

Data Admin Service

User Guide

Issue 01
Date 2023-03-02



Copyright © Huawei Technologies Co., Ltd. 2023. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Overview	1
2 Permissions Management	2
2.1 Creating a User and Granting Permissions	2
2.2 DAS Custom Policies	3
3 Logging In to the DAS Console	4
4 Adding Login Information	5
5 Logging In to a Database	7
6 Modifying Login Information	8
7 Deleting Login Information	9
8 Logging In to Databases Shared by Others	10
9 MySQL DB Instance Management (Development Tool)	12
9.1 Homepage	12
9.1.1 Overview	12
9.1.2 Creating a Database	13
9.1.3 Dropping a Database	13
9.1.4 Data Dictionary	13
9.2 SQL Operations	14
9.2.1 SQL Window	14
9.2.2 Executing SQL Statements	14
9.2.3 Executing SQL Plan	15
9.2.4 SQL Favorites	16
9.2.5 SQL History	16
9.3 Database Management	17
9.3.1 Table Management	18
9.3.1.1 Creating a Table	18
9.3.1.2 Opening a Table	20
9.3.1.3 Viewing Table Details	21
9.3.1.4 Altering a Table	21
9.3.1.5 Renaming a Table	21
9.3.1.6 Clearing a Table	22

9.3.1.7 Dropping a Table.....	22
9.3.1.8 Maintaining a Table.....	22
9.3.2 View Management.....	23
9.3.2.1 Creating a View.....	23
9.3.2.2 Opening a View.....	24
9.3.2.3 Altering a View.....	25
9.3.2.4 Dropping a View.....	25
9.3.3 Stored Procedure Management.....	25
9.3.3.1 Creating a Stored Procedure.....	25
9.3.3.2 Altering or Executing a Stored Procedure.....	26
9.3.3.3 Dropping a Stored Procedure.....	26
9.3.3.4 Viewing Stored Procedure Details.....	27
9.3.4 Event Management.....	27
9.3.4.1 Creating an Event.....	27
9.3.4.2 Altering an Event.....	28
9.3.4.3 Dropping an Event.....	28
9.3.5 Trigger Management.....	28
9.3.5.1 Creating a Trigger.....	28
9.3.5.2 Altering a Trigger.....	29
9.3.5.3 Dropping a Trigger.....	29
9.3.6 Function Management.....	29
9.3.6.1 Creating a Function.....	29
9.3.6.2 Altering or Executing a Function.....	30
9.3.6.3 Dropping a Function.....	30
9.3.6.4 Viewing Details of a Function.....	30
9.4 Import and Export.....	30
9.4.1 Data Import.....	30
9.4.2 Data Export.....	31
9.5 Structure Management.....	32
9.5.1 Table Structure Comparison and Synchronization.....	32
9.6 Data Scheme.....	33
9.6.1 Data Generator.....	33
9.6.2 Data Tracking and Rollback.....	34
9.7 Background Tasks.....	36
9.7.1 Task Scheduling.....	36
9.7.2 Scheduled Backup.....	37
9.8 Account Management.....	38
9.8.1 Database Account Security.....	38
9.8.2 Creating a User.....	39
9.8.3 Editing User Information.....	42
9.8.4 Deleting a User.....	43
10 RDS for PostgreSQL Instance Management (Development Tool).....	44

10.1 Homepage.....	44
10.1.1 Overview.....	44
10.1.2 Creating a Database.....	45
10.1.3 Dropping a Database.....	45
10.2 SQL Operations.....	45
10.2.1 SQL Window.....	45
10.2.2 Executing SQL Statements.....	46
10.2.3 Executing the SQL Plan.....	47
10.2.4 SQL Favorites.....	47
10.2.5 SQL History.....	48
10.3 Database Management.....	48
10.3.1 Table Management.....	49
10.3.1.1 Opening a Table.....	49
10.3.1.2 Viewing Table Details.....	50
10.3.2 View Management.....	50
10.3.2.1 Opening a View.....	50
10.3.2.2 Obtaining View Details.....	51
10.3.3 Stored Procedure Management.....	51
10.3.3.1 Viewing Stored Procedure Details.....	51
10.3.4 Trigger Management.....	51
10.3.4.1 Viewing Trigger Details.....	51
10.3.5 Sequence Management.....	51
10.3.5.1 Viewing Sequence Details.....	52
10.4 Import and Export.....	52
10.4.1 Data Import.....	52
10.4.2 Data Export.....	53
10.5 Account Management.....	53
10.5.1 Role Management.....	54
11 DDS DB Instance Management (Development Tool).....	55
11.1 Homepage.....	55
11.1.1 Overview.....	55
11.1.2 Creating a Database.....	56
11.1.3 Dropping a Database.....	56
11.2 Command Operation.....	56
11.2.1 Command Window.....	56
11.2.2 Command History.....	57
11.3 Database Management.....	58
11.3.1 Collection Management.....	58
11.3.1.1 Creating a Collection.....	58
11.3.1.2 Opening a Collection.....	59
11.3.1.3 Altering Validator.....	60
11.3.1.4 Renaming a Collection.....	60

11.3.1.5 Clearing a Collection.....	60
11.3.1.6 Dropping a Collection.....	61
11.3.1.7 Creating an Index.....	61
11.3.1.8 Editing an Index.....	61
11.3.1.9 Deleting an Index.....	62
11.3.2 View Management.....	62
11.3.2.1 Creating a View.....	62
11.3.2.2 Opening a View.....	63
11.3.2.3 Editing a View.....	63
11.3.2.4 Dropping a View.....	63
11.4 Account Management.....	64
11.4.1 User Management.....	64
11.4.2 Role Management.....	65
12 Distributed DDM DB Instance Management (Development Tool).....	66
12.1 Homepage.....	66
12.1.1 Overview.....	66
12.1.2 Creating and Deleting a Database.....	67
12.2 SQL Operations.....	67
12.2.1 SQL Window.....	67
12.2.2 Executing SQL Statements.....	68
12.2.3 Executing SQL Plan.....	70
12.2.4 SQL Favorites.....	70
12.2.5 SQL History.....	72
12.3 Database Management.....	72
12.3.1 Table Management.....	74
12.3.1.1 Creating a Table.....	74
12.3.1.2 Opening a Table.....	74
12.3.1.3 Viewing Table Details.....	75
12.3.1.4 Altering a Table.....	75
12.3.1.5 Clearing a Table.....	76
12.3.1.6 Dropping a Table.....	76
12.4 Account Management.....	76
12.4.1 User Management.....	77
13 Intelligent O&M.....	79
13.1 Overview.....	79
13.2 MySQL DB Instances.....	79
13.2.1 Function Overview.....	79
13.2.2 Performance.....	82
13.2.2.1 Real-Time Diagnosis.....	82
13.2.2.2 Real-Time Performance.....	82
13.2.2.3 Performance History.....	84
13.2.2.4 Performance Trends Comparisons.....	84

13.2.2.5 Custom Graphs.....	85
13.2.3 Sessions.....	85
13.2.3.1 Real-Time Sessions.....	85
13.2.3.2 Emergency Channel.....	85
13.2.4 SQL.....	86
13.2.4.1 Slow Query Logs.....	86
13.2.4.2 SQL Explorer.....	87
13.2.4.3 SQL Diagnosis.....	88
13.2.4.4 SQL Statement Concurrency Control.....	89
13.2.5 Daily Reports.....	91
13.2.6 Locks and Transactions.....	92
13.2.6.1 InnoDB Lock Query.....	92
13.2.6.2 Metadata Locks.....	92
13.3 DDM Instances.....	93
13.3.1 Function Overview.....	93
13.3.2 SQL.....	95
13.3.2.1 Slow Query Logs.....	95
13.3.2.2 SQL Explorer.....	96
14 Audit.....	98
14.1 DAS Key Operations Supported by CTS.....	98
14.2 Querying Traces.....	100
A Change History.....	101

1 Overview

On the cloud, managing data is complex. With DAS, you can manage your data easily and securely.

