. Otherwise, set the value of JAVA_HOME in the attach.sh file to the absolute path of the Java runtime environment (JRE). If you use JRE instead of Java Development Kit (JDK), you must manually copy the tools.jar file to the ${JAVA_HOME}/lib/ directory.
- Configure the javaagent.config file: profiler.collector.ip = 11.23.45.67 # Specify the IP address of the collector. profiler.collector.port = 9996 # Specify the port of the collector. profiler.app.name = adamApp # Specify the name of the application. The name must be 1 to 20 characters in length and must contain letters and digits. profiler.app.port = 8080 # Specify the startup port of the application. Multiple ports are available for an application, and you need to specify only the startup port. Specify only one startup port regardless of the number of applications on a Java virtual machine (JVM). profiler.applicationservertype = TOMCAT # Specify the container type of the application middleware. Valid values: TOMCAT, JBOSS, and WEBLOGIC.

Optional configurations

Configure prefixes for the directories in which the Java code to be tested resides. Replace the directories in the following example. Separate multiple directories with commas (,). Each directory must have at least two levels of subdirectories. We recommend that you specify no more than five directories. If databases are changed, the collector can provide specific suggestions to modify application configurations based on the precise call stack information.

- If some directories cannot be provided, you can leave this parameter unspecified. You can configure filters on the analysis page of the Alibaba Cloud Advanced Database & Application Migration (ADAM) console.

```
profiler.classpath.whitelist = com.alibaba.javaagent,com.alibaba.adam
```

- If the preceding whitelist is specified, you can leave this parameter unspecified. You can also specify a blacklist to filter out unnecessary call stack information.

```
profiler.classpath.blacklist =org.apache,net.sf
```

- Specify the limit of the CPU utilization. If the limit is reached, data collection is paused.

```
profiler.cpu.threshold = 85
```

- Specify the interval at which system information is collected. Unit: minutes. Default value: 15.

```
profiler.sys.send.interval = 15
```

- Specify the interval at which dynamic SQL information is collected. Unit: minutes. Default value: 15.

```
profiler.sql.dynamic.send.interval = 15
```

Startup methods

- Method 1: Monitor an application only once without restarting the application

  Add system operation permissions to the javaagent directory. Make sure that the account used to start the agent is the same as the account used to start the application. Run the following commands by replacing ${pid} with the process ID of the application:

  - If you use a UNIX-based server, run the following command:

    ```
    ./attach.sh -p ${pid}
    ```

  - If you use a Windows-based server, run the following command:

    ```
    java -cp "%JAVA_HOME%\lib\tools.jar;%cd%\javaagent-bootstrap.jar" com.alibaba.adam.javaagent.bootstrap.AgentAttacher -p ${pid}
    ```

  If the log file in the javaagent directory contains a success message, the agent has been started. If inbound traffic is received by the application and the collector directory contains a subdirectory that contains data, the agent has been started and has sent data to the collector.

  - Make sure that the obtained process ID of the application is correct.
  - Make sure that the account used to start the agent is the same as the account used to start the application. Make sure that the agent and the application have identical system operation permissions. Otherwise, the agent cannot monitor the application.

    > **Note** If the application is started by viewing Windows Registry as a system user, the agent cannot be started by using Method 1 due to inconsistent account permissions.

- Method 2: (Recommended) Start monitoring when an application starts

  Replace ${javaagent_path} with the javaagent directory in the configuration file, and then restart the application.

  - UNIX-based server
    - Tomcat: Add the following code to the last CATALINA_OPTS configuration item in the catalina.sh startup file:

      ```
      CATALINA_OPTS="$CATALINA_OPTS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - JBoss: Add the following code to the last JAVA_OPTS configuration item in the run.conf startup file:

      ```
      JAVA_OPTS="$JAVA_OPTS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - WebLogic: Add the following code to the last JAVA_OPTIONS configuration item in the startWebLogic.sh startup file:

      ```
      JAVA_OPTIONS="$JAVA_OPTIONS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - WebSphere:
      - Method 1: Create a configuration file and add the following code to the configuration file:

        ```
        JAVA_OPTS="$JAVA_OPTS -javaagent:/home/admin/javaagent/javaagent-bootstrap.jar"
        ```

      - Method 2: Add the following code to the common JVM parameters:

        ```
        name:javaagent  value:/home/admin/javaagent/javaagent-bootstrap.jar
        ```

    - Docker container deployed in a Kubernetes cluster: Place the javaagent directory into a Docker image, add the corresponding javaagent configuration file, and then deploy the image.

  - Windows-based server
    - Tomcat: Add the following code to the last CATALINA_OPTS configuration item in the catalina.bat startup file:

      ```
      CATALINA_OPTS="$CATALINA_OPTS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - JBoss: Add the following code to the last JAVA_OPTS configuration item in the run.conf startup file:

      ```
      JAVA_OPTS="$JAVA_OPTS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - WebLogic: Add the following code to the last JAVA_OPTIONS configuration item in the startWebLogic.cmd startup file:

      ```
      JAVA_OPTIONS="$JAVA_OPTIONS -javaagent:${javaagent_path}/javaagent-bootstrap.jar"
      ```

    - WebSphere:
      - Method 1: Create a configuration file and add the following code to the configuration file:

        ```
        JAVA_OPTS="$JAVA_OPTS -javaagent:/home/admin/javaagent/javaagent-bootstrap.jar"
        ```

      - Method 2: Add the following code to the common JVM parameters:

        ```
        name:javaagent  value:/home/admin/javaagent/javaagent-bootstrap.jar
        ```

After the application is started, view the application logs. If the `Java Agent load successfully!` message is displayed, the agent is started. If an error occurs, modify the script based on the error message. If no agent information is displayed, the agent was not loaded because the configuration path is invalid.

Method comparison

Startup method	Benefit	Drawback	Scenario
Method 1	You can configure the agent without restarting the application even if the application has never been monitored by the agent.	You must manually start the agent each time you restart the application. The account used to start the agent must be the same as the account used to start the application.	This method applies to scenarios in which you will not restart your applications for a long time and an application restart affects your business.
Method 2	When you restart the application, the agent automatically starts and continuously collects application data. You do not need to consider account permissions.	When you collect application data for the first time, you must restart the application to start the agent.	This method applies to scenarios in which you can restart your applications to continuously collect data without affecting your business.

Usage notes

- When the application stops, the agent also stops regardless of which method was used to start the agent.
- You can stop the agent in one of the following ways:
  - If the agent is started by using Method 1, restart the application or run the ./attach.sh -p ${pid} -s command.
  - If the agent is started by using Method 2, remove the configurations of the Java agent and restart the application.
- If you want to monitor the application after the agent has been stopped, you must restart the application.
  - If the agent is started by using Method 1, start the agent manually after the application restarts.
  - If the agent is started by using Method 2, wait for the agent to automatically start after the application restarts.
- JBoss clusters are divided into Community Edition clusters and Enterprise Edition clusters. If the jboss.modules.system.pkgs configuration item exists, you must add the com.alibaba.adam.javaagent directory and restart the application to validate the monitoring settings regardless whether you use Method 1 or 2.

Collect data

Go to the directory in which the collector is installed, compress the files in the data/ directory in the ZIP format by application name, and then separately upload each application profile. Each application corresponds to one profile. You can compress the data collected for an application from different IP addresses into one ZIP file. However, you cannot compress data collected for different applications into one ZIP file. Otherwise, an upload error occurs.

> **Note** In most cases, data of one to seven days is collected.

What to do next

Upload an application profile.

6.5.4.4. Create an application profile

Advanced Database & Application Migration (ADAM) allows you to use application profiling to analyze the data collected from a single application by using intelligent analysis algorithms.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Application Evaluation and Transformation.
3. Click the **Create Application Profile** tab.
4. On the Create Application Profile tab, click **Create Profile**.
5. On the Basic Settings tab of the **Create Application Profile** panel, set the **Profile Name**, **Type**, and **Report Language Type** parameters and click **Upload** to upload dynamic data files.
6. Click the **Database Profile** tab and click **Select**.
7. In the **Select Associated Database Profile** panel, select a database profile, click **Add**, and then click the icon in the upper-right corner of the panel.
8. Select the added database profile and click **Create**. The application profile is created based on intelligent analysis results of data files. This process takes 1 minute to 10 minutes.

View the application profile

An application profile consists of the following four parts:

- Summary
  - Complexity is calculated based on conditions such as application scenarios and database features. This dimension indicates the application usage. A higher score indicates a more complex application that requires more transformations.
  - Session indicates the connection status of the application. A higher score indicates a larger number of connections to the application. This dimension is significant for configuring the connection pool when you transform the application.
  - Risk indicates the potential performance bottlenecks and stability risks of the application, especially the performance risks of SQL operations.
  - Hotspot indicates whether the database contains frequently accessed objects. A higher score indicates that the database to access contains objects that are frequently accessed.

- Scale indicates the number of deployment units or instances of the application.
- Load indicates the running performance of the application.

- System Information

  The System Information tab displays the system parameters that are collected by ADAM Application Collector to help you evaluate the status of the application.

- Object Overview

  The Object Overview tab displays information about the SQL statements and database objects that the application collects and analyzes.

- Object Details

  The Object Details tab displays the relationships among database objects, SQL statements, and application code that are analyzed by ADAM. The section on the left side of the tab lists the database objects accessed by the application in the form of a tree diagram that uses schema and object type as dimensions. The section on the right side of the tab lists database objects and the corresponding SQL statements used to access the objects.

  You can also click the **Call Stack Settings** tab to configure a blacklist of call stacks. This way, you can filter out the information of call stacks that you do not need.

What to do next

Evaluate applications

6.5.4.5. Evaluate applications

After you create application profiles from all the collection packages, you can create an application evaluation project.

Create an application evaluation project

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Application Evaluation and Transformation.
3. Click the **Evaluation Application** tab.
4. Click **Create Application Evaluation Project**.
5. In the **Create Application Evaluation Project** panel, set the **Project Name**, **Project Type**, **Destination Database Version**, and Report Language parameters in the **Basic Information** step, and then click **Next: Select Application**.
6. Select one or more application profiles that you want to evaluate and click **Next: Select Database Evaluation Project**.
7. Select a database evaluation project that has been completed and click **Create and Start Evaluation Project**.

Application evaluation details

Application evaluation details contain the evaluation results, project overview, and joint profiles.

- **Evaluation Result**: This metric shows the evaluation results of applications.

- **Migration Overview**: This metric shows the overall migration and transformation of applications and databases.
  - **Migration Score**: ADAM scores the difficulty of migration and transformation tasks. A higher score indicates a lower transformation cost for applications.

    > **Note** The migration score is subject to the integrity of collected data. We recommend that you estimate the migration cost based on your business needs.

  - **Overall Compatibility**: This metric shows the compatibility of application SQL statements and database objects. SQL statements collected from databases are subject to the database system. Therefore, the compatibility of SQL statements is not used for reference.
  - **Application Transformation item**: This metric shows the number of application transformation items.
  - **DB Transformation**: This metric shows the number of database transformation items.
- **Overall Compatibility**: This metric indicates the compatibility of database objects, database SQL statements, and application SQL statements.
- **Architecture Blueprint**: The architecture blueprint shows the status of migration groups in a topology.

- **Project Summary**: This metric shows the basic project information.
- **Joint Profiling**: This metric shows the relationships between applications and databases.

6.5.4.6. Perform static application transformation

Advanced Database & Application Migration (ADAM) allows you to scan static code to identify the SQL statements that need to be transformed. ADAM transforms the SQL statements that can be automatically replaced, or sends notifications to you if some SQL statements cannot be automatically replaced.

Prerequisites

The type of the source database is Oracle or Db2 for LUW.

Create a transformation project

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Application Evaluation and Transformation.
3. Click the **Transform Applications** tab.
4. On the Transform Applications tab, click **Create Transformation Project**.
5. In the **Create Transformation Project** panel, set the **Source Database Type**, **Destination Database Type**, **Destination Database Version**, and **Architecture** parameters and click **Upload** to upload data files. Then, click **Create**.

   > **Note** When you create a transformation project, the built-in analysis program of ADAM is automatically run. This process takes 1 minute to 10 minutes.

Details of static application transformation

The details of the static application transformation include the following sections in the console: **Project Summary**, **Transformation Overview**, and **Transform Applications**.

- Project Summary: displays the basic information of the transformation project.
  - **File Name**
  - **Source Database Type**
  - **Destination Database Type**
  - **Destination Database Version**
  - **Architecture**
  - **Code Blocks**
- Transformation Overview: displays SQL statements and script control statements in charts and tables.
  - **Not Required**: The code blocks run in the destination database without transformation.
  - **Automatic**: The code blocks are transformed by ADAM. You need to only replace the original code blocks with the transformed code blocks in the destination database.
  - **Manual**: The transformation requirements of the code blocks are listed. You must manually transform these code blocks.
  - **SQL Unrecognized**: The SQL statements in the code blocks cannot be identified because the SQL statements are invalid or code blocks are run in a special way.
- Transform Applications: Click **Details** in the **Actions** column to view the details of code block transformation.

6.5.5. Migration lab

6.5.5.1. Periodically collect information about SQL statements

This topic describes how to periodically and continuously collect information about SQL statements from Oracle databases.

Context

The periodic SQL collection feature allows you to continuously collect information about SQL statements from Oracle databases based on a custom collection interval and automatically merges the information about SQL statements.

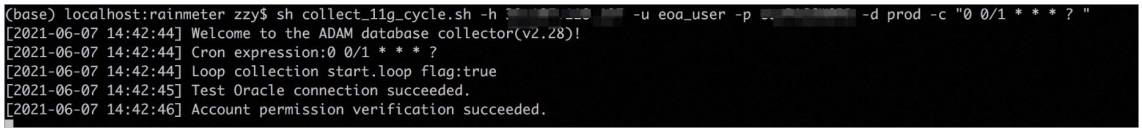
> **Note** You can periodically collect information about SQL statements only by using an offline collector.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Migration Lab.
3. In the **Periodic SQL Collection** section, click **Download** to download the collector and account information.

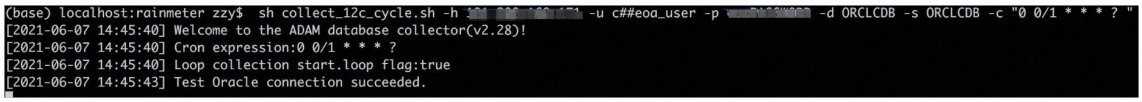
4. Start collection.
   - If the version of the source Oracle database is 11g or earlier, run the following command:

     ```
     sh collect_11g_cycle.sh -h <ip> -u <username> -p <password> -d <service_name> -c <cro
     n>
     ```

   - If the source Oracle database is a container database (CDB) and its version is 12c or later, run the following command:

     ```
     sh collect_12c_cycle.sh -h <ip> -u <username> -p <password> -d <service_name> -s <sid
     > -c <cron>
     ```

     Note Parameter description:
     - -h: the IP address of the source database.
     - -u: the name of the account that is used for collection. In this example, the account name is eoa_user.
     - -p: the password of the account. In this example, the password is eoaPASSW0RD.
     - -P: the port that you want to use to collect the information about the source database. Example: 1521.
     - -d: the service name of the source database. If the source database is an Oracle 12c database, specify the service name of the specified pluggable database (PDB).
     - -s: the name of the Oracle database instance. This option is required only for Oracle 12c.
     - -c: the CRON expression that is used to specify the collection interval.

5. Return to the Migration Lab page. In the **Periodic SQL Collection** section, click **Details**.
6. Click **Create Project**.
7. In the **Create Periodic Collection Project** panel, enter the project name, select the database type from the drop-down list, and then click **Upload** to upload the out/data.zip file. After the file is uploaded, click Create.
8. Find the project that you create in the previous step, and click **Details** in the **Actions** column to view the collection results.

6.5.5.2. Perform an SQL comparison test

This topic describes how to use the SQL comparison test platform provided by ADAM.

Context

The SQL comparison test platform is a database test platform built based on the JMeter test engine. All SQL statements retained during the use of the ADAM service can be tested in the comparison test platform. You can build SQL test sets, create and execute test tasks, and view test results.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, choose **DTS**.
3. In the left-side navigation pane, choose Heterogeneous Database Migration(ADAM) > Migration Lab. In the **SQL Comparison Test** section, click **Test**.
4. Click **Create SQL Test Set**.
5. In the **Create SQL Test Set** panel, enter a name for the test set and set the parameters.
   i. After you configure the source database information, click **Test Connectivity** to test the connectivity of the source database.

Parameter	Description
Database Type	The type of the source database .
Instance Region	The region that is closest to the geographic area in which the source database resides.
Instance Type	The type of the source database. By default, **Database with Public IP Address** is selected.
Hostname or IP Address	The IP address of the server that hosts the source database.
Source Database Identifier	The identifier of the source database. You can specify the identifier in one of the following formats: ■ Service name ■ SID
Database Name	The name of the source database.
Port	The port number of the source database.
Database Account	The username that is used to log on to the source database.
Database Password	The password that is used to log on to the source database.

ii. After you configure the destination database information, click **Test Connectivity** to test the connectivity of the destination database.