DAS provides differentiated UIs for different scenarios and offers you a list of functions in a clear and friendly manner.

Development Tool

This feature allows you to manage DB instances on a web console, simplifying database management and improving working efficiency and data security.

The way to manage databases varies with database type. This document introduces database management operations on different types of databases.

- [MySQL DB Instance Management \(Development Tool\)](#)
- [DDS DB Instance Management \(Development Tool\)](#)
- [RDS for PostgreSQL Instance Management \(Development Tool\)](#)

Intelligent O&M

Allows you to comprehensively analyze DB instances to understand their performance and operating statuses. Intelligent O&M provides multi-dimensional and real-time performance monitoring information, including real-time performance of a single instance, performance comparisons of multiple instances, and performance dashboard of multiple instances.

2 Permissions Management

2.1 Creating a User and Granting Permissions

You can use IAM to implement refined permission control for DAS resources. To be specific, you can:

- Create IAM users for employees from different departments of your enterprise in your account. In this way, each IAM user has a unique security credential to use DAS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or cloud service to perform efficient O&M on your DAS resources.

If your account does not require individual IAM users, skip this chapter.

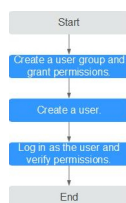
[Figure 2-1](#) describes how to grant permissions to a user group.

Prerequisites

Before granting permissions to a user group, learn about the DAS system policies that can be added, and select a policy based on service requirements, as shown in [Permissions Management](#).

Process Flow

Figure 2-1 Process for granting DAS permissions



1. Create a user group and assign permissions to it.
Create a user group on the IAM console, and attach the **DAS FullAccess** policy to the group.

2. Create a user and add it to a user group.
Create a user on the IAM console and add the user to the group created in [1](#).
3. Log in and verify permissions.
In the service list, choose **Data Admin Service**. On the displayed DAS console, click **Add Login**. If login information can be successfully added, the DAS policy has already taken effect.

2.2 DAS Custom Policies

Custom policies can be created to supplement the system-defined policies of DAS.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
 - JSON: Edit JSON policies from scratch or based on an existing policy.
- . This section describes example custom policies of DAS.

Examples of DAS Custom Policies

Example 1: Assign only DAS permissions.



```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "das:*:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example 2: Assign users the permission to use Intelligent O&M but not Development Tool.

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "das:connections:list"
      ],
      "Effect": "Allow"
    }
  ]
}
```

3 Logging In to the DAS Console

Logging In to Console

- Step 1** On the login page, enter your account and password to log in to the management console.
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner, and under **Database**, click **Data Admin Service**.
- End

4 Adding Login Information

Scenarios

This section describes how to add database login information so that you can log in to a database for visualized data management operations.

NOTE

- If you have not logged in to a DB instance through the created login over a year, the system automatically clears all information, including the saved password, of this login.
- You can log in to the same database either on the DAS console or using other clients. Both methods do not affect each other.
- DAS allows you to manage DB instances of different types of engines. An RDS for MySQL instance is used in the following example.

Adding Login Information for RDS Instances

- Step 1** In the navigation pane on the left of the DAS console, choose **Development Tool** to go to the login list page.
- Step 2** Click **Add Login**.
- Step 3** On the displayed page, select the DB engine, source database, and target DB instance, enter the login username, password, and description (optional), and enable **Collect Metadata Periodically** and **Show Executed SQL Statements**.

 **NOTE**

- After you specify **Source Database**, the system automatically lists all the DB instances from this source under the current account.
- The username and password required for adding the login are the database username and password.
- To delete the password, you can modify or delete the login.
- You are advised to enable **Collect Metadata Periodically**. If it is disabled, DAS obtains only the structured data from databases in real time, and the performance of databases is affected.

The collection time cannot be customized. Once **Collect Metadata Periodically** is enabled, DAS collects metadata at 20:00 every day (UTC time). If you are not using a UTC time, convert the time according to your local time zone. You can also click **Collect Now** to collect metadata at any time you want.

- You are advised to enable **Show Executed SQL Statements**. With it enabled, you can view the executed SQL statements under **SQL Operations > SQL History** and execute them again without entering the SQL statements.

Step 4 Test the connection as needed. If a message indicating that the connection failed is displayed, modify the connection according to the failure causes contained in the message.

Step 5 Click **OK**.

----End

Adding Login Information for Self-built Databases on ECSs

Step 1 In the login list of Development Tool, click **Add Login**.

Step 2 On the displayed page, select the DB engine, source database, and target DB instance, enter the port, login username, password, and description (optional), and enable **Collect Metadata Periodically** and **Show Executed SQL Statements**.

Step 3 Click **Test Connection**. If a message is displayed indicating connection successful, continue with the operation. If a message is displayed indicating connection failed and the failure cause is provided, make modifications based on the error message.

 **NOTE**

- Self-built databases on ECSs support MySQL 5.5, 5.6, 5.7, and 8.0 DB instances. HA clusters are not supported.
- The username and password required for adding the login are the database username and password.
- The port is the port of your self-built DB instance. Ensure that the security group of the ECS allows access to this port. To avoid conflicts with reserved ports, you are advised to set a port number ranging from 1024 to 65535.
- To delete the password, you can modify or delete the login.

Click **OK**.

----End

5 Logging In to a Database

Scenarios

On the DAS console, log in to a DB instance for database operations and management.

Procedure

Step 1 Log in to the DAS console.

Step 2 In the login list of Development Tool, locate the DB instance you want to log in to and click **Log In** in the **Operation** column.

Select an instance type to display your desired DB instances.

----**End**

6 Modifying Login Information

Scenarios

DAS allows you to modify login information, such as username and password. You can also choose whether to remember password, or enable or disable periodic metadata collection.

Procedure

- Step 1** In the login list of Development Tool, locate the DB instance for which you want to modify its login information and click **Modify** in the **Operation** column.
- Step 2** On the displayed page, modify the information as required.
- If you do not want to modify the information, click **Cancel**.
 - If you want to modify the information, make the modifications, confirm the settings, and click **OK**.

----End

7 Deleting Login Information

Scenarios

Delete the login information you no longer need.

NOTICE

A login represents an entry for managing databases. Deleted logins have no impact on your database performance but cannot be recovered, exercise caution when you perform this operation.

Procedure

Step 1 In the login list of Development Tool, locate the DB instance for which you want to delete its login information and click **Delete** in the **Operation** column.

Step 2 In the displayed dialog box, click **Yes**.

----End

8 Logging In to Databases Shared by Others

Scenarios

You have a database instance and want to share it with another account or another user of the same account, but you do not want this other account or user to know the password of the shared login, or view your database instances besides the shared one.

Prerequisites

- There are two accounts, or one account and two IAM users.
- You have obtained the ID of the IAM user.
- You have logged in to the DAS console.

Use Constraints

- Users having shared database logins cannot add databases or users.
- Shared databases do not have the data tracking, rollback, and background task features.

Procedure

- Step 1** Log in to the DAS console using the username and password of an IAM user.
- Step 2** On the overview page, click **Go to Development Tool**.
- Step 3** In the right pane, click the **Database Logins Shared by Others** tab.
- Step 4** Click **View My User ID**. On the displayed page, copy the user ID.
- Step 5** Log in to the DAS console using the username and password of the account.
- Step 6** On the overview page, click **Go to Development Tool**.
- Step 7** On the **My Database Logins** tab, locate the login you want to share and click the number in the **Additional Users** column.
- Step 8** On the displayed **View Additional Users** page, click **Add User**.

Step 9 On the displayed **Add User** page, select the adding method.

- **Manual input:** Enter the ID of the user that you want to share your database login with. If multiple users need to be added, separate their IDs with semicolons (;).
- **Synchronize IAM account:** Select another IAM user of the current account.
- **Synchronize EPS user:** Select the target users from the group of the enterprise project.

Step 10 Confirm the settings and click **OK**.

Step 11 Return to the DAS console and check whether the users are successfully added.

Step 12 Click **Log In** to log in to the shared database.

----End

9 MySQL DB Instance Management (Development Tool)

9.1 Homepage

9.1.1 Overview

Logging In to DAS

Step 1 Obtain the login permission and log in to the DAS console.

Step 2 In the login list of Development Tool, choose **RDS DB instances** and **MySQL** from the drop-down lists, locate the DB instance you want to log in to, and click **Log In** in the **Operation** column.

----End

Function Overview

Table 9-1 Functions

Function	Description
Top menu bar	<ul style="list-style-type: none">• SQL Operations• Database Management• Import and Export• Structure Management• Data Scheme• Background Tasks• Account Management
Account information and logout	Displays the current account information and a logout button.

Function	Description
DB instance information	Displays the current database information, such as the DB engine logo, DB instance name, and DB engine version.
Database list	Displays the databases of the current DB instance, including user databases and system databases. <ul style="list-style-type: none">• Allows you to search for a database by name.• Allows you to create databases.

9.1.2 Creating a Database

Procedure

- Step 1** In the database list of the **Home** page, click **Create Database**.
- Step 2** On the displayed page, enter a database name and select a character set.
- Step 3** Click **OK**. The database you create appears in the database list.

----End

9.1.3 Dropping a Database

Procedure

- Step 1** In the database list, locate the database you want to delete and choose **More > Drop** in the **Operation** column.
- Step 2** In the displayed dialog box, click **Yes**.

----End

9.1.4 Data Dictionary

Prerequisites

The current database contains tables.

Procedure

- Step 1** In the database list, locate the target database and click **Data Dictionary** in the **Operation** column.
- Step 2** On the displayed page, view the content in **Tables** and **List Table Schema** of the current database.
- Step 3** Click **Export in PDF** to download it to your local PC if needed.

----End

9.2 SQL Operations

DAS manages database objects through SQL statements, facilitating user operations.

9.2.1 SQL Window

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**.

Step 2 In the navigation pane, select a database, and then specify a table or view.

- **Execute SQL:** executes SQL statements.
- **Execute SQL Plan:** reports the execution of SQL statements to facilitate troubleshooting and optimize SQL processing performance.
- **SQL Favorites:** allows you to add, view, and manage frequently used SQL statements.
- **Format SQL:** improves the readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics.

The formatting takes effect for all the SQL statements in the SQL window. You cannot format only the selected statements.

- **SQL Input Prompt:** helps you to select your desired database, table, or field name as prompted to quickly enter statements in the SQL window.
- **Overwrite/Append Mode:**
 - **Append Mode:** Each time a SQL statement is executed, the new result set is appended to the previous one.
 - **Overwrite Mode:** Each time a SQL statement is executed, the new result set overwrites the previous one.
- **Full Screen:** shows SQL statements on a full screen.

Step 3 After SQL statements are executed, view SQL execution status on the **Executed SQL Statements, Messages, and Result Set** tab pages.

----End

9.2.2 Executing SQL Statements

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**. In the SQL window, select a database from the navigation pane.

Step 2 Enter SQL statements in the SQL window and click **Execute SQL**.

- Enabling **SQL Input Prompt** is recommended. It helps you select the required database, table, or field name to enter statements in the SQL window.
- To protect query result sets from being displayed as garbled characters caused by UTF-8, the default encoding format, select an encoding format as needed.

- To execute multiple SQL statements at a time, separate them using semicolons (;). In this case, click **Full Screen** to view the logics in SQL statements clearly.
 - To execute some of the SQL statements, select the statements before executing them.
 - To execute all the SQL statements, do not select any SQL statements or select all of them.

Step 3 View the SQL execution details on **Executed SQL Statements** and **Messages** tab pages, including the SQL execution history, execution status, impact scope, execution progress, and elapsed time.

Step 4 On the **Result Sets** tab, view the execution details. You can perform the following operations on the result sets:

- **Copy Row/Copy Column:** copies a row or column for reuse.
- **Column Settings:** customizes the columns to display when there are a large number of columns in the query result.
- **Binary to Hexadecimal:** converts the binary data in the result set into hexadecimal data for display.
- **Refresh:** refreshes the changed data.
- **Row Details:** displays the column field name, value, and type of the selected row.
- **Add:** adds an empty row to the result set.
- **Submit:** views the SQL statements to be modified. After you click **OK**, the result set is updated to the latest.
- **Delete Row:** deletes the selected row, including the data.
- **Export:** allows you to export the data in a SQL or CSV file. A maximum of 10,000 rows of data can be exported.
- **Export More:** redirects you to the data export page and allows you to export over 10,000 rows of data.

 **NOTE**

- If the result set involves a view, you cannot edit the data.
- If the type of the result set is metadata, the data cannot be edited or displayed on multiple pages.
- If the result set involves multiple tables, you cannot edit the data.
- If the result set involves only one table and does not include all the primary key columns of the table or no primary key is found, you cannot edit the data.
- Virtual tables (for example, tables generated during SQL execution) cannot be edited.

----End

9.2.3 Executing SQL Plan

Scenarios

This section describes how to analyze the execution efficiency of the current SQL statement.

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL Window**. In the SQL window, select a database from the navigation pane.
- Step 2** Enter SQL statements in the SQL window and click **Execute SQL Plan**.
- On the **Executed SQL Statements** tab page, view the historical SQL executions.
 - On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.
 - On the **Execution Plan** tab page, view the execution status for each SQL statement.