Parameter	Description
Database Type	The type of the destination database.
Instance Region	The region that is closest to the geographic area in which the destination database resides.
Instance Type	The type of the destination database. By default, **Database with Public IP Address** is selected.
Hostname or IP Address	The IP address of the server that hosts the destination database.
Database Name	The name of the destination database.
Port	The port number of the destination database.
Database Account	The username that is used to log on to the destination database.
Database Password	The password that is used to log on to the destination database.

- Database Name: Enter the name of the destination database.
- Port: Enter the port number of the destination database.
- Database Account: Enter the username that is used to log on to the destination database.
- Database Password: Enter the password that is used to log on to the destination database.

iii. Select the database evaluation project. The database evaluation project is created in the Evaluate Dest. DB wizard on the Database Evaluation page.

iv. Specify the mode of SQL statement collection. Select Periodic Collection and Adapter.

v. Click **Create**.

6. Create an SQL test task.

i. On the **SQL Test Sets** page, find the test set and click **Details** in the **Actions** column to view the SQL statements in the test set.

ii. On the **SQL Test Set Details** page, select the SQL statements on which you want to perform a comparison test. Check whether the SQL statements are compatible based on the information in the **Compatible** column. Do not select incompatible SQL statements.

iii. Click **Create SQL Test Task**. In the dialog box that appears, enter the task name, number of threads, and number of times that a loop needs to be repeated, and set the Test Type parameter to **Comparison Test**. Click **Create**.

> **Note** The maximum number of threads allowed is 10,000, and the maximum number of times that a loop can be repeated is 10,000. If a specified number exceeds the corresponding limit, the task fails to be created.

7. Execute the SQL test task. Find the created SQL test task and click **Start** in the **Actions** column. The task enters the **Running** state.

After the task is complete, the state of the task changes to **Successful** or **Failed**.

8. View the results.

   Click **Details** to view the results of the task.

6.5.5.3. Use ADAM SQL Adapter to transform SQL statements

Advanced Database & Application Migration (ADAM) SQL Adapter can automatically transform SQL statements that are incompatible between PolarDB databases and Oracle databases. This reduces the cost of SQL statement transformation after applications are migrated to PolarDB databases. This topic describes how to use SQL Adapter.

Overview

- SQL Adapter can quickly verify PolarDB features without modifying the business code.
- SQL Adapter can automatically detect all the SQL statements of an application and provide transformation suggestions.
- For SQL statements that cannot be automatically transformed, SQL Adapter can verify the correctness of these SQL statements.

Deploy the environment

Before SQL Adapter is deployed, make sure that ADAM V5.0 is installed.

1. Log on to the DMS console.
2. In the top navigation bar, click DTS. In the left-side navigation pane, click Migration Lab.
3. In the **ADAM SQL Adapter** section, click **Apply**.

   After the package is downloaded, run the following command to decompress the package:

   ```
   tar -xvf adam-adapter.tar.gz
   ```

4. Configure the source database and destination PolarDB database.

```
# 参考配置
#adam.database.url=jdbc:mariadb://Adam.mysql.rds.aliyuncs.com/adam_studio_saas
#adam.database.username=123
#adam.database.password=123

#target.database.1.url =jdbc:polardb://Adam.o.polardb.rds.aliyuncs.com:1521/adam
#target.database.1.user =123
#target.database.1.password=123
#target.database.1.currentSchema=123

adam.database.url=jdbc:mariadb://
adam.database.username=
adam.database.password=

target.database.1.url =jdbc:polardb://
target.database.1.user =
target.database.1.password=
target.database.1.currentSchema=
```

> **Note** If multiple accounts are used to log on to the application that you want to migrate, you can use target.databases.2 to add configuration items.

5. Start SQL Adapter by running the following command:

```
sh run.sh start
```

6. (Optional) Restart SQL Adapter by running the following command:

```
sh run.sh restart
```

7. (Optional) Stop SQL Adapter by running the following command:

```
sh run.sh stop
```

8. (Optional) Customize the startup script.
   - The default port number of the startup script is 8888. You can modify the port number.
   - The type parameter specifies the mode in which custom rules take effect. The default value is CACHE, which indicates that all rules are loaded after SQL Adapter starts. However, SQL Adapter must be restarted if new rules are customized.
   - In RUNTIME mode, rules are loaded in real time when SQL Adapter performs the transformation. This may affect the efficiency of executing SQL statements.

```
curr_dir=$(pwd)
port=8888
# type is CACHE/Runtime
type=CACHE
```

Procedure

1. Transform the code of your application.

   Modify the Java Database Connectivity (JDBC) driver used by the application and change the connection URL of JDBC to the URL of SQL Adapter. SQL Adapter does not process the password, so the username and password do not need to be modified.
2. Log on to the DMS console.
3. In the top navigation bar, click DTS. In the left-side navigation pane, click Migration Lab.
4. Click **Details** to view the SQL statements that need to be transformed.

   Compatibility types include incompatible, compatible, and compatible after modification. SQL statements that are incompatible or compatible after modification must be transformed.
5. Click **Custom Rules**. In the Custom Rules panel, set the parameters to customize a rule.

   The following table describes the parameters.

Parameter	Description
Rule Type	○ Text Replacement: specifies the text to be matched and replaced. ○ Regular Expression Replacement: matches the text to be replaced by using a regular expression.
Text to Match	The text to be matched and replaced, or the regular expression used to match the text to be replaced.
Text After Replacement	The text to replace after matching.
Effective Scope	○ Global: The rule applies to all SQL statements. ○ Specified SQL: The rule applies only to the specified SQL statements.

6. Click **Custom Rule List** to view the customized rules.

Usage notes

- The port number of SQL Adapter must be available so that applications can access SQL Adapter.
- By default, the type parameter is set to CACHE. If you need to modify your application while SQL Adapter is running, we recommend that you set the parameter to RUNTIME.
- SQL Adapter records all SQL statements that have been transformed and migrated. However, the same SQL statements are recorded only once during transformation.
- Custom rules do not take effect on CALL statements.

6.5.6. SQL conversion

The SQL converter provided by the Advanced Database & Application Migration (ADAM) module allows you to convert SQL scripts used for Oracle, Teradata, or Db2 databases to SQL scripts supported by MySQL, PolarDB for Oracle, or AnalyticDB for PostgreSQL databases.

Procedure

1. Log on to the DMS console.

2. In the top navigation bar, click DTS. In the left-side navigation pane, click SQL Conversion.
3. Select the type of the source SQL script.

   Supported types of source SQL scripts are Oracle, Teradata, and Db2.
4. Select the type of the converted SQL script.

   Supported types of converted SQL scripts are MySQL, PolarDB for Oracle, and AnalyticDB for PostgreSQL.
5. Enter scripts that conform to the specifications of the source SQL script type in the field on the left side of the page.
6. Click **Convert**.

   On the right side of the page, view the converted SQL scripts.

7.Security management

7.1. Apply for permissions

You can apply for the query, change, and export permissions on a database, table, or column. After the database owner approves your application, you can query, change, and export data.

Permissions

- Query permissions: the permissions to execute SQL statements in the SQL Console to query the data of the object on which you want to apply for the permissions.
- Change permissions: the permissions to submit tickets to change data or synchronize data in a database or table. You cannot change data without approval.
- Export permissions: the permissions to submit tickets to export data from the object on which you want to apply for the permissions. You cannot export data without approval.

Permission categories that are supported by different control mode

Permission category	Permissions	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Instance logon	After you obtain the instance logon permissions on an instance, you can use the preset database account and password to log on to the instance.	✓	✓	×
Database	Database permissions are classified into query, export, and change permissions. After you obtain the database permissions on a database, you can access the following resources of the database: 1. All the insensitive fields. 2. All the tables to which row-level control settings are not applied. 3. All new tables. • Query permissions: You can execute SQL statements in the SQL Console to query data. • Change permissions: You can submit data change and data import tickets. • Export permissions: You can submit data export tickets.	×	×	✓

Permission category	Permissions	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Table	Table permissions are classified into query, export, and change permissions. After you obtain the table permissions on a table, you can access all data in the table except sensitive fields. • Query permissions: You can execute SQL statements in the SQL Console to query data. • Change permissions: You can submit data change and data import tickets. • Export permissions: You can submit data export tickets.	×	×	✓
Sensitive field	Sensitive field permissions are classified into query, export, and change permissions. After you obtain the sensitive field permissions on sensitive fields in a table, you can access all sensitive fields in the table. Before you apply for the sensitive field permissions, you must obtain the database and table permissions to which the sensitive fields belong. • Query permissions: You can execute SQL statements in the SQL Console to query data. • Change permissions: You can submit data change and data import tickets. • Export permissions: You can submit data export tickets.	×	×	✓
Database owner	• The owner of a database can manage the permissions on the database. For example, the owner of a database can grant or revoke permissions on the database and tables in the database. • The owner of a database can query all data in the database except sensitive or confidential fields. The owner can also submit tickets to perform operations on the data and schemas in the database without the need to apply for permissions. • DMS automatically identifies and assigns database owners to owner nodes in approval processes.	✓	✓	✓

Permission category	Permissions	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Table owner	• The owner of a table can manage the permissions on the table. For example, the owner can grant or revoke permissions on the table. • The owner of a table can query all data in the table except sensitive or confidential fields.	✓	✓	✓
Programmable object	Programmable object permissions are classified into query, export, and change permissions. • Query permissions: You can execute SQL statements in the SQL Console to query data. • Change permissions: You can submit data change and data import tickets. • Export permissions: You can submit data export tickets.	×	×	×
Instance performance	You can apply for permissions to view the performance of instances that are managed in Security Collaboration mode.	×	×	✓
Instance owner	• The owner of an instance can manage the permissions on the instance. For example, the owner of an instance can grant or revoke permissions on the instance. • The owner of an instance can query all data in the databases of the instance except sensitive or confidential fields. The owner can also submit tickets to perform operations on the data and schemas of the instance without the need to apply for permissions.	✓	✓	✓

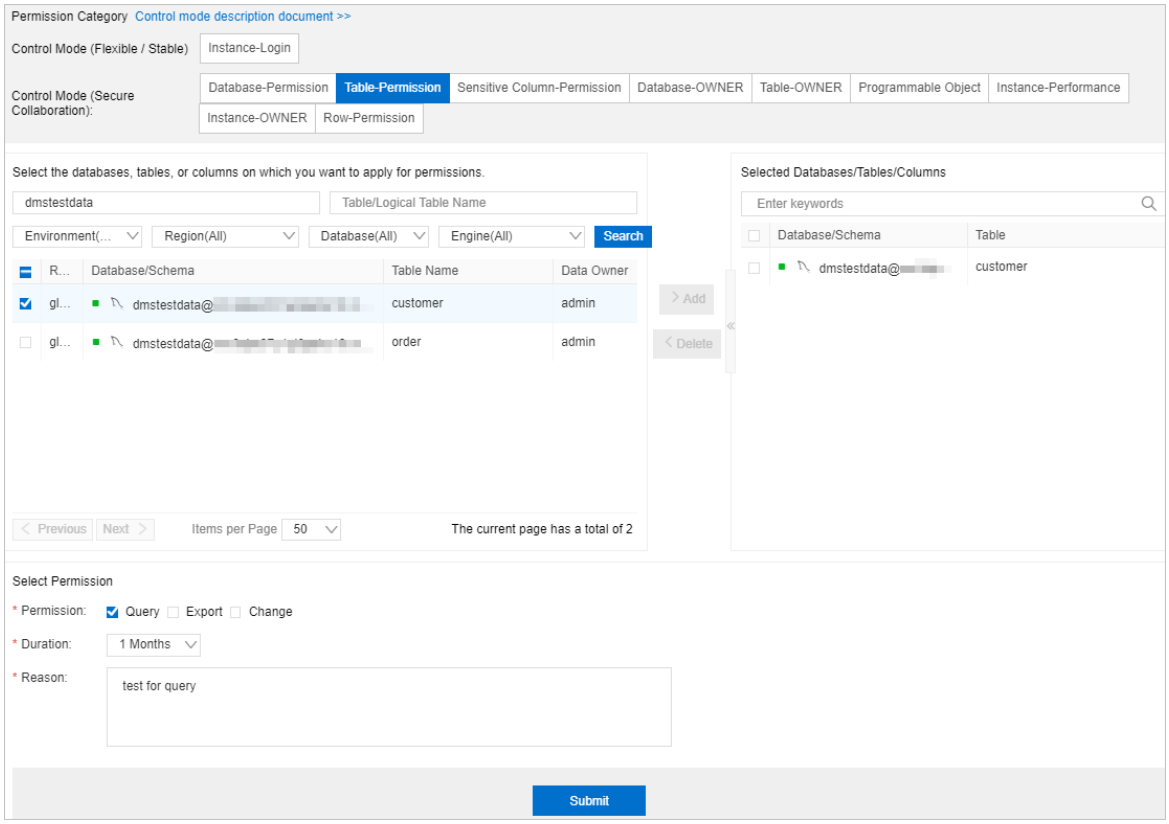
Permission category	Permissions	Control mode		
		Flexible Management	Stable Change	Security Collaboration
Row	Row permissions are classified into query, export, and change permissions. You can apply for permissions on specific values of a managed field in a table. You can also apply for permissions on all values of a managed field in a table. • Query permissions: You can execute SQL statements in the SQL Console to query data. • Change permissions: You can submit data change and data import tickets. • Export permissions: You can submit data export tickets.	×	×	×

Apply for permissions

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Permission**.

   > **Note** In the top search box on the Permission page, you can search for databases or tables by name. In the search results, find the required database or table and click **Access apply** in the **Actions** column.

3. Click **Access apply** and select a permission category. For more information about different permission categories, see the Permission categories that are supported by different control mode section of this topic.
4. Set the required parameters of the permissions for which you want to apply.



i. Add objects on which you want to apply for permissions, such as databases, tables, or columns.

You can enter a keyword to search for a database name or table name, select the objects on which you want to apply for permissions, and then click **Add** to add the objects to the **Selected Databases/Tables/Columns** section.

> **Note** The keywords that you enter can contain percent signs (%) as wildcards.

ii. Select the permissions for which you want to apply and specify the duration for which you want to have the permissions. Then, enter the reason for your application.

5. Click **Submit** and wait for approval.

> **Note** You can view the status of application ticket in the My Tickets section on the Home page of the **DMS console**.

Manage permissions

Management type	Operation	Description
Active management	Release permissions	On the Home page of the DMS console, click **Accessible Assets**. Select the object on which you want to release permissions and click **Release Permission**.
Passive management	N/A	The owner of a database can view and assess the rationality of permission assignments at any time and manage the permissions that are granted to users.

> **Note** All the operations that you perform to apply for, release, revoke, and grant permissions are recorded in operation logs. To view the operation logs, click **Security and Specifications** in the top navigation bar. Then, select **Operation Audit** in the left-side pane and click the Operation Logs tab.

7.2. Security rules

7.2.1. Manage security rules

Security rules are implemented by using a collection of domain-specific languages (DSLs) to control user access to databases based on several factors. These factors include the type of database, the syntax of database operations, and the number of affected rows. You can use security rules to standardize database operations, development processes, and approval processes as required. This topic describes how to manage security rules.

Prerequisites

You are a DBA or a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Security Rules**.
3. Perform one of the following operations based on your business requirements:
   - Create a rule set

     Click **Create Rule Set**. In the Create Rule Set dialog box, set the Engine Type, Rule Set Name, and Remarks parameters, and click **Submit**.
   - Edit a rule set
     a. Find the required rule set and click **Edit** in the **Actions** column of the rule set.
     b. In the left-side pane, click the required rule subset, such as **SQL Console**. On the right side of the page, select a checkpoint.
     c. Find the required rule and click **Edit** next to the rule. For more information about the related syntax, see DSL syntax for security rules.

     > **Note** You can also delete or disable a rule.
   - Create a similar rule set
     a. Find the required rule set and click **Create As** in the **Actions** column of the rule set.
     b. In the dialog box that appears, enter a name and a description for the new rule set.
     c. Click **Submit**. The system copies the configurations of the original rule set to the new rule set.
   - Delete a rule set

     Find the required rule set and click **Delete** in the **Actions** column of the rule set. In the message that appears, click **OK**.

> **Note**
> - A deleted rule set cannot be recovered. Proceed with caution.
> - You can delete only custom rule sets. You cannot delete built-in rule sets.

- Set a rule set as the default rule set

  Find the required rule set and click **Set as Default** next to the rule set. In the message that appears, click **OK**. The rule set is used as the default rule set for the related database engine.

7.2.2. DSL syntax for security rules

DMS provides a domain-specific language (DSL) to describe security rules. You can use the DSL syntax to define security rules. This allows you to define database development standards based on your business requirements.

Overview

The DSL syntax can include one or more conditions and related actions that are specified by an IF-ELSE statement.

> **Note** The if clause is required. Zero or more elseif clauses can be specified. Zero or one else clause can be specified.

Example 1: If Condition 1 is met, DMS performs Action 1.

```
if
    Condition 1
then
    Action 1
end
```

Example 2: If Condition 1 is met, DMS performs Action 1. If Condition 2 is met, DMS performs Action 2. If Condition 1 and Condition 2 are not met, DMS performs Action 3.

> **Note** If the `else Action 3` clause is removed and Condition 1 and Condition 2 are not met, DMS performs no action.

```
if
    Condition 1
then
    Action 1
elseif
    Condition 2
then
    Action 2
[else Action 3]
end
```

DSL syntax

- Conditional clauses

  DMS uses conditional clauses to evaluate whether to perform actions. The result of a conditional clause is true or false. A conditional clause consists of one or more connectors, operators, and factors. Connectors include AND and OR. Factors are predefined system variables. The following examples are valid conditional clauses:

```
1. true                    // This is the simplest conditional clause. The result is true
.
2. 1 > 0
3. 1 > 0 and 2 > 1
4. 1 <= 0 or 1 == 1
```

- Connectors

  Connectors include AND and OR. The AND connector has higher priority than the OR connector. The two connectors have lower priority than operators. For example, a conditional clause is `1 <= 0 or 1 == 1`. DMS evaluates the result of the `1 <= 0` expression and the result of the `1 == 1` expression. Then, DMS evaluates the result of the OR expression based on the preceding results.

- Operators

  Operators are used to connect factors and constants to perform logical operations. The following table describes the operators that are supported by DMS.

Operator	Description	Examples
==	Evaluates whether a value is equal to another value.	1 == 1
!=	Evaluates whether a value is not equal to another value.	1 != 2
>	Evaluates whether a value is greater than another value.	1 > 2
>=	Evaluates whether a value is greater than or equal to another value.	1 >= 2
<	Evaluates whether a value is less than another value.	1 < 2
<=	Evaluates whether a value is less than or equal to another value.	1 <= 2
in	Evaluates whether a value belongs to an array of values.	'a' in ['a', 'b', 'c']
not in	Evaluates whether a value does not belong to an array of values.	'a' not in ['a', 'b', 'c']
matchs	Evaluates whether a string matches a regular expression.	'idxaa' matchs 'idx\w+'

Operator	Description	Examples
not matchs	Evaluates whether a string does not match a regular expression.	'idxaa' not matchs 'idx\w+'
isBlank	Evaluates whether a value is empty.	'' isBlank
isNotBlank	Evaluates whether a value is not empty.	'' isNotBlank

> **Note**
>
> - If you need to use a backslash (\) in a regular expression, you must add another backslash (\) as an escape character before the backslash that you want to use. For example, if you want to write the `idx_\w+` expression, you must enter `idx_\\w+`.
> - If a conditional clause includes nested expressions, we recommend that you enclose the required expressions in parentheses (). For example, a conditional clause is `1 <= 2 == true`. To specify the priority, you can change the clause to `(1 <= 2) == true`. DMS first evaluates the result of the `1 <= 2` expression in the parentheses.

- Factors

  A factor is a predefined variable in DMS. You can use factors to obtain the context to be validated by security rules. The context includes command categories and the number of affected rows. A factor name is prefixed by `@fac.`.Each tab of the Security Rules tab includes different factors for different checkpoints. The following table describes the factors that are supported by DMS.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.sql_type	The type of the SQL statement, for example, UPDATE or INSERT. For more information, see the SQL subcategories that are described in the "SQLConsole for relational databases" topic.
@fac.detail_type	The type of the data change. Valid values: ○ COMMON: a normal data modify ticket ○ CHUNK_DML: a lock-free data modify ticket ○ PROCEDURE: a programmable object ticket ○ CRON_CLEAR_DATA: a history data clean ticket ○ BIG_FILE: a large data import ticket
@fac.is_logic	A Boolean value that indicates whether the affected database is a logical database.
@fac.extra_info	Other information about the ticket. This factor is not in use.

Factor	Description
@fac.is_ignore_affect_rows	A Boolean value that indicates whether to skip the validation.
@fac.insert_rows	The number of data rows to be inserted.
@fac.update_delete_rows	The number of data rows to be updated.
@fac.max_alter_table_size	The size of the largest tablespace in which the table to be modified is stored.
@fac.is_has_security_column	A Boolean value that indicates whether sensitive fields are specified in the SQL statement to be executed.
@fac.security_column_list	The sensitive fields that are specified in the SQL statement to be executed.
@fac.risk_level	The risk level that is identified.
@fac.risk_reason	The reason based on which the operation is identified as this risk level.

Note You can use factors in conditional clauses. For example, you can write `@fac.sql_type == 'DML'` to evaluate whether an SQL statement is a DML statement.

- Action clauses

  An action indicates an operation that is performed when the if clause evaluates to true. For example, DMS can disable the submission of a ticket, select an approval process, approve a ticket, or reject a ticket. An action indicates the usage of a security rule. An action name is prefixed by `@act.` .Each tab of the Security Rules tab includes different actions for different checkpoints. The following table describes the actions that are supported by DMS.

Action	Description
@act.allow_submit	Requires the submission of SQL statements to be executed in a ticket.
@act.allow_execute_direct	Allows the execution of SQL statements in the SQLConsole.
@act.forbid_execute	Disables the execution of SQL statements.
@act.mark_risk	Marks the risk level of a data change. Example: `@act.mark_risk 'medium-level risk: online environment'` .

Action	Description
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

- Predefined functions

  DMS provides predefined functions that can be used in conditional clauses and action clauses. A function name is prefixed by `@fun.`.

Function	Description	Format
@fun.concat	Connects strings to form a single string. Output: a string. Input: multiple strings.	@fun.concat('d', 'm', 's') // The output is the string 'dms'. @fun.concat('[Development standards] The [', @fac.column_name, '] You must enter remarks.') // The output is a prompt that reminds the user who submits the ticket to enter a value in the field.
@fun.char_length	Calculates the length of a string. Output: an integer. Input: a string.	@fun.char_length('dms') // The output is 3. @fun.char_length(@fac.table_name) // The output is the length of the table name.
@fun.is_char_lower	Evaluates whether all the letters in a string are lowercase letters. Output: true or false. Input: a string.	@fun.is_char_lower('dms') // The output is true. @fun.is_char_lower(@fac.table_name) // If the output is true, it indicates that all the letters in the table name are lowercase.
@fun.is_char_upper	Evaluates whether all the letters in a string are uppercase letters. Output: true or false. Input: a string.	@fun.is_char_upper('dms') // The output is false. @fun.is_char_upper(@fac.table_name) // If all the letters in the table name are uppercase letters, the output is true.
@fun.array_size	Counts the number of values in an array. Output: an integer. Input: an array of values.	@fun.array_size([1, 2, 3]) // The output is 3. @fun.array_size(@fac.table_index_array) // The output is the number of indexes of the table.
@fun.add	Adds multiple numeric values. Output: a numeric value. Input: multiple numeric values.	@fun.add(1, 2, 3) // 6
@fun.sub	Deducts a numeric value from another numeric value. Output: a numeric value. Input: two numeric values.	@fun.sub(6, 1) // 5

Function	Description	Format

@fun.between	Evaluates whether a value belongs to a specific closed range. The supported data types are NUMERIC, DATE, and TIME. Output: true or false. Input: three values. The first value is the value to be evaluated. The second value indicates the lower limit. The third value indicates the upper limit.	@fun.between(1, 1, 3) // The output is true because the value 1 belongs to [1, 3].
		@fun.between(2, 1, 3) // The output is true because the value 2 belongs to [1, 3].
		@fun.between(7, 1, 3) // The output is false because the value 7 does not belong to [1, 3].
		@fun.between(@fac.export_rows, 2001, 100000) // If the number of exported rows belongs to [2001, 100000], the output is true.
		@fun.between(@fun.current_datetime(), '2019-10-31 00:00:00', '2019-11-04 00:00:00') // If the current date and time belong to [2019-10-31 00:00:00, 2019-11-04 00:00:00], the output is true.
		@fun.between(@fun.current_date(), '2019-10-31', '2019-11-04') // If the current date belongs to [2019-10-31, 2019-11-04], the output is true.
		@fun.between(@fun.current_time(), '13:30:00', '23:59:59') // If the current time belongs to [13:30:00, 23:59:59], the output is true.

@fun.current_datetime	Returns the current date and time, in the format of yyyy-MM-dd HH:mm:ss. Output: a string. Input: none.	@fun.current_datetime() // For example, the output is 2019-10-31 00:00:00.
@fun.current_date	Returns the current date, in the format of yyyy-MM-dd. Output: a string. Input: none.	@fun.current_date() // For example, the output is 2020-01-13.
@fun.current_time	Returns the current time, in the format of HH:mm:ss. Output: a string. Input: none.	@fun.current_time() // For example, the output is 19:43:20.

DSL configuration examples

Limit the number of SQL statements in a ticket: If the number of SQL statements in a ticket exceeds 1,000, DMS rejects the ticket and returns the related message.

```
if
    @fac.sql_count > 1000
then
    @act.reject_execute 'The number of SQL statements in a ticket cannot exceed 1,000.'
else
    @act.allow_execute
end
```

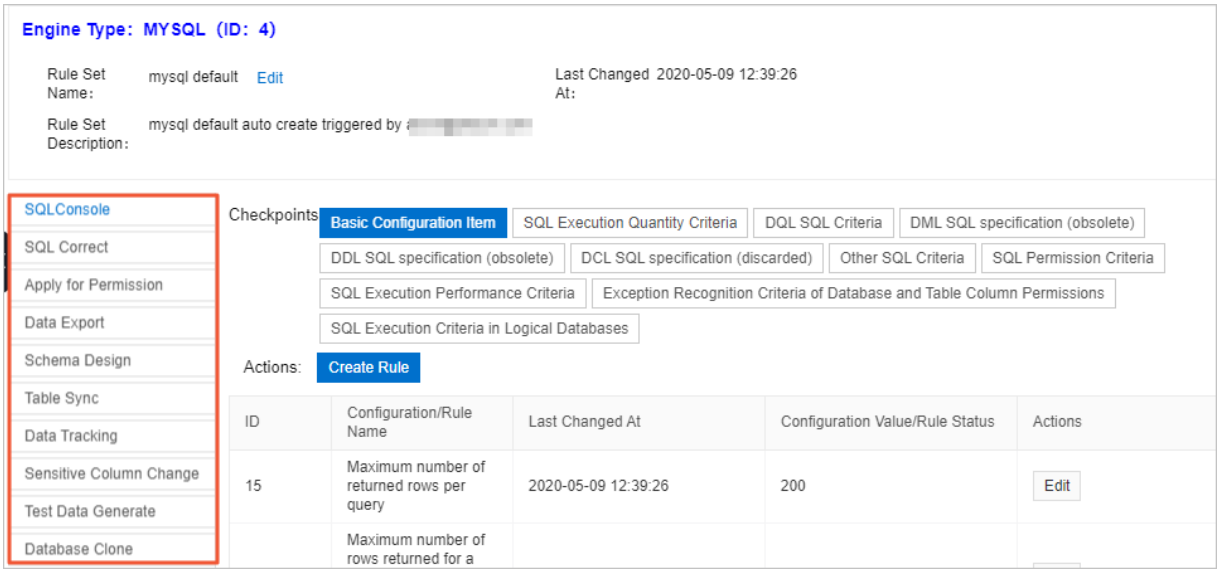
Allows the submission of only data manipulation language (DML) statements: If the SQL statements in a ticket are DML statements such as the UPDATE, DELETE, and INSERT statements, DMS allows the execution of the statements.

```
if
    @fac.sql_type in [ 'UPDATE','DELETE','INSERT','INSERT_SELECT']
then
    @act.allow_submit
end
```

7.2.3. Security rules

7.2.3.1. Overview of security rule sets

Security rule sets are implemented by using a collection of domain-specific languages (DSLs) to control user access to databases based on several factors. These factors include the type of databases, the syntax of database operations, and the number of affected rows. You can use security rule sets to standardize database operations, development processes, and approval processes as required.



This topic describes the features that are supported by security rule sets. You can click the link of a feature to view the information about the feature. The information includes the basic configuration items, checkpoints, factors, actions, and supported statements or commands.

- SQLConsole for relational databases
- SQLConsole for MongoDB
- SQLConsole for Redis
- Data change
- Permission application
- Data export
- Schema design
- Database and table synchronization
- Sensitive field change
- Test data generation
- Database cloning

7.2.3.2. Manage security rules under checkpoints

This topic describes how to configure a security rule under a checkpoint.

Procedure

In this example, a security rule set that is configured for MySQL databases is used.

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Security Rules**.
3. Find the security rule set that you want to manage and click **Edit** in the **Actions** column.
4. Click a tab on the left side of the page, and then click a checkpoint based on your business requirements.

   For example, click **SQL Console** on the left side of the page, and then click **SQL Execution Quantity Criteria**.

   > **Note**
   > - For more information about the tabs and checkpoints, see Overview of security rule sets.
   > - You can click **Create Rule** to create a security rule. For more information about the syntax, see DSL syntax for security rules.

5. Find the security rule that you want to manage and click **Edit** in the **Actions** column.

   > **Note** You can also click **Disable** to disable a security rule or click **Delete** to delete a security rule.

6. In the Change Rule - SQL Console dialog box, modify the domain-specific language (DSL) statements of the security rule based on your business requirements. For more information about the syntax, see DSL syntax for security rules.

   For example, change the maximum number of SQL statements that can be executed at a time from 1,000 to 500.

   > **Note**
   > - A large number of security rule templates are provided for each checkpoint. You can click **Load from Template Database** to use a template.
   > - For more information about factors and actions, see Overview of security rule sets.

7. Click **Submit**.

7.2.3.3. SQLConsole for relational databases

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for relational databases on the SQLConsole tab, such as MySQL databases.

Default security rules

- Constraints on SQL statement categories: No constraints are imposed on data query language (DQL) statements. By default, DML statements, DDL statements, data control language (DCL) statements, and SQL statements that cannot be identified by DMS are all blocked. To execute DML, DDL, or DCL statements on the SQLConsole tab, you must configure and enable corresponding security rules.
- Constraints on permissions on databases, tables, and fields: By default, users can perform operations on databases, tables, and fields without permission validation. To enable permission validation, you must configure and enable security rules under the **SQL Permission Criteria** checkpoint. For more information, see Supported checkpoints.

Basic configuration items

Configuration item	Description
Maximum number of returned rows per query	The maximum number of rows that can be returned for a query.
Maximum number of rows returned for a single query with sensitive column conditions	The maximum number of rows that can be returned for a query that contains query conditions for sensitive fields.
Limit the maximum allowed SQL full table scan (MB)	The maximum size of data that can be scanned. Before an SQL statement is executed, DMS checks the execution plan. If the size of the data to be scanned exceeds the specified threshold, the SQL statement fails to be executed. **Note** This item can be configured only for MySQL and Oracle databases.
Turn off the execution of change SQL validation affects the number of rows and prompts	Specifies whether to check the number of rows to be affected and display a prompt before DMS executes an SQL statement to change data. By default, this item is disabled.
How many rows does result set page support	The maximum number of rows that can be returned in the query result set on the SQLConsole tab.
Does the result set support paging	Specifies whether the query result set can be displayed on multiple pages on the SQLConsole tab.
Does the result set support editing	Specifies whether the query result set can be edited on the SQLConsole tab.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
SQL Execution Quantity Criteria	Allows you to limit the number of SQL statements that can be submitted at a time.
DQL SQL Criteria	Allows you to set constraints on DQL statements.
Other SQL Criteria	Allows you to set constraints on multiple categories of SQL statements. Different enterprises may define different high-risk SQL statements, which may include specific subcategories of DML, DCL, and DDL statements. **Note** You can also set constraints on SQL statements that cannot be identified by DMS.
SQL Permission Criteria	Allows you to set constraints on the execution of SQL statements from the aspect of permissions. For example, DMS checks whether a user has the required permissions on the corresponding databases, tables, and fields.
SQL Execution Performance Criteria	Allows you to set constraints on the execution of SQL statements from the aspect of performance. For example, you can specify that a DML statement is not executed if the number of rows to be affected by the statement exceeds the specified threshold, or that a DDL statement is not executed if the size of the table involved exceeds the specified threshold.
Exception Recognition Criteria of Database and Table Column Permissions	After a user submits SQL statements on the SQLConsole tab, DMS parses the SQL statements and checks whether the user has the required permissions on the corresponding databases, tables, and fields. You can configure security rules under this checkpoint to ensure that if exceptions occur when DMS parses complex SQL statements, these statements can be executed. **Note** If you configure and enable security rules under the **Exception Recognition Criteria of Database and Table Column Permissions** checkpoint, security rules under the SQL Permission Criteria, DQL SQL Criteria, Other SQL Criteria, and SQL Execution Performance Criteria checkpoints are automatically disabled.
SQL Execution Criteria in Logical Databases	This checkpoint is reserved for logical databases and not suitable for physical databases.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors for relational databases on the SQLConsole tab.