----End

9.2.4 SQL Favorites

Scenarios

DAS allows you to add frequently used SQL statements and view and manage SQL statements.

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL Window**.
- Step 2** In the SQL window, click **SQL Favorites**.
- **Add SQL Statement:** allows you to add your desired SQL statements. Specifically, specify the title, select the application scope, enter SQL statements, and click **Save**.
 - **Insert SQL Statement:** allows you to insert the SQL statements you have added. If there are no SQL statements stored, this option will not be displayed.
 - **Manage SQL Statements:** allows you to manage frequently used SQL statements, including adding, editing, and deleting statements.

----End

9.2.5 SQL History

Prerequisites

Enable **Save Executed SQL Statements** after you log in to the DAS console.

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL History**.
- Step 2** On the displayed page, search for execution information about the target SQL statement by time range, database name, or keyword.
- To access the **Database Management** page, click a database name.

- To copy and use a SQL statement, click it in the **SQL Statement** column.
- To execute a SQL statement, click **Open in SQL Window** in the **Operation** column.

 **NOTE**

After you enable **Save Executed SQL Statements**, executed SQL statements will be recorded in the DAS management host.

----End

9.3 Database Management

Function Overview

Table 9-2 Functions

Function	Description
Sidebar	Consists of six types of objects (tables, views, stored procedures, events, triggers, and functions).
Database information	Displays the name, IP address, character set, SQL window, and data dictionary of the current database.
Metadata Collection	Allows DAS to automatically collect only the data such as database names, table names, and field names in an instance (data in your tables is not included). NOTE <ul style="list-style-type: none">• If there are too many tables, the system will neither collect metadata nor display tables to minimize impacts on database performance.• To ensure stable performance, DAS will temporarily stop collecting data when the instance QPS is over 10,000 or TPS over 1,000.
List details	Displays the operation area for each type of object.

 **NOTE**

You can go to the **Database Management** page in either of the following ways:

- Choose **Database Management** on the top menu bar.
- In the database list of the **Home** page, locate the target database and click **Manage** in the **Operation** column.

Object List

You can manage tables, views, stored procedures, events, triggers, and functions of the MySQL DB instances. The main operations include creating, querying, and modifying objects.

- For details on how to manage tables, see [Table Management](#).
- For details on how to manage views, see [View Management](#).
- For details on how to manage events, see [Event Management](#).
- For details on how to manage stored procedures, see [Stored Procedure Management](#).
- For details on how to manage functions, see [Function Management](#).
- For details on how to manage triggers, see [Trigger Management](#).

9.3.1 Table Management

9.3.1.1 Creating a Table

Scenarios

A MySQL data table consists of basic information, field/column information, virtual columns, indexes, and foreign keys. The virtual columns, indexes, and foreign keys are optional and can be configured based on service requirements.

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Tables** and click **Create Table**.
- Step 2** On the displayed page, specify the details under **Basic Information** (**Table Name** is mandatory).
- Step 3** Click **Advanced Settings**, specify table parameters based on service requirements. These parameters are mandatory. To learn more about partitioned table, see the following note. Then click **Next**.

 **NOTE**

- Table partitioning is to divide a large table into several small tables based on conditions. Different rows in the table can be allocated to different physical partitions. Partitioned tables are not recommended because there are many constraints on MySQL partitioned tables.
- If you need to create a partitioned table, you can create one by referring to the following example. The supported partitioning methods are RANGE, LIST, COLUMNS, KEY, and HASH.

For example, if you want to create partitioned table **employees**, enter the following content for **Partition Definition**:

Example for creating a partitioned table:

```
CREATE TABLE employees (  
id INT NOT NULL,  
fname VARCHAR(30),  
lname VARCHAR(30),  
hired DATE NOT NULL DEFAULT '1970-01-01',  
separated DATE NOT NULL DEFAULT '9999-12-31',  
job_code INT NOT NULL,  
store_id INT NOT NULL  
)  
PARTITION BY RANGE (store_id) (  
PARTITION p0 VALUES LESS THAN (6),  
PARTITION p1 VALUES LESS THAN (11),  
PARTITION p2 VALUES LESS THAN (16),  
PARTITION p3 VALUES LESS THAN (21)  
);  
Partition definition:  
PARTITION BY RANGE (store_id) (  
PARTITION p0 VALUES LESS THAN (6),  
PARTITION p1 VALUES LESS THAN (11),  
PARTITION p2 VALUES LESS THAN (16),  
PARTITION p3 VALUES LESS THAN (21)  
)
```

Step 4 On the **Column** page, click **Add** and set **Column Name**, **Type**, **Length**, **Nullable**, **Primary Key**, **Comment**, and **Extended Information** as needed.

- If you do not need to add virtual columns, indexes, or foreign keys, click **Create** at the bottom of the page. In the displayed **SQL Preview** dialog box, click **Execute**.

 **NOTE**

- Only MySQL 5.6.5 and later support default DATETIME values.
- In versions earlier than MySQL 5.6.5, leave default values blank. Otherwise, an error occurs.
- If you need to add virtual columns, indexes, and foreign keys, click **Next**. Then, set the column name, type, length, nullable, primary key, expression, storage type, comment, and extended information. If you need to set the table index or foreign key, click **Next**. After the setting, click **Create**.

 **NOTE**

When you create a foreign key, the type of columns in the referenced table must be the same as that of included columns, and must be the primary key or have a unique index.

Step 5 In the **SQL Preview** dialog box, click **Execute** to create a table.

----End

 **NOTE**

- The length of a column name is limited. Enter no more than 64 characters for the MySQL engine.
- In the **Type** column, you can select only the parameters from the drop-down list box.
- In the **Length** column, you can change the length for some types of columns.
- If **Primary Key** is selected, **Nullable** will be grayed out.
- **Auto Increment** can be set for one column only. When it is selected, **Primary Key** must be selected, and a default value cannot be set.

9.3.1.2 Opening a Table

Scenarios

If a table has primary keys, you can add, delete, modify, or query table data as you do in Excel.

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Tables**, locate the target table, and click **Open** in the **Operation** column.
- Step 2** On the table details page, double-click a cell to edit data. After adding or editing data, submit and save the changes.

Table 9-3 Functions

Item	Description
Where Condition	Filters records.
Copy Row	Copies data of the selected row. Specifically, double-click the target row and then click Copy Row .
Copy Column	Allows you to copy all data in a column by selecting this column from the Copy Column drop-down list.
Column Settings	Allows you to set the columns you want to display.
Binary to Hexadecimal	Prevents binary data in columns from being displayed as garbled characters. If there is binary data in columns, you must select this option.
Refresh	Allows you to update table data manually.

Item	Description
Row Details	Shows the detailed information of a specified row.
Add	Allows you to add rows.
Submit	Allows you to submit and save the changes to data.
Delete Row	Allows you to delete the selected rows.
Export	Exports a maximum of 10,000 data records as CSV or SQL files.

----End

9.3.1.3 Viewing Table Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** page, select **Tables**, locate the target table, and click **View** in the **Operation** column.
- Step 3** In the displayed dialog box, view basic table information and SQL statements for creating the database.

----End

9.3.1.4 Altering a Table

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** page, select **Tables**, locate the target table, and click **Alter** in the **Operation** column.
- Step 3** Modify the table information as required. After the modification is complete, click **Alter**.
- Step 4** In the **SQL Preview** dialog box, click **Execute**.

----End

9.3.1.5 Renaming a Table

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** page, select **Tables**, locate the target table, and click **Rename** in the **Operation** column.

Step 3 In the displayed dialog box, enter a new table name and click **OK**.

----End

9.3.1.6 Clearing a Table

NOTICE

Cleared tables cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 On the top menu bar, choose **Database Management**.

Step 2 On the displayed **Objects** page, select **Tables**, locate the target table, and choose **More > Clear** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

9.3.1.7 Dropping a Table

NOTICE

Dropped tables cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 On the top menu bar, choose **Database Management**.

Step 2 On the displayed **Objects** page, select **Tables**, locate the target table, and choose **More > Drop** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

9.3.1.8 Maintaining a Table

Scenarios

While working with MySQL databases, you do a lot of changes such as data insert, update, and deletion, which may cause table fragmentation. As a result, the database server performance is deteriorated. To handle this, periodic maintenance is required.

Functions

- Check
The CHECK TABLE statement adds a read-only lock to the table.
Allows you to check whether there are errors in database tables using the CHECK TABLE statement. You can check a table with any of the following methods:
 - **General:** Scan rows to verify that deleted links are valid. Alternatively, calculate a key checksum for the rows and verifies the validity using the obtained checksum.
 - **Fast:** Check only tables that have not been closed properly.
 - **Quick:** Do not scan rows or check for incorrect links.
 - **Changed:** Check only tables that have been changed since the last check or tables that have not been closed properly.
 - **Extended:** Search for keywords in each row. This ensures that the table is 100% consistent, but takes a long time.
- Repair
Allows you to use the REPAIR TABLE statement to repair possibly corrupted or incorrect tables. You can repair tables using any of the following three methods:
 - **Check:** a simple repair, which repairs data and index files.
 - **Quick:** the quickest repair, which repairs only index files, but not data files.
 - **Extended:** the slowest repair, which creates indexes row by row to repair data and index files.

Procedure

The following uses the **Check** operation as an example to explain how to maintain tables.

Step 1 On the top menu bar, choose **Database Management**.

Step 2 On the displayed **Objects** page, select **Tables**, locate the target table, and choose **More > Maintain > Check > Check** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

9.3.2 View Management

9.3.2.1 Creating a View

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Views**.

Step 2 In the upper left corner of the page, click **Create View**.

Step 3 On the displayed page, specify information such as view name and definer, choose security, check option, and algorithm from drop-down lists as required, and click **Create**.

Step 4 In the displayed dialog box, click **Execute**.

----End

9.3.2.2 Opening a View

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Views**, locate the target view, and click **Open** in the **Operation** column.

Step 2 Check the view information on the details page or click **Where Condition**.

Table 9-4 Functions

Item	Description
Where Condition	Filters records.
Copy Row	Allows you to copy data of a selected row. Specifically, select the target row and click Copy Row .
Copy Column	Allows you to copy data of a selected column. Specifically, click Copy Column and select a column from the drop-down list.
Copy	Allows you to copy a piece of data. Specifically, hover the cursor on the data and click Copy .
Column Settings	Allows you to set the columns you want to display.
Binary to Hexadecimal	Prevents binary data in columns from being displayed as garbled characters. If there is binary data in columns, you must select this option.
Refresh	Allows you to update table data manually.
Row Details	Shows the detailed information of a specified row.
Add	Allows you to add rows.
Submit	Allows you to submit and save the changes to data.
Delete Row	Allows you to delete the selected rows.
Export	Exports a maximum of 10,000 data records as CSV or SQL files.

 NOTE

View data can be only queried, but not be edited, because views do not have primary keys.

----End

9.3.2.3 Altering a View

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Views**, locate the target view, and click **Alter** in the **Operation** column.
- Step 2** On the displayed page, modify the information including the security, algorithm, and view definition statement and click **Alter**.
- Step 3** In the displayed dialog box, click **Execute**.

----End

9.3.2.4 Dropping a View

NOTICE

Dropped views cannot be recovered. Exercise caution when performing this operation.

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Views**, locate the target view, and click **Drop View** in the **Operation** column.
- Step 2** In the displayed dialog box, click **Yes**.

----End

9.3.3 Stored Procedure Management

9.3.3.1 Creating a Stored Procedure

Overview

A stored procedure is a set of SQL statements used to implement specific functions. After being compiled, stored procedures are stored in databases. You can execute the stored procedures by specifying their names and related parameters.

Scenarios

A stored procedure consists of a set of SQL statements that can complete specific tasks or process complex services. This section describes how to create a stored procedure. You can use loop statements in a stored procedure to insert objects repeatedly.

Prerequisites

- You have obtained the CREATE ROUTINE permission.
- You have obtained the Execute permission.

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Stored Procedures**, and click **Create Stored Procedure**.
 - Step 2** On the displayed page, enter the stored procedure name and description, and click **OK**.
 - Step 3** On the displayed page, click **Option**, set required parameters, and click **OK**. Then, save and execute the stored procedure.
 - Step 4** View the execution status in the **Message** area.
- End

9.3.3.2 Altering or Executing a Stored Procedure

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Stored Procedures** on the left, locate the target procedure, and click **Alter or Execute** in the **Operation** column.
 - Step 2** On the displayed page, click **Option** to set parameters as required and click **OK**. Then, click **Save**.
 - Step 3** Execute the stored procedure and view the execution status in the **Message** area.
- End

9.3.3.3 Dropping a Stored Procedure

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Stored Procedures** on the left, locate the target stored procedure, and click **Drop** in the **Operation** column.
- Step 2** In the displayed dialog box, click **Yes**.

NOTICE

Dropped stored procedures cannot be recovered. Exercise caution when performing this operation.

----End

9.3.3.4 Viewing Stored Procedure Details

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Stored Procedures**.

Step 2 In the right pane, click **View Details**.

----End

9.3.4 Event Management

9.3.4.1 Creating an Event

Scenarios

This section describes how to create an event to periodically respond to specific operations.

When **event_scheduler** for RDS DB instances is set to **ON** and the event function is enabled, you can perform operations on events.

When **event_scheduler** is set to **OFF**, but the event function is enabled, you can only create events and the events do not take effect.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Events**, and click **Create Event**.

Step 2 Specify an event name (mandatory) and event definition statements (mandatory), set the execution time, status, and comment, and click **Create**.

- **Event Definition Statements**

Indicates the SQL statements to be executed when a scheduled event is triggered.

- **Dropped upon expiration**

- Indicates that the events to be executed at a fixed point in time will be deleted after they are executed.

- Indicates that the events to be periodically executed will be deleted at the specified end time.

Step 3 In the displayed dialog box, click **Execute**. If no error is reported, the event takes effect.