Factor	Description
@fac.sql_count	The number of SQL statements that are submitted at a time.

Factor	Description
@fac.select_sql_count	The number of DQL statements among the SQL statements that are submitted at a time.
@fac.dml_sql_count	The number of DML statements among the SQL statements that are submitted at a time.
@fac.sql_type	The category and subcategory of the SQL statement. For more information, see Supported SQL statements.
@fac.sql_sub_type	
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.fulltable_delete	A Boolean value that indicates whether the current SQL statement deletes a full table. Valid values: *true* and *false*.
@fac.fulltable_update	A Boolean value that indicates whether the current SQL statement updates a full table. Valid values: *true* and *false*.
@fac.current_sql	The current SQL statement.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.
@fac.user_is_sec_admin	A Boolean value that indicates whether the current user is a security administrator. Valid values: *true* and *false*.
@fac.sql_affected_rows	The number of rows to be affected by the current SQL statement. **Warning** This factor triggers COUNT operations, which may affect the database performance. Use this factor with caution.
@fac.sql_relate_table_store_size	The estimated total size of the table to be accessed by the current SQL statement. Unit: MB. **Note** This value is estimated based on the metadata that is obtained by DMS. It is not an actual value.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for relational databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to execute the current SQL statement.
@act.allow_execute	Allows the current SQL statement to be executed.
@act.reject_sql_type_execute	Rejects the request to execute a specific subcategory of SQL statements. You must specify an SQL statement subcategory after the action name. Example: `@act.reject_sql_type_execute 'UPDATE'`.
@act.allow_sql_type_execute	Allows a specific subcategory of SQL statements to be executed. You must specify an SQL statement subcategory after the action name. Example: `@act.allow_sql_type_execute 'UPDATE'`.
@act.check_dml_sec_column_permission	Checks whether a user has the required permissions on sensitive fields. If the user does not have the permissions, the DML statement for data change is not executed.
@act.uncheck_dml_sec_column_permission	Does not check whether a user has the required permissions on sensitive fields.
@act.check_sql_access_permission	Checks whether a user has the required permissions, such as query and change permissions, on the databases, tables, and fields that are involved in the SQL statements to be executed.
@act.uncheck_sql_access_permission	Does not check whether a user has the required permissions on the objects that are involved in the SQL statements to be executed.
@act.enable_sec_column_mask	De-identifies sensitive fields in query result sets that are returned for SQL statements that are submitted by users who do not have permissions on the sensitive fields.
@act.disable_sec_column_mask	Does not de-identify sensitive fields in query result sets that are returned for SQL statements that are submitted by users who do not have permissions on the sensitive fields.

Supported SQL statements

Category	Subcategory
DQL	• SELECT • DESC • EXPLAIN • SHOW

Category	Subcategory
DML	• INSERT • INSERT_SELECT • REPLACE • REPLACE_INTO • UPDATE • DELETE • MERGE
DDL	• DATABASE_OP • CREATE • CREATE_INDEX • CREATE_VIEW • CREATE_SEQUENCE • CREATE_TABLE • CREATE_SELECT • TRUNCATE • DROP_INDEX • DROP_VIEW • DROP_TABLE • RENAME • ALTER • ALTER_INDEX • ALTER_VIEW • ALTER_TABLE • ALTER_SEQUENCE • CREATE_FUNCTION • CREATE_PROCEDURE • ALTER_FUNCTION • ALTER_PROCEDURE • DROP_FUNCTION • DROP_PROCEDURE
DCL	• GRANT • DECLARE • SET • ANALYZE • FLUSH • OPTIMIZE • KILL

7.2.3.4. SQLConsole for MongoDB

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for MongoDB databases on the SQLConsole tab.

Basic configuration items

Maximum number of returned rows per query: the maximum number of rows that can be returned for a query.

Supported checkpoints

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
User Permission Validation	Allows you to specify whether to check the permissions of specific users when they submit commands.
Collection Statement Criteria	Allows you to specify whether to allow DMS to run a specific category of commands.
DB Statement Criteria	
Cache Query Statement Criteria	
User Management Statement Criteria	
Role Management Statement Criteria	
Replication Set Statement Criteria	
Sharding Statement Criteria	

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as command categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors for MongoDB databases on the SQLConsole tab.

Factor	Description

Factor	Description
@fac.sql_sub_type	The subcategory of the current command. For more information about the supported commands, see Supported MongoDB commands.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as `DEV` or `PRODUCT`.
@fac.current_sql	The current command.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.
@fac.user_is_sec_admin	A Boolean value that indicates whether the current user is a security administrator. Valid values: *true* and *false*.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for MongoDB databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to run the current command.
@act.allow_execute	Allows the current command to be run.
@act.reject_sql_type_execute	Rejects the request to run a specific subcategory of commands. You must specify a subcategory after the action name. Example: `@act.reject_sql_type_execute 'UPDATE'`.
@act.allow_sql_type_execute	Allows a specific subcategory of commands to be run. You must specify a subcategory after the action name.

Supported MongoDB commands

Category	Subcategory	Command

Category	Subcategory	Command
Collection commands	Query commands	• aggregate • find • findOne • count • distinct • getIndexes • getShardDistribution • isCapped • stats • dataSize • storageSize • totalIndexSize • totalSize
	Data update commands	• insert • save • findAndModify • remove • update
	Collection modification commands	• drop • renameCollection
	Index modification commands	• createIndex • createIndexes • dropIndexes • reIndex
	Other commands	validate

Category	Subcategory	Command
Database commands	Database query commands	• commandHelp • currentOp • getCollectionInfos • getCollectionNames • getLastError • getLastErrorObj • getLogComponents • getPrevError • getProfilingStatus • getReplicationInfo • getSiblingDB • help • isMaster • listCommands • printCollectionStats • printReplicationInfo • version • serverBuildInfo • serverStatus,stats
	Collection creation commands	createCollection
	High-risk commands	• dropDatabase • fsyncLock • fsyncUnlock • killOp • repairDatabase • resetError • runCommand
Commands related to the query plan cache	Read commands	• getPlanCache • getPlansByQuery • listQueryShapes
	Write commands	clearPlansByQuery
	User query commands	• getUser • getUsers

Category	Subcategory	Command
User management commands	User modification commands	• createUser • changeUserPassword • dropUser • dropAllUsers • grantRolesToUser • revokeRolesFromUser • updateUser
Role management commands	Role query commands	• getRole • getRoles
	Role modification commands	• createRole • dropRole • dropAllRoles • grantPrivilegesToRole • revokePrivilegesFromRole • revokeRolesFromRole • updateRole
Replica set commands	N/A	• help • printReplicationInfo • status • conf
Sharding commands	N/A	• getBalancerState • isBalancerRunning

7.2.3.5. SQLConsole for Redis

DMS allows you to manage relational and non-relational databases on the SQLConsole tab. The definition and classification of security rules are different for relational and non-relational databases. This topic describes the security rules for Redis databases on the SQLConsole tab.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Permission Execution Statement Criteria	Allows you to set constraints on the permissions for command execution.

Checkpoint	Description
Statement Criteria: Keys	Allows you to specify whether to check the permissions of specific users when they submit commands.
Statement Criteria: String	Allows you to specify whether to allow the execution of various Redis commands.
Statement Criteria: List	
Statement Criteria: SET	
Statement Criteria: SortedSet	
Statement Criteria: Hash	
Statement Criteria: Other	

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors for Redis databases on the SQLConsole tab.

Factor	Description
@fac.cmd_type	The type of the Redis command. For more information about valid values, see Supported Redis commands.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as `DEV` or `PRODUCT`.
@fac.is_read	A Boolean value that indicates whether the current command is a read command. Valid values: *true* and *false*.
@fac.is_write	A Boolean value that indicates whether the current command is a write command. Valid values: *true* and *false*.
@fac.current_sql	The current command.
@fac.user_is_admin	A Boolean value that indicates whether the current user is a DMS administrator. Valid values: *true* and *false*.

Factor	Description
@fac.user_is_dba	A Boolean value that indicates whether the current user is a DBA. Valid values: *true* and *false*.
@fac.user_is_inst_dba	A Boolean value that indicates whether the current user is the DBA of the current instance. Valid values: *true* and *false*.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions for Redis databases on the SQLConsole tab.

Action	Description
@act.reject_execute	Rejects the request to run the current command.
@act.allow_execute	Allows the current command to be run.

Supported Redis commands

Category	Subcategory	Command
Key-related commands	Key-related read commands	• EXISTS • TTL • PTTL • RANDOMKEY • TYPE • SCAN • OBJECTS

Category	Subcategory	Command
	Key-related write commands	• DEL • DUMP • EXPIRE • EXPIREART • MOVE • PERSIST • PEXPIRE • PEXPIREAT • RENAME • RENAMENX • RESTORE • SORT • TOUCH • UNLIMK • WAIT • MIGRATE
String-related commands	String-related read commands	• GET • GETRANGE • BITCOUNT • GETBIT • MGET • STRLEN • BITOPS
	String-related write commands	• APPEND • BITFIELD • BITOP • DECR • DECRBY • GETSET • INCR • INCRBY • INCRBYFLOAT • MSET • MSETNX • PSETEX • SET • SETNX

Category	Subcategory	Command
List-related commands	List-related read commands	• LINDEX • LLEN • LRANGE
	List-related write commands	• BLPOP • BRPOP • BRPOPLPUSH • LINSERT • LPOP • LPUSH • LPUSHX • LREM • LSET • LTRIM • RTOP • RPOPLPUSH • RPUSH • RPUSHX
Set-related commands	Set-related read commands	• SCARD • SISMEMBER • SRANDMEMBER • SSCAN
	Set-related write commands	• SADD • SMOVE • SPOP • SREM

Category	Subcategory	Command
Sorted set-related commands	Sorted set-related read commands	• ZCARD • ZCOUNT • ZLEXCOUNT • ZRANGE • ZRANGEBYLEX • ZRANGEBYSCORE • ZRANK • ZREVRNGE • ZREVRANGEBYLEX • ZREVRANGEBYSCORE • ZREVRANK • ZSCAN • ZSCORE
	Sorted set-related write commands	• ZADD • ZINCRBY • ZINTERSTORE • ZPOPMAX • ZPOPMIN • ZREM • ZUNIONSTORE • BZPOPMIN • BZPOPMAX
Hash-related commands	Hash-related read commands	• HEXISTS • HGET • HLEN • HMGET • HSCAN • HSTRLEN
	Hash-related write commands	• HDEL • HINCRBY • HINCRBYFLOAT • HMESET • HSET • HSETNX

7.2.3.6. Data change

In DMS, you can execute SQL statements for data changes. However, the execution requires a high level of security. DMS allows you to configure security rules on the SQL Correct tab to validate the submission and approval of tickets for data changes. Only the SQL statements that are validated by the security rules can be executed.

Background information

Based on a DSL, new security rules are flexible to use. You can apply new security rules to define risk levels for tickets so that a ticket can be submitted to the approval process that is designed for the specified risk level. For more information, see DSL syntax for security rules.

Basic configuration items

Configuration item	Description
Data change default approval Template	By default, this approval template takes effect if you do not configure different approval rules for data changes at different risk levels under the Risk Approval Rules checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template. For more information, see Customize approval processes.
Data Change risk level list	This risk level list defines risk levels that are used in the **Risk Identification Rules** and **Risk Approval Rules** checkpoints to identify and classify risks in data changes. You can set risk levels based on the type and scenario of data changes. Data changes at different risk levels are submitted to different approval processes. DMS allows you to set the following four risk levels: • *low*: a low risk level. • *middle*: a medium risk level. • *high*: a high risk level. • *highest*: a critical risk level.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description

Checkpoint	Description

SQL execution rules	SQL execution rules are used to limit the SQL statements that can be submitted for execution. If you do not enable SQL execution rules, all SQL statements that are used for data changes cannot be executed. Assume that you want to use DML statements to change the data of a database in an online environment. You can create the following SQL execution rule:
	Example:
	`if`
	`  @fac.env_type not in ['product']`
	`  and`
	`  @fac.sql_type in [ 'UPDATE','DELETE','INSERT']`
	`then`
	`  @act.allow_submit`
	`end`
	Note:
	The preceding rule specifies that you can only submit data change tickets to execute UPDATE, DELETE, and INSERT statements on a database that is deployed in an online environment.

Risk Identification Rules	If a ticket conforms to the preset SQL execution rules, DMS continues to validate the ticket based on the risk identification rules. Risk identification rules are used to identify and classify risks in data changes. You can create risk identification rules based on your database environment, the number of rows in which data is affected, and the categories and subcategories of SQL statements.
	Note Different risk identification rules apply to different check items. DMS automatically identifies the highest risk level for a data change. For example, if the risk level of a data change is identified as high, medium, and low by one, three, and five risk identification rules, DMS assumes that the data change is at high risk.
	Example:
	`if`
	`  @fac.env_type not in ['product','pre']`
	`then`
	`  @act.mark_risk 'low 'Low risk level: offline environment'`
	`end`
	Note: The preceding rule specifies that if the destination database is deployed in an offline environment, data changes are at low risk.

Checkpoint	Description
Risk Approval Rules	After the risk level of a data change is identified by the risk identification rules, DMS processes the ticket based on the risk approval rules. You can customize risk approval rules under the **Risk Approval Rules** checkpoint. **Note** • If a data change does not hit risk approval rules, DMS uses the default approval template that is specified under the Basic Configuration Item checkpoint to process the ticket. • By default, an offline environment is identified as a factor at low risk and requires no approval.
Batch Data import rules	These rules apply only to the validation of data import tickets. You can use the default rules that are provided in templates, or configure rules based on your actual needs.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the SQL Correct tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.sql_type	The type of the SQL statement. The value is the subcategory of the SQL statement, such as UPDATE or INSERT. For more information, see Supported SQL statements.
@fac.detail_type	The type of the data change. Valid values: • *COMMON*: a Normal Data Modify ticket. • *CHUNK_DML*: a Lock-Free Data Modify ticket. • *PROCEDURE*: a Programmable Object ticket. • *CRON_CLEAR_DATA*: a History Data Clean ticket. • *BIG_FILE*: a Large Data Import ticket.
@fac.is_logic	A Boolean value that indicates whether the database to be affected is a logical database.
@fac.extra_info	The additional information about the data change. This factor is not in use.
@fac.is_ignore_affect_rows	A Boolean value that indicates whether to skip the validation.
@fac.insert_rows	The number of rows of data to be inserted.

Factor	Description
@fac.update_delete_rows	The number of rows of data to be updated.
@fac.max_alter_table_size	The size of the largest tablespace where the table to be modified is stored.
@fac.is_has_security_column	A Boolean value that indicates whether the SQL statement to be executed involves sensitive fields.
@fac.security_column_list	A list of sensitive fields that the SQL statement to be executed involves.
@fac.risk_level	The risk level of the operation that is to be performed by the SQL statement.
@fac.risk_reason	The reason for identifying the operation to be performed as at the risk level.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.` .The following table describes the supported action on the SQL Correct tab.

Action	Description
@act.allow_submit	Requires the submission of SQL statements to be executed in a ticket.
@act.allow_execute_direct	Allows the execution of SQL statements in the SQLConsole.
@act.forbid_execute	Forbids the execution of SQL statements.
@act.mark_risk	Marks the risk level of a data change. Example: `@act.mark_risk 'Medium risk level: online environment'` .
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.3.7. Permission application

DMS allows you to configure security rules on the Access apply tab to validate applications for permissions, including permissions on instances, databases, and tables.

Background information

In DMS, security rules are flexible to use. You can apply security rules to define risk levels for tickets so that a ticket can be submitted to the approval process that is designed for the specified risk level. For more information about the DSL syntax, see DSL syntax for security rules.

Basic configuration items

The following table describes the basic configuration items that are supported on the Access apply tab.

Configuration item	Description
[Instance-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for instance permission applications at different risk levels under the **Validation for Instance Permission Application** checkpoint. **Note** In the Switch Approval Template dialog box, you can change the approval process of the default approval template.
[DB-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for database permission applications at different risk levels under the **Database Permission Application Validation** checkpoint.
Table-permission request default approval Template	By default, this approval template takes effect if you do not set different approval processes for table permission applications at different risk levels under the **Table Permission Application Validation** checkpoint.
[Programmable object-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for programmable object permission applications at different risk levels under the **Programmable object verification** checkpoint.
[Field-permission application] default approval Template	By default, this approval template takes effect if you do not set different approval processes for sensitive field permission applications at different risk levels under the **Sensitive Field Application Validation** checkpoint.
Line-permission application default approval Template	By default, this approval template takes effect if you do not set different approval processes for row permission applications at different risk levels under the **Line permission application verification** checkpoint.
[Owner-application] default approval template (when the resource has no Owner)	By default, this approval template takes effect if you do not set different approval processes for data ownership applications at different risk levels under the **Owner Application Validation** checkpoint and the data that is involved in the application has no owner.

Configuration item	Description
[Owner-application] default approval template (when the resource has an Owner)	By default, this approval template takes effect if you do not set different approval processes for data ownership applications at different risk levels under the **Owner Application Validation** checkpoint and the data that is involved in the application has one or more owners.