----End

9.3.4.2 Altering an Event

Scenarios

This section describes how to alter an event.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Events**, locate the target event, and click **Alter** in the **Operation** column.

Step 2 On the displayed page, modify the required information and click **Alter**.

Step 3 In the displayed dialog box, click **Execute**.

----End

9.3.4.3 Dropping an Event

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Events**, locate the target event, and click **Drop** in the **Operation** column.

Step 2 In the displayed dialog box, click **Yes**.

----End

9.3.5 Trigger Management

9.3.5.1 Creating a Trigger

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** page, choose **Triggers**, and click **Create Trigger**.

Step 2 On the displayed page, set required parameters and edit trigger definition statements. Then, click **Create**.

Step 3 In the displayed dialog box, click **Execute**.

----End

9.3.5.2 Altering a Trigger

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Triggers**, locate the target trigger, and click **Alter** in the **Operation** column.
- Step 2** On the displayed page, modify the required parameters and click **Alter**.
- Step 3** In the displayed dialog box, click **Execute**.

----End

9.3.5.3 Dropping a Trigger

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Triggers**, locate the target trigger, and click **Drop Trigger** in the **Operation** column.
- Step 2** In the displayed dialog box, click **Yes**.

NOTICE

Dropped triggers cannot be recovered. Exercise caution when performing this operation.

----End

9.3.6 Function Management

9.3.6.1 Creating a Function

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Functions**, and click **Create Function**.
- Step 2** In the displayed dialog box, specify the function name, returned value, and description, and click **OK**.
- Step 3** On the displayed page, edit the function body.
- Step 4** In the upper left corner, click **Option**. In the displayed dialog box, set required parameters and click **OK**.
- Step 5** Click **Save**, or click **Execute** and view the status in the **Message** area.

----End

9.3.6.2 Altering or Executing a Function

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the **Objects** page, select **Functions**, locate the target function, and click **Alter or Execute Function** in the **Operation** column.
- Step 2** On the displayed page, click **Option** and set parameters as required. Then, click **Save** to save the settings or click **Execute** to invoke the function.
- End

9.3.6.3 Dropping a Function

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** page, select **Functions**, locate the target function, and click **Drop Function** in the **Operation** column.
- Step 2** In the displayed dialog box, click **Yes**.

NOTICE

Dropped functions cannot be recovered. Exercise caution when performing this operation.

----End

9.3.6.4 Viewing Details of a Function

Procedure

- Step 1** In the top menu bar, choose **Database Management**. In the navigation pane on the left, choose **Functions**, locate the function whose details you want to view and click **View Function** in the **Operation** column.
- Step 2** In the displayed dialog box, view function details and click **Close**.
- End

9.4 Import and Export

9.4.1 Data Import

Scenarios

Import data from your local PC or OBS bucket for data backup and migration.

Operation Description

Import data into a table for backup or migration. If you import a CSV or SQL file, the file must have the same data type as the target table.

Constraints

- Only one file that is no larger than 1 GB can be imported at a time.
- Data files in the CSV or SQL format can be imported.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.

Procedure

Step 1 On the top menu bar, choose **Import and Export > Import**.

Step 2 Import a file from your local PC or an OBS bucket.

- From your local PC

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

For security purposes, imported files are stored in OBS buckets.

NOTE

- To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
 - If you select **Delete the uploaded file upon an import success**, the file you upload will be automatically deleted from the OBS bucket after being imported to the destination database.
- From an OBS bucket

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

NOTE

The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 4 View the import progress in the task list or check task details.

----End

9.4.2 Data Export

Prerequisites


You have created a user database on the MySQL database management page. For details, see [Creating a Database](#).

Scenarios

DAS allows you to export a large amount of data at a time to facilitate data query or to back up data for migration. This section describes how to export data.

Procedure

Step 1 On the login page, enter your account and password to log in to the management console.

Step 2 Click  in the upper left corner and select a region and project.

Step 3 Under **Database**, click **Data Admin Service**.

Step 4 Locate the target database and click **Log In** in the **Operation** column. The homepage of the MySQL database management is displayed.

Step 5 On the top menu bar, choose **Import and Export > Export**.

Step 6 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Step 7 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

NOTE

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- Databases are classified into user databases and system databases. System database cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.
- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 8 After settings are complete, click **OK**.

Step 9 In the task list, view the task ID, type, status, and progress.

Step 10 Click **Details** in the **Operation** column to view task details.

----End

9.5 Structure Management

9.5.1 Table Structure Comparison and Synchronization

Scenarios

When you perform a migration or verification, you can check the structure difference by comparing and synchronizing table structures.

Procedure

- Step 1** On the top menu bar, choose **Structure Management > Table Structure Comparison and Synchronization**.
- Search for tasks by task status and view task details.
- Step 2** On the displayed page, click **Create Task** in the upper left corner.
- Step 3** On the displayed page, specify the source database, target instance, target database, error tolerance, and synchronization type. Click **Next**.
- Step 4** View comparison objects and click **Compare**.
- You can also select the items you want to skip and click **Skip** to cancel the comparison.
- Step 5** View the comparison progress. In the comparison item list, click **View Logs** in the **Operation** column to obtain comparison details. You can also click **Download DDL** as needed.
- Step 6** Click **Next**. On the **Synchronize** page, view the basic information about this comparison task, such as the source instance, source database, target instance, target database, and synchronization type, and specify the items to be synchronized.
- Step 7** Skip synchronization items that may cause high risks and click **Synchronize**.
- Step 8** After synchronization is complete, click **View Logs** in the **Operation** column to obtain comparison details. You can download DDL as required.
- End

9.6 Data Scheme

9.6.1 Data Generator

Scenarios

During the functional testing of a program, a large amount of test data complying with specific rules needs to be inserted into the database. DAS allows you to automatically, efficiently generate test data based on specific rules.

Procedure

- Step 1** On the top menu bar, choose **Data Scheme > Data Generator**.
- Step 2** On the displayed page, click **Create Task**. In the displayed dialog box, configure the required parameters.
- Rows to Generate
A maximum of 1,000,000 rows of data can be generated.

- **Conflict Policy**
If you select **Skip**, the system skips data rows in conflict and continues generating data. If you select **Replace**, the system replaces existing rows with new ones that have the same primary key.
 - **Generation type**
You can set the rules for randomly generated data based on the column settings. For example, if the column type is time, you can set the start time, end time, and format, or select **Based on current time**.
- Step 3** Click **Preview** to check whether the generated data is as expected. If not, modify the settings and click **Generate**. Alternatively, click **Generate** with no modification.
- Step 4** In the task list, locate the created task and click **Details**. You can also delete the task as required.

----End

9.6.2 Data Tracking and Rollback

Scenarios

- Audit core data changes, collect change statistics, and view sensitive information. For example, you can use this function to query bank statements, statistics on new orders, and key configuration changes.
- Roll back the misoperation to the state before it is operated. For example, you can use the rollback function when the WHERE condition is not added during DBA configuration update, configuration data is deleted by mistake, or a large amount of dirty data and ripple effects are generated due to program bugs.