Supported checkpoints

When a user submits a ticket to apply for permissions, DMS checks whether the ticket conforms to rules that are specified under checkpoints. The ticket can be submitted only after DMS determines that the ticket conforms to all rules that are specified under checkpoints.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Owner Application Validation	Allows you to set approval processes or constraints for **Instance-OWNER**, **Table-OWNER**, and **Database-OWNER** tickets.
Validation for Instance Permission Application	Allows you to set approval processes or constraints for **Instance-Performance** and **Instance-Login** tickets.
Database Permission Application Validation	Allows you to set approval processes or constraints for **Database-Permission** tickets.
Table Permission Application Validation	Allows you to set approval processes or constraints for **Table-Permission** tickets.
Programmable object verification	Allows you to set approval processes or constraints for **Programmable Object** tickets.
Sensitive Field Application Validation	Allows you to set approval processes or constraints for **Sensitive Column-Permission** tickets.
Line permission application verification	Allows you to set approval processes or constraints for **Row-Permission** tickets.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and database names. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Access apply tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.

Factor	Description
@fac.schema_name	The name of the database.
@fac.perm_apply_duration	The period of time during which the applicant needs the permission. Unit: hours.
@fac.column_security_level	The security level of the field. Valid values: • *sensitive* • *confidential* • *inner*

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported action on the Access apply tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.3.8. Data export

DMS allows you to manage security rules on the Data Export tab to validate the permissions of applicants on involved databases, tables, sensitive fields, and rows during the submission and approval of data export tickets. This helps ensure data security.

Basic configuration items

Data export default approval Template: the default approval template that takes effect if you do not set different approval processes for data export tickets at different risk levels under the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template. For more information, see Customize approval processes.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Pre-check Validation	Allows you to specify whether to validate the permissions of applicants on involved databases, tables, sensitive fields, and rows by configuring security rules.
Approval Rule Validation	Allows you to submit data export tickets to different approval processes by configuring security rules. For example, you can submit tickets for exporting more than a specific number of rows to an approval process and other tickets to another approval process.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Data Export tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.is_ignore_export_rows_check	A Boolean value that indicates whether to skip the check on the number of rows to be affected.
@fac.export_rows	The number of rows to be exported.
@fac.include_sec_columns	A Boolean value that indicates whether the data to be exported contains sensitive fields.
@fac.sec_columns_list	The sensitive fields that require or do not require approval before data is exported. The sensitive fields are displayed in the format of `Table name.Field name,[Table name.Field name, ...]`.
@fac.user_is_admin	A Boolean value that indicates whether the applicant is a DMS administrator.
@fac.user_is_dba	A Boolean value that indicates whether the applicant is a DBA.
@fac.user_is_inst_dba	A Boolean value that indicates whether the applicant is the DBA of the current instance.
@fac.user_is_sec_admin	A Boolean value that indicates whether the applicant is a security administrator.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`. The following table describes the supported actions on the Data Export tab.

Action	Description
@act.do_not_approve	Allows a ticket to be processed without approval.
@act.choose_approve_template	Specifies an approval template.
@act.choose_approve_template_with_reason	Specifies an approval template with a reason provided.
@act.forbid_submit_order	Forbids a ticket from being submitted.
@act.enable_check_permission	Validates the permissions of an applicant on involved databases and tables.
@act.disable_check_permission	Does not validate the permissions of an applicant on involved databases and tables.
@act.enable_check_sec_column	Validates the permissions of an applicant on involved sensitive fields.
@act.disable_check_sec_column	Does not validate the permissions of an applicant on involved sensitive fields.

7.2.3.9. Schema design

DMS allows you to configure security rules on the Schema Design tab to check the design rules and risk identification rules that apply to schema design tickets. This helps ensure data security.

Basic configuration items

Configuration item	Description
Enable non-peer Publishing	Specifies whether to enable non-peer publishing. By default, data changes to a table can be published only to a table with the same name in another database. After you enable non-peer publishing, you can perform data changes on all tables. **Warning** This feature may bring high risks. We recommend that you proceed with caution and enable this feature only for special requirements.
R & D process	The whole process of a schema design ticket. It is the most important configuration item on the Schema Design tab. For more information about the parameters of the configuration item, see Parameters involved in the R&D process.

Configuration item	Description
Field type configuration	The supported data types of fields to be added.
Index type configuration	The supported data types of indexes to be added.
It is forbidden to modify the original field data type	Specifies whether to prohibit the data types of the original fields from being modified when the original table is to be modified.
Prohibit deleting original fields	Specifies whether to prohibit the existing fields from being deleted when the original table is to be modified. **Note** We recommend that you enable this feature because deleting existing fields may bring high risks.
Prohibit renaming original fields	Specifies whether to prohibit the existing fields from being renamed when the original table is to be modified. **Note** We recommend that you enable this feature because reaming existing fields may bring high risks.
Table character set license configuration	The range of character sets that are allowed to be used when you create a table. For example, you can specify utf8 and utf8mb4.
Default approval template for Structural design	The default approval template that is used for a schema design ticket if you do not configure the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.
When published, the ticket will automatically advance to the end state	The point that is used to stop the schema change process. If you enable this feature, after the node that is set as the anchor in the R&D process is run, DMS automatically turns the ticket to the Finished state. **Note** To use this feature, you must set the last node in the R&D process as the anchor.

Supported checkpoints

Note Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

The Schema Design tab contains the following two processes:

- Process of saving changes: DMS provides the following three checkpoints for this process. The checkpoints validate the table headings, fields, and indexes.

- Save Changes and Validate Header
- Save Changes and Validate Field
- Save Changes and Validate Index

- Process of applying changes: DMS provides the following five checkpoints for this process. The first four checkpoints identify the risks that arise from changing schemas without locking tables, and the last checkpoint assigns an approval process to each type of risk.
  - Table Creation Risk Control
  - Field Change Risk Control
  - Index Change Risk Control
  - SQL Execution Risk Control
  - Approval Rule Validation

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors on the Schema Design tab.

Factor	Description
@fac.table_kind	The type of the table whose schema is to be changed. Valid values: • *new*: a newly created table. • *old*: an existing table.
@fac.column_kind	The type of the field to be changed. Valid values: • *new*: a newly created field. • *old*: an existing field.
@fac.xxxx_old	The value of an existing field or index that is used for comparison.
@fac.column_is_primary	A Boolean value that indicates whether the current field serves as a primary key. Valid values: *true* and *false*.
@fac.column_type_support_default	A Boolean value that indicates whether the data type of the current field supports a default value. Valid values: *true* and *false*. **Note** For example, a field of the CHAR type supports a default value, whereas a field of the TEXT type does not.
@fac.index_kind	The type of the index to be changed. Valid values: • *new*: a newly created index. • *old*: an existing index.
@fac.index_column_count	The number of fields in the index.

Factor	Description
@fac.change_type	The type of the schema change to be performed by DDL statements. Valid values: • *add*: adds one or more fields or indexes. • *modify*: modifies one or more fields or indexes. • *delete*: deletes one or more fields or indexes.
@fac.altered_table_size	The size of the table whose schema is to be changed. Unit: MB.
@fac.online_execute	A Boolean value that indicates whether the schema change can be performed in an online environment. Valid values: *true* and *false*.
@fac.change_risk_level	The risk level of the schema change. Valid values: • *high*: a high risk level. • *middle*: a medium risk level. • *low*: a low risk level.
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Schema Design tab.

Action	Description	Format
@act.block_submit	Blocks the submission of the schema change and displays the error message. This action can be used in the process of saving changes.	@act.block_submit 'Reason for blocking the submission'
@act.show_warning	Displays the error message without blocking the submission of the schema change. This action can be used in the process of saving changes.	@act.show_warning 'Error message'
@act.mark_middle_risk	Specifies that the schema change is at medium risk. This action can be used in the process of identifying the risk level.	@act.mark_middle_risk 'Reason for the identification'
@act.mark_high_risk	Specifies that the schema change is at high risk. This action can be used in the process of identifying the risk level.	@act.mark_high_risk 'Reason for the identification'

Action	Description	Format
@act.forbid_submit_publish	Rejects the ticket. This action can be used in the process of setting the approval process.	@act.forbid_submit_publish 'Reason for the rejection'
@act.do_not_approve	Specifies the ID of an approval template.	N/A
@act.choose_approve_template		
@act.choose_approve_template_with_reason		

Parameters involved in the R&D process

Parameter	Description
Step	• The type of the node. Valid values: • **Design**: The design node in the R&D process is generated by default and cannot be removed. It determines the environment where the schema change is designed. • **Publish**: A publish node in the R&D process is used to publish the schema change after the change is designed. You can set multiple publish nodes.
Node Name	The name of the node. The node name can be up to 10 characters in length.
Database Environment	The environment where the node is run.
Execution Strategy	• The way in which the node is run. Valid values: • **Immediately**: The node is run immediately after it is approved. • **Periodically**: The node is run at the time that you specify. **Note** If a node is approved before the specified point in time, it is run as scheduled. Otherwise, the node is interrupted and not run.
Can Go Back	Specifies whether a publish node can be rolled back to the design node.
Can Skip	Specifies whether the current node can be skipped.
Anchor	The point that is used to stop the schema change process. If you set a node as the anchor, after the node is published, the nodes that follow the anchor cannot be run and the schema change process ends. At this time, the ticket enters the Published state.
Actions	The operation that you can perform on a publish node. You can remove a publish node as required.

7.2.3.10. Database and table synchronization

DMS allows you to configure security rules on the Table Sync tab to validate operations that are related to schema synchronization, empty database initialization, and table consistency repair.

Basic configuration items

Configuration item	Description
Enable execution capability	Specifies whether to enable SQL-based synchronization. If this configuration item is set to OFF, applicants can compare table schemas but cannot execute SQL statements to synchronize databases and tables. Other configuration items and security rules you set under checkpoints on the Table Sync tab also become invalid.
Database table synchronization default approval Template	The default approval template for database and table synchronization applications. You can use the default approval template or click Switch Approval Template and select another template. For more information, see Customize approval processes.
Analysis phase script Expiration Time (unit: hours)	The timeout period of the analysis phase. You can set an appropriate timeout period in which synchronization can be canceled if schemas are changed in the destination database.

Supported checkpoints

The Table Sync tab contains three checkpoints that are corresponding to the three features that are supported by the tab. The three checkpoints are unrelated to each other. For example, when you submit a Schema Synchronization ticket, only the basic configuration items and the security rules that are specified under the Schema Synchronization Validation checkpoint are used to validate the ticket.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Checkpoint	Description
Schema Synchronization Validation	Allows you to set approval processes or constraints for Schema Synchronization tickets.
Empty Database Initialization Validation	Allows you to set approval processes or constraints for Empty Database Initialization tickets.
Table Consistency Repair Validation	Allows you to set approval processes or constraints for Repair Table Consistency tickets.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors on the Table Sync tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.schema_name	The name of the schema.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Table Sync tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.3.11. Sensitive field change

The topic describes the security rules on the Sensitive Column Change tab.

Basic configuration items

Sensitive column default approval Template: the default approval template that takes effect if you do not set approval processes for tickets that apply to change the security levels of sensitive fields under the Approval Rule Validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.

Supported checkpoints

Approval Rule Validation: When a user submits a ticket to change the security level of a sensitive field, DMS checks whether the ticket conforms to the rules that are specified under the Approval Rule Validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`.The following table describes the supported factors on the Sensitive Column Change tab.

Factor	Description

@fac.column_level_change_type	The type of security level change that the applicant wants to perform on a sensitive field. Valid values:
	• *upper*: raises the current security level, including the following three cases:
	  ◦ From inner to sensitive
	  ◦ From inner to confidential
	  ◦ From sensitive to confidential
	• *sensitive_to_inner*: lowers the security level from sensitive to inner.
	• *confidential_to_sensitive*: lowers the security level from confidential to sensitive.
	• *confidential_to_inner*: lowers the security level from confidential to inner.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Sensitive Column Change tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.3.12. Test data generation

This topic describes the security rules on the Test Data Generate tab.

Supported checkpoints

Approval rule validation: When a user submits a ticket to generate test data, DMS checks whether the ticket conforms to the rules that are specified under the Approval rule validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported factors

A factor is a built-in variable in DMS. You can use factors to obtain the context to be validated by security rules, such as SQL statement categories and the number of rows to be affected. A factor name starts with the prefix `@fac.`. The following table describes the supported factors on the Test Data Generate tab.

Factor	Description
@fac.env_type	The type of the environment. The value is the display name of the environment type, such as DEV or PRODUCT.
@fac.schema_name	The name of the schema.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`. The following table describes the supported actions on the Test Data Generate tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.3.13. Database cloning

This topic describes the security rules on the Database Clone tab.

Basic configuration items

Database clone default approval Template: the default approval template that takes effect if you do not set approval processes for database clone tickets under the Approval rule validation checkpoint. In the Switch Approval Template dialog box, you can change the approval process of the default approval template.

Supported checkpoints

Approval rule validation: When a user submits a database clone ticket, DMS checks whether the ticket conforms to the rules that are specified under the Approval rule validation checkpoint.

> **Note** Various security rule templates are built in each checkpoint. You can configure security rules based on these templates or create custom security rules as needed.

Supported actions

An action in a security rule is an operation that DMS performs when the if condition in the rule is met. For example, DMS can forbid the submission of a ticket, approve a ticket, or reject a ticket. An action in a security rule denotes the purpose of the security rule. An action name starts with the prefix `@act.`.The following table describes the supported actions on the Database Clone tab.

Action	Description
@act.forbid_submit_order	Forbids a ticket from being submitted. The statement is in the following format: `@act.forbid_submit_order 'Reason for forbidding the submission of the ticket'`.
@act.do_not_approve	Specifies the ID of an approval template.
@act.choose_approve_template	
@act.choose_approve_template_with_reason	

7.2.4. Configure security rules for a database instance

This topic describes how to configure security rules for a database instance.

Prerequisites

- You are a database administrator (DBA) or a Data Management (DMS) administrator.
- The control mode of the database instance is **Security Collaboration**.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Data Assets**. In the left-side navigation pane, click **Instances**.
3. Find the database instance for which you want to configure security rules and choose **More > Edit** in the Actions column.
4. (Optional)In the **Edit** dialog box, set the Control Mode parameter to **Security Collaboration** in the Basic Information section.
5. Select a security rule from the **Security Rules** drop-down list.
6. Click **Submit**.

7.3. Customize approval processes

Data Management (DMS) allows you to configure instance-level security rules so that you can customize different approval processes for different database instances or database operations. However, instance-level security rules have some limits in the production environment. This topic describes how to customize an approval process.

Prerequisites

You are a database administrator (DBA) or a DMS administrator.

Context

You can customize approval processes to resolve issues encountered in the following scenarios:

- Each database instance has only one DBA. However, multiple DBAs are included in an approval process to ensure business continuity regardless of whether one of the DBAs is unavailable.
- If multiple business units share the same database instance, each business unit must approve the tickets for their respective business operations in an approval process.

Procedure

This topic describes how to customize an approval process and specify multiple DBAs in the approval process. You can perform similar steps to customize an approval process in other scenarios.

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Approval Processes**.
3. Create an approval node.
   i. Click the **Approval Node** tab. Then, click **Create Approval Node**.
   ii. In the Create Approval Node dialog box, set the parameters. The following table describes the parameters that you can specify for the approval node.

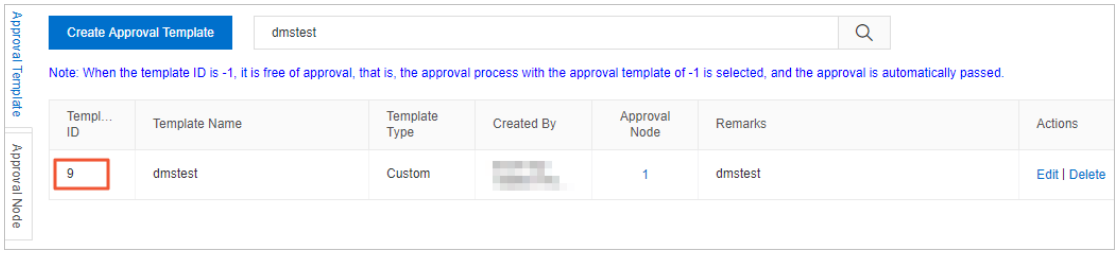
Parameter	Description
Node Name	The name of the approval node. The name must be globally unique.
Description	The description of the approval node. This parameter distinguishes the approval node from other approval nodes.
Approver	The Apsara Stack tenant accounts of the approvers for the approval node. You can search for approvers by keyword. Prefix match is used. **Note** Do not assign only one approver to an approval node. In most cases, we recommend that you set at least two approvers for each approval node.

   iii. Click **Submit**.
4. Create an approval template.
   i. Click the **Approval Template** tab. Then, click **Create Approval Template**.

ii. In the Create Approval Template dialog box, set the parameters. The following table describes the parameters that you can specify for the approval template.