Procedure

- Step 1** On the top menu bar, choose **Data Scheme > Data Tracking and Rollback**.
- Search for tasks by time range or database name.
 - You can delete tasks as required by selecting check boxes next to task IDs. The deletion operation is synchronized to DAS.
 - In the task list, click the task ID to view task details.
 - In the task list, locate the target task and click **View Task** in the **Operation** column. Then, you can view task details under **Task Information**, **Log Search**, and **Rollback Task List**.

NOTE

A created task will expire in 15 days. Once the task expires, the system automatically retrieves and deletes the changes in DAS.

- Step 2** On the displayed page, click **Create Tracking Task**. In the displayed dialog box, specify task details, including the task name, time range, and database name.

NOTE

Users who create a tracking task for the first time need to view and agree to the agreement.

- Step 3** Click **Precheck** to obtain the operation details at a specific point in time.

 **NOTE**

- During the pre-check, binlog files are displayed based on time range. RDS DB instances with the backup function enabled periodically back up the binlog file to the OBS bucket. The backup delay is no more than five minutes.
- When you create a tracking task, there may be no changes in the latest five minutes. In this case, you can create a task again later.

Step 4 Click **Read Logs** to obtain log details.

When reading logs, the system initiates binlog parsing and stores log changes for filtering and displaying data.

 **NOTE**

- You can search for logs only after all logs are successfully read.
- If a new task is started before the last tracking task is completely read, it is normal that the log start time is later than the end time.

Step 5 Click **Log Search** to obtain details about the event changes.**Step 6** Create a rollback task if you need to roll back multiple events. Specifically, click **Create Rollback Task** on the **Log Search** tab page. In the displayed dialog box, specify event IDs and storage of rollback statements, set advanced settings as required, and click **OK**. **NOTE**

- The start and end event IDs are the ones in the task list and must be entered in ascending order.
- Combing changes
Changing a record (primary key) three consecutive times (for example, 1->2->3->4) equals the change of 1 to 4 (1->4).
- Statement Type
Generate event rollback SQL statements: generates SQL statements with reverse changes based on the images before and after data change.
Obtain the original data before changes: generates rollback tables and insert statements by mirroring before data change.

Step 7 In the rollback task list, view the current rollback task information or create a rollback task.

- Locate the target rollback task and click **View Details** in the **Operation** column to view task details.
- In the **Operation** column, click **Download** to download the compressed data package of the task.
- Enter a task ID in the search box in the upper right corner of the task list to search for the required task.
- In the **Operation** column, you can sort tasks by task ID, start event ID, end event ID, file size, and status.

 **NOTE**

Changes (such as insert->delete, delete->insert, update->update) on the same record will be combined or canceled. So, the generated file may have no rollback SQL statements or original data.

----End

9.7 Background Tasks

9.7.1 Task Scheduling

Scenarios

DAS allows you to execute SQL statements by scheduling tasks. Scheduling types include upon submission, scheduled, and periodic. You can select the scheduling type when creating a task. Error control and transaction control can be performed on SQL statements during scheduling, and task dependency chains can be set for dependent SQL statements.

Procedure

Step 1 On the top menu bar, choose **Background Tasks > Task Scheduling**.

 **NOTE**

You can create a scheduling task only after agreeing to save the SQL statements in the database.

Step 2 On the **Scheduling Tasks** page, click **Create Task**.

Step 3 On the displayed page, set the task name, scheduling type, and execution time tolerance.

Upon submission: Indicates that a scheduled task is executed immediately after being submitted. The task is executed only once.

Scheduled: Indicates that a task is executed at a scheduled point in time after being submitted. The task is executed only once.

Periodic: Indicates that a task is executed periodically at the specified time after being submitted.

Execution Time Tolerance: Indicates the length of time (in seconds) to wait for a task to be executed before it is no longer scheduled. However, this will not affect the task if it is scheduled again later.

Step 4 Under **Add SQL Statements**, click **Add**. On the displayed page, set parameters as needed and click **Save**. Then, submit the task.

 **NOTE**

After a task is executed, it is automatically displayed in the list of finished tasks.

Step 5 In the scheduling task list, view periodic tasks.

You can search for tasks by status, scheduling type, task ID, or task name.

- **Details:** Click it to view the task information.

View the execution logs and group details under the group list and view the SQL statements on the group details page.

Click the name of a periodic task to go to the **Task Information** page. Then, click **Schedule Now** to trigger a scheduling immediately. Additionally, view the execution details on the **Scheduling Execution Records** page.

- **Execution Records:** Click it to view the task execution details and logs.
- **View Logs:** Click it to view log details.
- **Terminate:** Click it to stop a scheduled task. A stopped task will be moved to the **Finished Tasks** list.
- **Pause:** Click it to pause a task. The task status changes from **Scheduling** to **Paused**. You can also click **Resume** to restore the scheduling.

Step 6 On the **Finished Tasks** page, view tasks that are executed upon submission or at scheduled points in time.

You can search for tasks by status, scheduling type, task ID, or task name.

- **Details:** Click it to view the task information.
View the execution logs and group details under the group list, and view or modify the SQL statements on the group details page.
- **Execution Records:** Click it to view the task execution details, group execution status, SQL statements, and group logs.
- **View Logs:** Click it to view log details.
- **Delete:** Click it to delete a task from the database.

----End

9.7.2 Scheduled Backup

Scenarios

Scheduled backup allows you to back up important database tables and data or SQL result sets so that data can be restored timely in case of a data loss.

Procedure

Step 1 On the top menu bar, choose **Background Tasks > Scheduled Backup**.

Step 2 On the task list page, click **Set AK/SK** in the upper left corner.

NOTE

Storage: The system stores encrypted files in OBS buckets.

AK and **SK:** Access key ID (AK) and secret access key (SK) are required when cloud services are accessed using development tools (APIs, CLI, and SDKs).

Step 3 Select an OBS bucket, enter the AK and SK, and click **Test AK/SK**.

Step 4 After the test succeeded, click **OK**.

Step 5 On the task list page, click **Create DB Backup Task**.

Step 6 Enter a task name, select a task type, and select the database and storage information.

Step 7 Confirm the settings and click **OK**.

- Step 8** On the task list page, click **Details** to view the task details.
- Step 9** On the task list page, click **Create SQL Result Set Backup Task**.
- Step 10** Enter a task name, select a task type, and set the time, backup file format, database, and storage information.
- Step 11** Confirm the settings and click **OK**.
- Step 12** On the task list page, click **Execution Records** to view the backup status.
- Step 13** Click **Details** to view the backup logs.
- Step 14** Click **Download** to download the backup file.
- Step 15** On the task list page, click **Terminate** or **Pause** to stop or pause a task.
- End

9.8 Account Management

9.8.1 Database Account Security

To ensure that RDS MySQL databases run properly and prevent account information from being deleted by mistake, DAS hides the following accounts of MySQL database users.