Parameter	Description
Template Name	The name of the approval template. The name must be globally unique.
Description	The description of the approval template. This parameter distinguishes the approval template from other approval templates.
Approval Node	Click Add Node and click Select in the Actions column of the approval nodes that you want to use. In this example, the system node Owner and the approval node that is created in Step 3 are selected to allow multiple DBAs to participate in the approval process. **Note** The approval process is implemented based on the values of the Approval Order parameter in ascending order.

iii. Click **Submit**.
After the approval template is created, you can view the template ID. In this example, the template ID is 9.



5. Apply a new approval process.

   This example shows how to edit a rule that is applied to medium-level risk approval processes under the **Risk Approval Rules** checkpoint. You can perform similar steps to apply a rule to other scenarios.

   i. In the top navigation bar, click Security and Specifications. In the left-side navigation pane, click Security Rules.

   ii. Find the rule set that you want to edit and click **Edit** in the **Actions** column.

   iii. Click the **SQL Correct** tab.

   iv. Select **Risk Approval Rules** as the checkpoint.

   v. Find the rule that is related to the medium-level risk approval process and click **Edit**.

vi. In the **Rule DSL** field, change the template ID.

> **Note** In this example, change `3` to `9`, as shown in the preceding figure. The ID 9 is the ID of the approval template that is created in Step 4.

vii. Click **Submit**.

Result

If the data change tickets that you submit match the rule, all specified DBAs receive ticket approval notifications and can participate in the approval process.

7.4. Configure access IP address whitelists

Data Management (DMS) allows you to configure access IP address whitelists. You can allow users to access DMS only from a specific trusted network environment. This way, the service scope of DMS is controlled.

Prerequisites

You are a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Access IP Whitelists**.
3. Perform one or more of the following operations based on your business requirements:
   - Enable or disable the whitelist control feature

     Click **Click to Open** or **Click to Close** to enable or disable the whitelist control feature.
   - Create a whitelist
     a. Click **Create Whitelist**.

b. In the dialog box that appears, enter the IP addresses and description.

> **Note**
>
> - Separate IP addresses with semicolons (;). Make sure that each IP address in a whitelist is unique.
> - You can specify IP addresses such as 10.23.12.24 or CIDR blocks such as 10.23.12.24/24, where /24 indicates the length of the IP address prefix in the CIDR block. The IP address prefix can be 1 to 32 bits in length.
> - A value of 0.0.0.0/0 indicates that all IP addresses are allowed.

c. Click **Submit**.

- Modify a whitelist
  a. Find the IP address whitelist that you want to modify and click **Edit** in the **Actions** column.
  b. In the dialog box that appears, modify the IP address information.
  c. Click **Submit**.
- Delete a whitelist
  a. Find the IP address whitelist that you want to delete and click **Delete** in the **Actions** column.
  b. In the message that appears, click **OK**.

> **Note** You cannot delete all IP address whitelists. At least one IP address whitelist must be retained.

7.5. Use the operation audit feature

The operation audit feature provided by Data Management (DMS) allows you to query the information about operations that are performed in DMS, including SQL statements that are used in the SQLConsole, tickets, logon information, and operation logs. This helps you troubleshoot database issues and provides data for operation audit.

Feature modules

The following table describes the two modules of the operation audit feature in DMS: Operation Logs and Operation Audit.

Module	Description	Item
Operation Logs	Displays the logs of all the operations that are performed in DMS.	Includes the logs of management and configuration operations, SQL statements that are used in the SQLConsole, tickets, and logon information.

Module	Description	Item
Operation Audit	Displays all the operations that are performed on the databases in DMS. **Note** This module provides a user interface (UI) for you to audit operations in a centralized manner. This also helps you troubleshoot database issues with ease.	Includes SQL statements that are used in the SQLConsole, tickets, and logon information. **Note** Only DMS administrators, database administrators (DBAs), ticket submitters, and stakeholders involved in the ticket approval process are allowed to view the ticket details.

Note

- If the control mode of an instance is **Stable Change** or **Security Collaboration**, the log data of the instance is permanently retained in DMS. You can access and view the log data at any time.
- If the control mode of an instance is **Flexible Management**, you can view the log data only within seven days.

Procedure and supported roles

The following table describes the roles that you can assume to use the operation audit feature. It also shows you how to go to the Operation Audit tab in the DMS console.

Auditing dimension	Limits	Link to operation audit	Supported role
Database	You can view and audit only the operations that are performed on the current database.	• On the SQLConsole tab of the database that you want to audit, click the icon in the upper-right corner. • In the left-side navigation pane of the DMS console, click the instance to which the database you want to audit belongs, right-click the database, and then choose Audit > **Operation Audit**.	You can be a DMS administrator, a security administrator, a DBA, an instance owner, or a regular user. **Note** If you are a regular user, you can view and audit only the operations that you performed on the current database.

Auditing dimension	Limits	Link to operation audit	Supported role
Instance	You can view and audit only the operations that are performed on the current instance.	In the left-side navigation pane of the DMS console, right-click the instance that you want to audit and choose Audit > **Operation Audit**.	You can be a DMS administrator, a security administrator, a DBA, an instance owner, or a regular user. **Note** If you are a regular user, you can view and audit only the operations that you performed on the current instance.
Global	You can view and audit all the operations that are performed in DMS.	In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Operation Audit**.	You can be a DMS administrator, a security administrator, or a DBA.

Download operation records

This example shows you how to view and download all the SQL statements that are used in the SQLConsole in the last month.

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, click **Operation Audit**.
3. On the **Operation Audit** tab, click **SQL window list**.
4. Set the **Time** parameter to **Last One Month** and click **Search**.
   Then, DMS returns the search results.
5. Click the icon to download the results.

   Then, DMS exports an XLSX file that contains the search results displayed on the current page.

   Note To preview and export more results, you can set the **Items per page** parameter to **100**.

7.6. Manage sensitive data

Data Management (DMS) allows you to manage all classified sensitive and confidential fields in a unified manner. You can configure encryption algorithms for sensitive and confidential fields. This improves the control over the data masking feature.

Prerequisites

You are a security administrator, a database administrator (DBA), or an administrator.

Context

When you query a table that contains sensitive or confidential fields on which and you do not have permissions on the fields, the values of the fields are displayed as `******` in the query results. In this case, sensitive data is fully masked. In some scenarios, developers or test engineers may need to view a part of sensitive data for troubleshooting. To meet this requirement, you can configure masking algorithms to show some sensitive data.

Limits

- The sensitive data management feature applies only to relational databases such as MySQL. However, this feature is unavailable for NoSQL databases.
- To use this feature, the required database instance must be managed in security collaboration mode.

Procedure

1. Log on to the DMS console.
2. Specify security levels for fields in the required table.

   > **Note** If security levels are specified for the fields, skip this step.

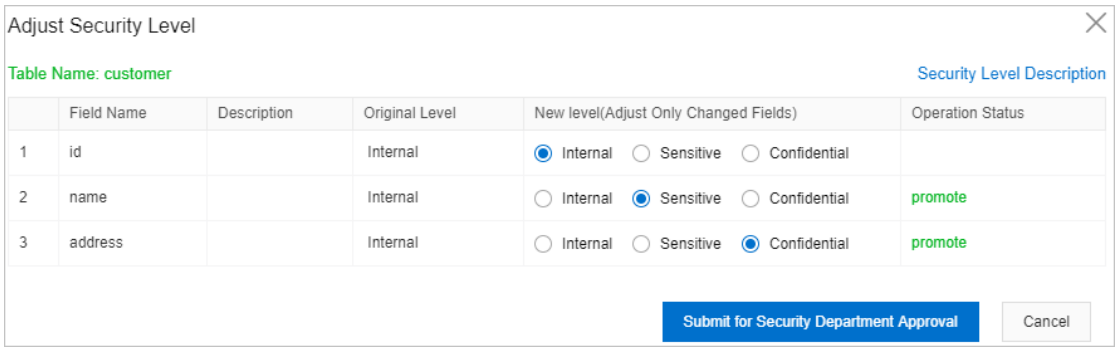
   i. In the left-side database instance list, click the icon next to the required database instance to show the databases in the instance.

   ii. Find the required database, right-click the database, and then select **Tables**.

   iii. Click the + icon next to the table name to show the table details.

   iv. Click **Adjust**.

   v. In the Adjust Security Level dialog box, change the security levels of fields.

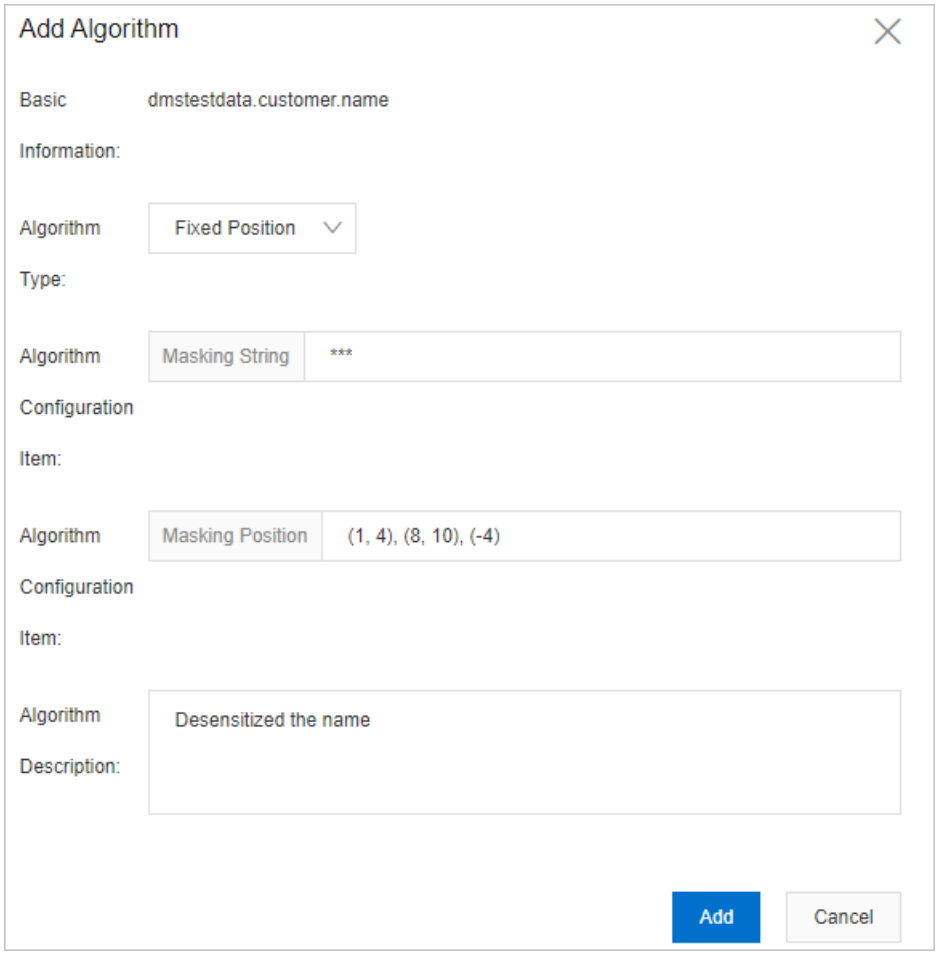


   vi. Click **Submit for Security Department Approval**.

   > **Note** The application to increase the security level of a field is automatically approved. The application to decrease the security level of a field is approved based on the approval process specified by an administrator or DBA.

   vii. In the message that appears, click **OK**.

3. In the top navigation bar, choose **More > System > Sensitive Data**.
4. Find the required field and click **Add Algorithm** in the **Actions** column of the field.
5. In the dialog box that appears, configure a masking algorithm.



Parameter	Description
Algorithm Type	The type of the algorithm. You can select an algorithm type based on your business requirements.

Parameter	Description
Algorithm Configuration Item	The algorithm configuration items vary based on the specified algorithm type. ○ **Fixed Position** algorithm type You must set the Masking String and Masking Position parameters. For example, you can set the Masking String parameter to ***. The Masking Position parameter specifies the positions of the characters to be masked in the field values. The positions are in the format of coordinates. Examples: ■ (1, 4): masks the first four characters. You can also enter (4) to simplify the format. ■ (-4): masks the last four characters. **Note** You can specify a maximum of three positions. For example, `(1, 4), (8, 10), (-4)` indicates to mask the first four characters, the eighth to tenth characters, and the last four characters. ○ **Fixed Character** algorithm type You must set the Masking String and Character to Be Replaced parameters. **Note** The Character to Be Replaced parameter specifies the characters that you want to mask in the format of a string. You can specify a maximum of three strings. ○ **Full Masking** algorithm type You need to set only the Masking String parameter.
Algorithm Description	Enter a description that can help you identify the algorithm.

6. Click **Add**.

7.6.1. Overview

Data Management (DMS) provides the sensitive data protection feature. You can use the feature to scan the metadata of a database for sensitive data. Then, you can mask and manage the sensitive data.

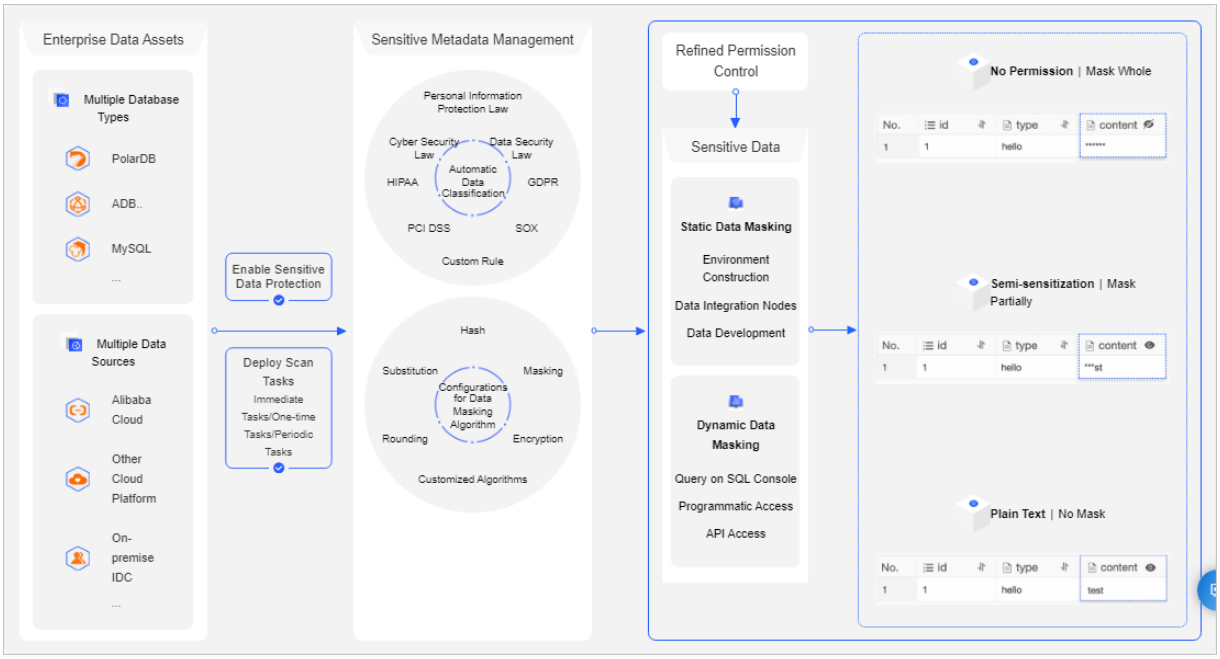
Supported databases

- Relational databases:
  - MySQL: ApsaraDB RDS for MySQL, PolarDB-X, and MySQL databases from other sources
  - SQL Server: ApsaraDB RDS for SQL Server and SQL Server databases from other sources
  - PostgreSQL: ApsaraDB RDS for PostgreSQL and PostgreSQL databases from other sources
  - ApsaraDB for OceanBase
  - Oracle
  - Dameng (DM)

- Data warehouses:
  - AnalyticDB for MySQL
  - AnalyticDB for PostgreSQL

Feature architecture

The sensitive data protection feature helps you effectively detect and protect sensitive data assets in your enterprise. This prevents sensitive data from being abused or leaked. The following figure shows the architecture of the sensitive data protection feature.



Features

- Sensitive data dashboard: DMS provides a sensitive data dashboard that helps an enterprise manage sensitive data in a centralized manner.
- Automated metadata scan
  - DMS allows you to customize a schedule for scanning data.
  - DMS automatically identifies and classifies enterprise sensitive data. This allows you to detect sensitive data in enterprise assets at the earliest opportunity and effectively manage the sensitive data.
  - DMS supports built-in and custom data classification templates to implement fine-grained classification and management of sensitive data. You can manage sensitive data based on the principle of least privilege.
- Sensitive data masking
  - DMS supports built-in and custom sensitive data masking rules. This helps you flexibly manage sensitive data masking algorithms. You can mask different fields based on your business scenarios, implement fine-grained permission control, and ensure least sensitive data exposure.
  - DMS provides a test environment that allows you to experiment with your sensitive data detection and masking rules.
  - DMS allows you to manage the access of users and applications to sensitive data.
- Sensitive data monitoring: DMS monitors the use of sensitive data, audits anomalous activities, and

generates alerts. This helps you trace anomalous activities and the source of data leaks.

Terms

- Sensitivity level: Certain business data, such as mobile numbers and ID card numbers, is sensitive. The fields that store sensitive data must be encrypted before they are displayed during regular queries. DMS supports the following three sensitivity levels based on the sensitivity of the data:
  - Low Sensitivity

    > **Note** For a database instance that is managed in Secure Collaboration mode, the security level of the data stored in the database instance is Low Sensitivity by default.

  - Moderate Sensitivity
  - High Sensitivity

    > **Note** After you set the sensitivity levels for data, take note of the following rules:
    > - When you query data in the SQLConsole, the Moderate Sensitivity and High Sensitivity fields on which you have no permissions are displayed as strings of asterisks (*) or in a custom manner.
    > - To query, export, or change Moderate Sensitivity or High Sensitivity fields, you must apply for the permissions on these fields.
    > - A database administrator (DBA) or DMS administrator can configure special approval processes for exporting or changing data that contains Moderate Sensitivity or High Sensitivity fields.

- Detection rule: DMS provides built-in sensitive data detection rules that are designed for different industries based on the relevant laws and regulations. You can also customize detection rules that detect sensitive data based on the metadata of a database or the data that is stored in the database.
- Data types: DMS provides data types that are defined based on various laws and regulations. You can also create custom data types.
  - Level 1 data types: include types such as personal information, enterprise information, and location information.
  - Level 2 data types: include types such as mobile phone numbers, email addresses, and bank card numbers.
- Masking algorithm: DMS supports hash, redaction, substitution, transformation, and encryption algorithms to mask data. You can configure custom masking rules based on the built-in masking algorithms.
- Masking policy: DMS generates a masking policy after you configure a masking rule for the selected sensitive fields.

7.6.2. Enable the sensitive data protection feature

If a database contains sensitive data, you can enable the sensitive data protection feature for the database. This way, Data Management (DMS) can scan the metadata in the database, and detect, mask, and manage the sensitive data. This topic describes how to enable the sensitive data protection feature. This topic also describes how to create a scan task to scan metadata.

Prerequisites

- You are a DMS administrator, a database administrator (DBA), or a security administrator.

  > **Note** To view the role of your account, move the pointer over the  icon in the upper-right corner of the DMS console.

- The database is supported by the sensitive data protection feature. The following types of databases are supported:
  - Relational databases: MySQL, SQL Server, PostgreSQL, Oracle, Dameng (DM), PolarDB-X, and ApsaraDB for OceanBase
  - Data warehouses: AnalyticDB for MySQL and AnalyticDB for PostgreSQL

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click Security and Specifications. In the left-side navigation pane, choose **Sensitive Data > Sensitive Data Dashboard**.
3. On the Sensitive Data Dashboard tab, click the **Not opened** tab in the **Instance List** section.
4. Find the instance for which you want to enable the sensitive data protection feature and click **Enable Now** in the **Operation** column.

   > **Note** Only instances for which the sensitive data protection feature is disabled appear on this tab.

5. In the **Enable Sensitive Data Protection** dialog box, turn on **Configure Scan Task**, set the Scan Method parameter, and then click **OK**.

   ### Configure a scan task for the instance

Option	Description
Immediate Task (Task Immediately Run Only Once)	After you configure an immediate task, DMS immediately scans the metadata in the specified database and marks sensitive data.
Scheduled Task (Task Run at Specified Time Only Once)	Specify a specific date and point in time. DMS automatically scans the metadata in the specified database and marks sensitive data as scheduled.
Periodic Task	Specify the time and interval to run the scan task. DMS automatically scans the metadata in the specified database and marks sensitive data on a regular basis.

6. In the **Scan Task Configuration Results** dialog box, click **Account Authorization**.

7. In the dialog box that appears, enter the username and password of the account that is used to log on to the destination database and click **OK**.
8. (Optional)To view the information of the scan task, click **Task details** in the **Operation** column. On the **Identification Tasks** tab, you can view the owner, the status, and the scan results of the scan task, and the time when the scan task was created, started, and completed.
9. (Optional)To view sensitive data and the sensitivity levels of the sensitive data in the specified instance, click **Sensitive Data List** in the **Operation** column, and click the **Field Control** tab. You can also manage sensitive fields on the Field Control tab. For example, you can adjust the sensitivity levels of fields, change the masking rules for fields, and grant permissions on fields. For more information, see Manage sensitive data.

7.6.3. Manage sensitive data

This topic describes how to adjust the sensitivity levels of fields and change the masking rules for fields. This topic also describes how to grant and revoke permissions on fields.

Prerequisites

You are a Data Management (DMS) administrator, a database administrator (DBA), or a security administrator.

> **Note** To view the role of your account, move the pointer over the  icon in the upper-right corner of the DMS console.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, choose **Sensitive Data > Sensitive Data List**.
3. On the Sensitive Data List page, click the **Field Control** tab.
   - Adjust the sensitivity level of one or more fields.
     a. Click the + icon to the left of a specific database. All fields of the database are displayed.
     b. Find the field for which you want to adjust the sensitivity level and click **Adjust Sensitivity Level** in the **Operation** column.

        > **Note** To adjust the sensitivity level of multiple fields to the same level, select the fields and click **Adjust Sensitivity Level** in the upper-left corner of this tab.

     c. In the **Sensitivity level adjustment** dialog box, select the sensitivity level that you want to set and click **Confirm**.
   - Change the masking rule for one or more fields.
     a. On the Field Control tab, select one or more fields for which you want to change the masking rule.
     b. Click **Change Masking Rule** in the upper-left corner of this tab.

c. In the **Data Masking Rule must be selected.** dialog box, select a custom masking rule and click **Save**. For more information about custom masking rules, see Create a data de-identification rule.

> **Note** The default data masking rule is DEFAULT. To reset the masking rule to DEFAULT for a field, click **Reset Masking Rule** in the **Operation** column.

- Grant permissions on fields.

> **Note** You can grant permissions on fields only for a database instance that is managed in Security Collaboration mode. You cannot grant permissions on fields for a database instance that is managed in Flexible Management or Stable Change mode.

a. On the Field Control tab, select one or more fields on which you want to grant permissions.

b. Click **Authorize User** in the upper-left corner of this tab.

c. In the **Authorize User** dialog box, select one or more users to which you want to grant permissions from the **Add User** drop-down list.

d. Set the parameters as required in the **Permission Configuration** section. The following table describes the parameters that you must specify. If you do not grant a user the permissions on fields, the values of the fields are encrypted to the user.

Parameter	Description
Permission	The type of the permissions that you want to grant. ■ Query: allows the selected users to query data by executing SQL statements in the SQLConsole. ■ Export: allows the selected users to submit tickets to export data. ■ Change: allows the selected users to submit tickets to change or import data.
Expire Date	The validity period of the permissions. **Note** If you want to grant permissions on the fields by day or hour, select **Others** from the drop-down list and specify the validity period.

e. Click **OK**.

- Revoke permissions on sensitive fields.
  - Find the sensitive field on which you want to revoke permissions and click **Management authority** in the **Operation** column.
  - On the Management authority page, select **Sensitive Column Permission** for the **Classification** parameter.
  - Click **Recycle Permission** in the **Actions** column.

> **Note**
>
> - To view the authorization details of the sensitive field, click **View Details** in the **Actions** column.
> - You can also grant or revoke other permissions on the database on the Management authority page.

7.6.4. Manage sensitive data detection rules

The sensitive data protection feature of Data Management (DMS) provides dozens of built-in sensitive data detection rules. These rules are designed based on the Cybersecurity Law of the People's Republic of China, the General Data Protection Regulation (GDPR), the Sarbanes-Oxley (SOX) Act, the Payment Card Industry (PCI) Data Security Standard (DSS), and the Health Insurance Portability and Accountability Act (HIPAA). These rules focus on protecting personal information. If the built-in sensitive data detection rules cannot meet your business requirements, you can create custom sensitive data detection rules.

Prerequisites

You are a DMS administrator, a database administrator (DBA), or a security administrator.

> **Note** To view the role of your account, move the pointer over the  icon in the upper-right corner of the DMS console.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, choose **Sensitive Data > Sensitive Data Identification**.
3. Click the **Identification Rules** tab.
4. Click **Create Rule**.
5. In the **Create Identification Rule** panel, set the parameters that are described in the following table. Then, click **Submit**.

Parameter	Description
Rule Name	The name of the sensitive data detection rule to be created. **Note** You cannot change the rule name after the rule is submitted.
Description	The description of the sensitive data detection rule. The description facilitates subsequent management.

Parameter	Description
Data type	The type of data to be detected by the rule. **Note** You can also manually add data types.
Sensitivity Level	The sensitivity level of a detected field. For more information, see Field security level. ○ Low Sensitivity: The Low Sensitivity level is derived from the Internal level of DMS. For a database instance that is managed in Secure Collaboration mode, the sensitivity level of the data stored in the database instance is Low Sensitivity by default. ○ Moderate Sensitivity: The Moderate Sensitivity level is derived from the Sensitive level of DMS. ○ High Sensitivity: The High Sensitivity level is derived from the Confidential level of DMS.
Rule Configurations	○ Metadata Scan: ■ Contain: If the name of a field contains the characters that you enter, the field is marked with the specified sensitivity level. ■ Exclude: If the name of a field contains the characters that you enter, the field is not marked with the specified sensitivity level. **Note** To use multiple keywords as filter conditions, separate them with commas (,). ○ Data Content Scan: Enter a regular expression that is used to match field values. **Note** To check whether the regular expression that you enter works as expected, enter test data and click **Test**. ■ If the message "The field matches the regular expression" is displayed, the regular expression works as expected. ■ If the message "The field does not match the regular expression" is displayed, the regular expression fails to match the test data and you need to modify the regular expression.

6. Enable the sensitive data detection rule that you create.

> Note
>
> - By default, a sensitive data detection rule is disabled after it is created. A sensitive data detection rule takes effect only after you enable the rule.
> - You cannot modify built-in sensitive data detection rules. However, you can disable built-in sensitive data detection rules.
> - After you enable or disable a sensitive data detection rule, the setting takes effect in the next scan task.

7.6.5. Create a data masking rule

The sensitive data protection feature of Data Management (DMS) provides a built-in data masking rule that masks the entire value of a field. You can also create custom data masking rules based on the built-in data masking algorithms. This topic describes how to create a data masking rule.

Prerequisites

You are a DMS administrator, a database administrator (DBA), or a security administrator.

> Note To view the role of your account, move the pointer over the  icon in the upper-right corner of the DMS console.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, choose **Sensitive Data > Data Masking Management**.
3. On the **Data Masking Rule** tab of the Data Masking Management page, click **Create Data Masking Rule**.
4. In the **Create Rule** panel, set the parameters as required.

   DMS provides five types of built-in data masking algorithms: hash (Hash), redaction (Cover up), substitution (Replacement), transformation (Transformation), and encryption (Encryption).

   - Hash
     - MD5: a widely used cryptographic hash function that can generate a 128-bit (16-byte) hash value.
     - SHA1: a cryptographic hash function that can generate a 160-bit (20-byte) hash value called a message digest.
     - SHA256: generates a 256-bit hash value.
     - HMAC: a cryptographic technique that uses keys and a hash function to perform authentication.
   - Cover up
     - Full cover: masks the entire value of a field.

       For example, if you want to fully mask the phone number 1381111****, set the Cover string parameter to **********. In this case, the phone number is masked to **********.

- Fixed position cover: masks the specified part of a field.

  For example, if you want to mask the second part of the IP address 192.168.255.254, set the Cover string parameter to *** and the Mask position configuration parameter to (5,7). In this case, the IP address is masked to 192.***.255.254.

- Fixed character mask: masks the specified characters of a field.

  For example, if you want to mask example in the email address username@example.com, set the Cover string parameter to ******* and the String to be obscured parameter to example. In this case, the email address is masked to username@*******.com.

- Replacement
  - Map replacement: replaces a specified string with another specified string.

    For example, if you want to replace ab in the string abcd with mn, set the Match String parameter to ab and the Replace By parameter to mn. In this case, the string is masked to mncd.

  - Random replacement: replaces the specified part of a field with the random characters that you specify.

    For example, if you want to replace username in the email address username@example.com with random characters, set the Replacement position parameter to (1,8) and the Random character parameter to abc. In this case, the email address may be masked to acbbbbac@example.com.

    > **Note** If you specify two or more random characters, the masking result is random.

- Transformation
  - Number rounding: rounds down a number to the Nth digit before the decimal point.

    For example, if you set the Keep the first decimal place parameter to 2, the number 1234.12 is masked to 1230.

  - Date rounded: rounds a date and time.

    For example, if you set the Date rounding level parameter to hour, 2021-10-14 15:15:30 is masked to 2021-10-14 15:00:00.

  - Character displacement: moves characters of a field leftward in a loop manner.

    For example, if you set the String left shift number parameter to 2, the number 345678 is masked to 567834.

- Encryption
  - DES: uses the Data Encryption Standard (DES) algorithm to encrypt data. The key is 8 characters in length, and the masking result is 16 characters in length.
  - AES: a more advanced encryption algorithm compared with the DES algorithm. The key is 16 characters in length, and the masking result is 32 characters in length.

5. (Optional)Test the masking rule.
   i. Enter the data to be masked.
   ii. Click **Test**.
   iii. Check whether the masking rule works as expected.

6. Click **Submit**.

> **Note** By default, the DEFAULT built-in rule is applied on sensitive data. For more information about how to apply a custom data masking rule on sensitive data, see Manage sensitive data.

7.6.6. Configure row-level access control

In some cases, different users may access different rows in the same table, which can be achieved by using views. Data Management (DMS) provides an alternative solution that is called the row-level access control feature to control access at the row level.

Prerequisites

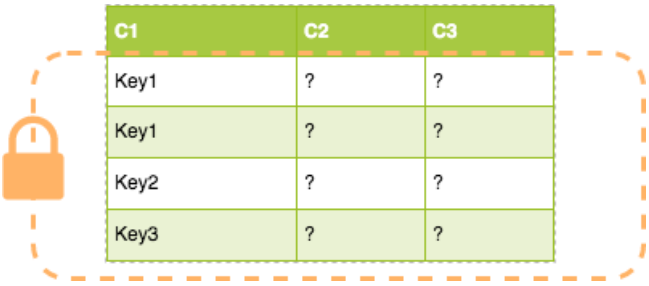
You are a DMS administrator, a database administrator (DBA), or a security administrator.

> **Note** To view the role of your account, move the pointer over the  icon in the upper-right corner of the DMS console.

Context

Row-level access control is used to provide horizontal data protection for tables. All the rows in a table are distinguished by one or more specified values. These values are called control values. To access a row that corresponds to a control value in the DMS console, you must have permissions on the row.

> **Note** A control value may correspond to multiple rows. If a user has permissions on a control value that corresponds to multiple rows, the user has permissions on all the rows that correspond to the control value.



Usage notes

- The sensitive data management feature applies only to relational databases such as MySQL. However, this feature is unavailable for NoSQL databases.
- To use the row-level access control feature, your database instance must be managed in Security Collaboration mode.
- The row-level access control feature applies only to physical databases. However, this feature is unavailable for logical databases.
- When you execute SQL statements to query, modify, or delete the data of a row-level control table, the following limits are set on filter conditions:

i. The control field must be specified in SQL statements to filter data.

ii. The system controls access to all the rows of a row-level control table. Users who do not have permissions on all rows can use only the `=` and `IN` operators to specify a control field. The control value that is specified in an SQL statement must be one of the control values for the table.

iii. Users who do not have permissions on all rows cannot use some operators, such as OR, XOR, and logical NOT.

Terms

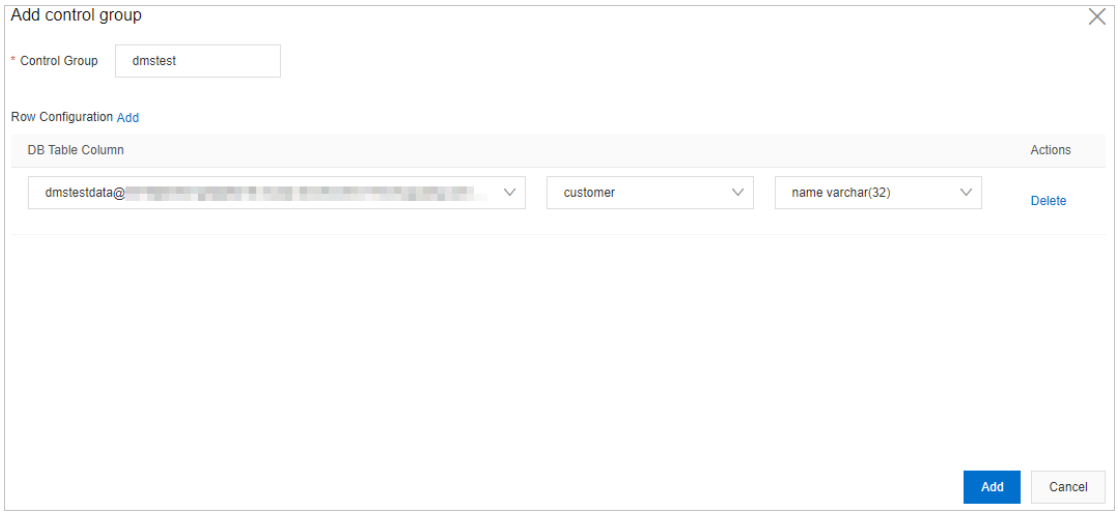
Term	Description
row permission	You can apply for permissions on a control value to access rows that correspond to the control value. Permissions on the rows of a table are defined as row permissions and are incorporated into the existing permissions of DMS. Permissions that can be controlled in Security Collaboration mode include permissions on databases, tables, fields, and rows.
single control value	When a user applies for permissions on the rows of a row-level control table, the user can select **Single** to apply for permissions on a single control value. **Note** A control value may correspond to multiple rows. If a user has permissions on a control value that corresponds to multiple rows, the user has permissions on all the rows that correspond to this control value.
all control values	When a user applies for permissions on the rows of a row-level control table, the user can select **ALL** to apply for permissions on all control values. After the application is approved, the user has permissions on all the rows of the table. In this case, the user can access the entire row-level control table without limits. Even if the control values are changed or more control values are added, the user still has permissions on all the rows of the table.
row-level control table	A table that requires row-level access control is called a row-level control table.
control field	A control field is a field to which the control values of a row-level control table are added.
control group	A control group is a group of row-level control tables that have the same control values. For example, if Table A and Table B have the same control values, you can add the two tables to a control group. This way, you can manage the two tables at the same time by using one set of control values.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Security and Specifications**. In the left-side navigation pane, choose **Sensitive Data > Sensitive Data List**.
3. On the Sensitive Data List page, click the **Row Control** tab.
4. Create a control group.

i. Click **Create Control Group**.

ii. In the Create Control Group dialog box, set the parameters that are described in the following table.



Parameter	Description
Control Group	Enter a name for the control group.
Row Configuration	Click **Add** to add a row configuration in which you can specify a database, a table, and a field. **Note** You can repeat this step to add multiple row configurations.
DB Table Column	Search for databases by keyword and select a database. Then, select a table and a field from the drop-down lists. **Note** The selected field is the control field.

iii. Click **Add**.

5. Add control values.

i. Find the created control group and click **Details** in the **Actions** column.

ii. Click **Add Row Value**.

iii. In the Import Row Value dialog box, specify whether to append row values and enter the required row values.

Note Separate multiple row values with commas (,).

iv. Click **Import**.

What to do next

After you configure row-level access control settings for a table, a user may still have no permissions on a control value that corresponds to one or more rows in the table. In this case, an error appears when the user queries row data. The error indicates that the user does not have permissions to access the row. The user can apply for permissions on the control value to access the rows. For more information, see Apply for permissions.

8.Create snapshots of full data on a T+1 basis

The T+1 full data snapshot feature of Data Management (DMS) allows you to create snapshots for specified tables every hour or day on a T+1 basis. This way, you can view the statistics on data by hour, day, or month. This topic describes how to create snapshots of full data on a T+1 basis by submitting a ticket.

Prerequisites

- A destination database where snapshots are stored is available.
  - Database type: AnalyticDB for MySQL.
  - The database instance to which the destination database belongs is managed in Secure Collaboration mode.
- A source database for which the snapshots are created is available.
  - Database type: ApsaraDB RDS for MySQL, PolarDB for MySQL, or a logical database that consists of multiple PolarDB for MySQL database shards.

    > **Note** To use a logical database, you must configure a logical database and the required logical tables first. For more information, see Logical database andLogical table.

  - The database instance to which the source database belongs must be managed in Security Collaboration mode. For more information about the Security Collaboration mode, see View the control mode of an instance.
- You log on to the DMS console by using an Alibaba Cloud account or as a RAM user to which the AliyunDTSFullAccess policy is attached.

  > **Note** If you do not attach the AliyunDTSFullAccess policy to the RAM user, a dialog box appears to inform you that you do not have the required permissions. For more information about how to attach the AliyunDTSFullAccess policy to a RAM user by using an Alibaba Cloud account or as a RAM user with the AdministratorAccess policy, see Grant permissions to a RAM user.

Context

The snapshot feature of traditional storage services backs up full data of the database or some tables at a specified point in time. If the amount of data is large, you cannot obtain the snapshots created at a specified point in time. In addition, your online database may be under pressure.

DMS parses database logs in real time to obtain incremental data of your database based on the real-time data synchronization feature of Data Transmission Service (DTS). This rarely affects data in the production environment. DMS stores the incremental data in history tables and schedules tasks to create snapshots of full data and store the snapshots to an AnalyticDB for MySQL database based on your configurations. Snapshots are partitioned by hour or day. Each partition contains full historical data.

> **Note** History tables provided by DMS not only store incremental data, but also record data changes of your database in real time. This allows you to query snapshots that are created at an arbitrary point in time as needed.

Scenarios

The T+1 full data snapshot feature is commonly used to store your business data to data warehouses. This feature allows you to synchronize full data on an hourly or daily basis. This way, you can view the statistics on data by hour, day, or month. You can use this feature in the following scenarios:

- Record the daily account balance for bill queries and account reconciliation in an accounting system.
- Record the daily price of a product to check whether the product is sold at the lowest ever price or whether the product is suitable for promotion.
- Collect statistics and calculate the total amount of orders on the previous day to obtain up-to-date information about business operations.

Limits

If the table schema of the source database changes, the data synchronization channel may become unavailable. Therefore, you can create only columns in the tables of the source database.

> **Note** If you use DDL statements to create or delete a table, clear data from a table, rename a table, delete a column, rename a column, or change column data, the data synchronization channel is interrupted. In this case, you must resolve the issue in Operation Center or submit a ticket.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **Solution**. In the left-side navigation pane, click **T+1 Full Data Snapshot**.
3. In the upper-right corner of the page, click **T+1 Full Data Snapshot**.
4. On the **T + 1 Full Snapshot Ticket** page, set the parameters that are described in the following table.

Parameter	Description
Ticket Name	The name of the ticket.
Snapshot Engine	The destination database used to store the snapshots.
Snapshot Data Source	The source database for which the snapshots are created.

Parameter	Description
Snapshot Table Settings	The tables for which you want to create snapshots. The following steps describe how to add one or more tables for which you want to create snapshots: i. In the left-side section, select one or more tables for which you want to create snapshots. ii. Click the » icon to add the selected tables to the right-side section. iii. In the **Snapshot Granularity** column, specify the scheduling cycle for creating snapshots. Valid values: ■ Hours: A snapshot is created every hour. ■ Day: A snapshot is created every day. iv. Select a field of the date and time type from the **Time Field** drop-down list. The data type of the field must be DATETIME, TIMESTAMP, or DATE. **Note** The system creates snapshots based on the time indicated by the specified time field rather than the system time.
Name of Hourly Snapshot Table	If you specify that a snapshot is created every hour, you must set the Prefix and Suffix parameters. By default, the value of the Prefix parameter is `ods_` and that of the Suffix parameter is `_dltahh`.
Name of Daily Snapshot Table	If you specify that a snapshot is created every day, you must set the Prefix and Suffix parameters. By default, the value of the Prefix parameter is `ods_` and that of the Suffix parameter is `_dlta`.
Full Snapshot Retention	Specifies whether to retain all snapshots. Valid values: ○ Yes: retains all snapshots. By default, each snapshot has 10,240 partitions and can be retained for up to 426 days. ○ No: does not retain all snapshots. You can specify the number of partitions for each snapshot based on the following formulas: ■ Number of partitions for an hourly created snapshot = Number of retention days × 24. ■ Number of partitions for a daily created snapshot = Number of retention days.

5. Click **Submit**.
   After the ticket is approved, snapshot tasks are run based on your configurations. The following list describes the execution cycle of the snapshot tasks:
   - A daily snapshot task starts from 01:00 to 01:10 every day. For example, a snapshot task creates a partition in the destination database and synchronizes all data generated before 00:00 on December 13, 2021, to this partition.
   - An hourly snapshot task starts within the first 5 minutes of each hour. For example, a snapshot

task creates a partition in the destination database and synchronizes all data generated before 16:00:00 on December 13, 2021, to this partition.

6. (Optional)View snapshots in the destination database.

   You can view the snapshots on the SQLConsole tab.

   In an hourly generated snapshot, if you want to query all data generated before 16:00:00 on December 13, 2021, execute the following SQL statement:

   ```
   SELECT * FROM 'Prefix_tablename_Suffix' WHERE ds='2021-12-13 16:00:00';
   ```

   > **Note**
   > - You can search for a snapshot by the prefix and suffix of its name.
   > - The partition key of a snapshot is the ds column. You can use the partition key to filter data and query the full data generated before a specified point in time.

7. (Optional)On the **T+1 Full Data SnapshotTickets** page, find the ticket whose information you want to view and click **Operation** in the **Actions** column.
   - View the execution information of a snapshot task:
     - If you select Day for the Snapshot Granularity parameter in the ticket, you can view daily snapshot tasks on the **Daily Tasks** tab. You can also stop or rerun a task.
     - If you select Hours for the Snapshot Granularity parameter in the ticket, you can view hourly snapshot tasks on the **Hourly Tasks** tab. You can also stop or rerun a task.
   - View the status of the data synchronization link: Click **Synchronization Link of Intermediate Table** to go to the **Task Management** page. On the Task Management page, you can view and manage the data synchronization channel.

9.System management

9.1. Manage users

Data Management (DMS) allows you to manage users. This topic describes how to add users and perform operations on users, such as modifying the permissions and roles of users.

Prerequisites

You are a DMS administrator.

Add a user

Add a user to the DMS tenant of your Alibaba Cloud account.

1. Log on to the DMS console.
2. In the top navigation bar, click **O&M**. In the left-side navigation pane, click **Users**.
3. In the upper-left corner of the page, click **New**.
4. In the **Add User** dialog box, enter the unique ID (UID) of your Alibaba Cloud account in the **Alibaba Cloud Account** field.

   > **Note** To view the UID, move the pointer over the  icon in the upper-right corner of the DMS console.

5. Select one or more system roles for the user to add.

Role	Description	Permission
Regular user	○ Regular users can perform operations on databases. For example, they can query and change data, or view and change schemas. ○ Regular users can be the R&D staff, testers, product staff, operations staff, or data analysts of enterprises.	○ Regular users cannot use the Instances, Users, Task, Configuration Management, Notification, Database Grouping, or Intelligent Operation feature in the DMS console. ○ To execute SQL statements on the SQLConsole tab or use the features of the Data Plans module, regular users must apply for the required permissions first.
Security administrator	○ Security administrators can perform operations such as determining the sensitivity levels of fields and auditing user operations. ○ Security administrators can be the internal auditors or security administrators of enterprises.	In addition to all the features that are available for regular users, security administrators can also use the **Operation Logs**, **Sensitive Data**, and **Data Protection** features.

Role	Description	Permission
DBA	○ Database administrators (DBAs) are responsible for database management, including managing database instances, database development standards and processes, and task execution. ○ DBAs in DMS can be the DBAs or O&M staff of enterprises.	In addition to all the features that are available for regular users, DBAs can also use all features except for the **Users** feature.
DMS administrator	○ No limit is set on the number of DMS administrators within a DMS tenant. ○ DMS administrators are approvers for the Admin approval step of an approval process.	○ Only DMS administrators can use the **Users** feature. ○ DMS can use all features in DMS.
Technical support personnel	Technical support personnel can be staff such as data analysts.	Technical support personnel have permissions on the metadata of database instances, databases, and tables. The permissions include viewing table details and exporting database schemas.

6. Click **OK**.

Perform operations on a user

Operation	Description	Procedure
Edit a user	Edit a user, such as editing the name and role of a user.	Find the user that you want to edit and click **Change** in the **Actions** column. In the Edit User dialog box, you can edit the display name, role, and mobile phone number of the user.
Grant permissions	Grant permissions on database instances, databases, tables, columns, and sensitive columns.	Find the user to which you want to grant permissions and move the pointer over **Authorize** in the **Actions** column. You can select an option based on your business requirements.
Enable access control	If you enable the access control feature for a user, the user can view and access only databases on which permissions are granted.	Find the user for which you want to enable the access control feature and choose **More > Access control** in the **Actions** column.

Operation	Description	Procedure
View permission details	View the permissions that are granted to a user, such as permissions on database instances, databases, tables, rows, sensitive columns, and the secure access proxy feature.	Find the user whose permissions you want to view and choose **More** > **Permission Details** in the **Actions** column.
Disable a user	If you disable a user, the user cannot perform operations as the user of the current tenant.	Find the user that you want to disable and choose **More** > **Disable** in the **Actions** column.
Remove a user	Remove a user from the current tenant.	Find the user that you want to remove and choose **More** > **Delete** in the **Actions** column.

9.2. Task management

The task management feature allows you to manage a variety of tasks that are created by using tickets. You can also use this feature to directly create SQL tasks.

Prerequisites

You are a database administrator (DBA) or a Data Management (DMS) administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **O&M**. In the left-side navigation pane, click **Task**.
3. On the **Task** tab, view and manage tasks that are created by using tickets.
4. Find a task and perform one of the following operations based on your business requirements:
   - Pause a task

     On the Task tab, find the task that you want to pause and click **Pause** in the Operation bar column.
   - Retry a task

     On the Task tab, find the task that is in the **Failure** state and click **Retry** in the Operation bar column.
   - Delete a task

     On the Task tab, find the task that you want to delete and click **Delete** in the Operation bar column. The task is in the **Delete** state and can no longer be run.
   - Create a task

     Click **Add SQL task**. In the Add SQL task dialog box, enter the task description, the database that you want to manage, and the SQL statements that you want to execute. Then, click **Submit Task**.
5. (Optional)Click **Add SQL Task**. In the dialog box that appears, set parameters as required and click

Submit Task.

9.3. Configuration

Data Management (DMS) allows DMS administrators to manage system configurations. If you are a DMS administrator, you can modify the system configuration items to flexibly meet your business requirements.

Prerequisites

You are a DMS administrator.

Procedure

1. Log on to the DMS console.
2. In the top navigation bar, click **O&M**. In the left-side navigation pane, click **Configuration Management**.
3. Find the required parameter and click **Change** in the **Actions** column of the parameter.

   > **Note** You can click **Change History** to view the change history of the parameter.

4. In the Change Parameter Configuration dialog box, enter the required value.
5. Click **Confirm Change**.

Types of data changes

key	value	Description
config_correct	Modify Config	Modifies configurations.
project_init_data	Init Project Data	Initializes the data for a project.
program_bug	Program Bug	Fixes a bug.
require_deal_without_backend_function	Requirements Without Backend Function	Manages the data of an application that does not support backend management.
history_data_clear	History Data Clean	Clears historical data.
test	Test	Runs a test.
mis_operation	Mis Operation	Restores data after a misoperation.
others	Others	Changes data for other reasons.

9.4. Database grouping

This topic describes how to create a database group in Data Management (DMS). You can use this feature to apply a data change or a schema change to all of the databases in a database group with ease.

Prerequisites

The databases that you want to add to a database group meet the following conditions:

- All of the instances to which the databases belong are managed in Security Collaboration mode.
- All of the databases are physical databases or logical databases.
- All of the databases are deployed in the same environment, such as the development environment.
- The engines of the databases are of the same type. For example, all of the databases are MySQL databases.

Create a database group

1. Log on to the DMS console.
2. In the top navigation bar, click **O&M**. In the left-side navigation pane, click **Database Grouping**.
3. Click **New Group**.
4. In the **NewGrouping** dialog box, enter a group name in the **Group name** field.
5. Set the Grouping type parameter to **General grouping** or **Remote live**.
6. Click **Add database**.
7. In the **Search database** dialog box, add multiple databases.
   i. Enter keywords to search for databases.
   ii. Click **Add** next to the database that you want to add.
   iii. Add other databases.
   iv. After the databases are added, close the **Search database** dialog box.
8. Click **Save**.

Scenarios

- Data change

  For example, you want to create a ticket to perform a data change on a database, and the database belongs to a database group. After you select the database, DMS displays a message to remind you that the selected database belongs to a database group. If you click **OK**, DMS adds all the other databases in the group as the databases on which the data change will be performed. This saves your effort in selecting databases one by one. If you click **Cancel**, the other databases in the group will not be selected.

  Data change operations supported by this feature include Change data and Import data.

- Schema design

  For example, you want to create a schema design ticket and select a database that belongs to a database group as a base database. After you click **Perform Changes to Base Database**, DMS displays a message. This message is used to remind you that the base database belongs to a database group and the current operation will apply to all the other databases in the group.

  For more information about how to use the schema design feature, see Design a schema.