Table 9-5 Accounts to be hidden

Account	Description
mysql.session@localhost	Internal database account configured in MySQL 5.7 or later.
mysql.sys@localhost	Internal database account configured in MySQL 5.7 or later.
mysql.infoschema@localhost	Database account.
root@%	Account configured when you create an RDS MySQL instance.
rdsAdmin@localhost	Management account, which has the highest superuser permission and is used to query and modify DB instance information, rectify faults, as well as migrate and restore data.
rdsBackup@localhost	Backup account, which is used to back up background data.
rdsMetric	Metering account, which is used by watchdog to collect database status data.

Account	Description
rdsRepl	Replication account, which is used to synchronize data from primary DB instances to standby ones or read replicas.
rdsProxy	Proxy account, which is used for authentication when the database is connected using the read/write splitting address. This account is automatically created when read/write splitting is enabled.
drsFull	Account added to the destination database when you start a full migration task.
drsIncremental	Account added to the destination database when you start an incremental migration task.

9.8.2 Creating a User

Scenarios

Multiple users with different permissions can be created to access a DB instance or database, but the permissions of these users cannot exceed the operation permissions of the account.

Procedure

- Step 1** On the top menu bar, choose **Account Management > User Management**.
- Step 2** On the displayed page, click **Create User**. Then, configure required information, such as basic information, advanced settings, global permissions, object permissions, and roles, and click **Save**. In the displayed dialog box, click **OK**. For details, see [Table 9-7](#).

NOTE

The role function is available only to MySQL 8.0.

You can click **Add DAS IP Address** to add the IP address of the DAS server to the user's host. In the user list, you can view all created users, or locate a user and click **Edit** and **Delete** in the **Operation** column to manage it.

[Table 9-6](#) describes the user permissions briefly. For details about permission usages and explanations, visit the [official website](#).

NOTE

You can add multiple IP addresses and separate them with commas (.). For example, if you enter **192.%,193.%** for **Host**, two users with host addresses of 192.% and 193.% will be created, respectively. The following is an example result.

<input type="checkbox"/>	Username	Host	Global Permissions	Object Permissions	Operation
<input type="checkbox"/>	user	193.%	0	0	Edit Delete
<input type="checkbox"/>	user	192.%	0	0	Edit Delete

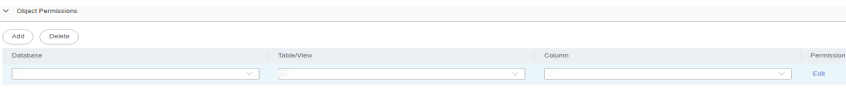
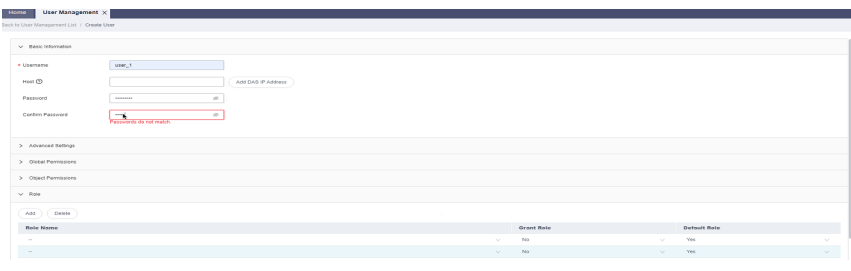
Table 9-6 Permission description

Permission	Description
SELECT	Permission to query a specified object, for example, a global or single database.
INSERT	Permission to insert data into a specified object.
UPDATE	Permission to update data of a specified object.
DELETE	Permission to delete data from a specified object.
CREATE	Permission to create databases and tables for a specified object.
DROP	Permission to delete databases, tables, and views of a specified object.
RELOAD	Permission to execute FLUSH .
PROCESS	Permission to execute SHOW PROCESSLIST to view all the processes.
GRANT	Permission to grant permissions to another account or reclaim permissions from another account.
REFERENCES	Permission to create foreign keys for a specified object.
INDEX	Permission to create or delete index for a specified object.
ALTER	Permission to execute ALTER TABLE for a specified object.
SHOW DATABASES	Permission to run the SHOW DATABASES command to show all databases.
CREATE TEMPORARY TABLES	Permission to execute CREATE TEMPORARY TABLE for a specified object.
LOCK TABLES	Permission to execute LOCK TABLES on the tables with the SELECT permission for a specified object.
EXECUTE	Permission to execute a stored procedure for a specified object.
REPLICATION SLAVE	Permission to allow the replica server to read binary log events from the source server.
REPLICATION CLIENT	Permission to query the location of the source server or replica server.
CREATE VIEW	Permission to create or alter a view for a specified object.
SHOW VIEW	Permission to execute SHOW CREATE VIEW for a specified object.
CREATE ROUTINE	Permission to create a stored procedure for a specified object.

Permission	Description
ALTER ROUTINE	Permission to alter or delete a stored procedure for a specified object.
CREATE USER	Permission to execute CREATE USER , DROP USER , RENAME USER , and REVOKE ALL PRIVILEGES for a specified object.
EVENT	Permission to create, modify, delete, or display the event scheduler for a specified object.
TRIGGER	Permission to execute a trigger for a specified object. Only users with this permission can create, delete, execute, or display triggers of a table.

Table 9-7 User information description

Parameter	Description
Username	Name of the new user
Host	Whitelisted IP address allowed to access the database server. If you leave this parameter empty or enter %, the user can access all IP addresses. If you enter multiple IP addresses, separate them with commas (,), for example, 192.%,193.% .
Password	Password of the user. The password: <ul style="list-style-type: none"> • Contains 8 to 32 characters. • Must contain at least three types of lowercase letters, uppercase letters, digits, and the following special characters: ~!@#%^*_-=+?,()& • Cannot be the username or the username backwards. • Cannot be a weak password.
Confirm Password	-
Max. Queries Per Hour	Maximum queries that this user can execute within an hour. If this parameter is left empty or set to 0 , there is no limit on the maximum queries.
Max. Updates Per Hour	Maximum updates that this user can execute within an hour. If this parameter is left empty or set to 0 , there is no limit on the maximum updates.
Max. Connections Per Hour	Maximum connections that this user can connect to the database server within an hour. If this parameter is left empty or set to 0 , there is no limit on the maximum connections.
Max. User Connections	Maximum concurrent connections to the server. If this parameter is left empty or set to 0 , there is no limit on the maximum concurrent connections.

Parameter	Description
SSL Type	Type of authenticating user requests to the server
Issuer	Authority that issues the X.509 certificate used for connecting users to the server
Subject	Subject of the X.509 certificate used for connecting users to the server
Algorithm	Method of encrypting connections between the user and the server
Global Permissions	Permissions granted to the user to perform operations on all databases in the instance
Object Permissions	Permissions granted to the user on specific objects For example, the following settings grant the permission to the user to query the ID of table user_test in database db_test . 
Role	Role of the user The role function is available only to MySQL 8.0. Role Name: You can select an existing username in this instance. Grant Role: Whether the new user can grant permissions of the selected role to another user. Default Role: Whether the new user can inherit permissions of the selected role. For example, the following settings indicate that user_2 can inherit permissions of user@ and can grant permissions of user@ to another user. 

----End

9.8.3 Editing User Information

Scenarios

This section describes how to edit user information, including the username, password, global permissions, object permissions, advanced settings, and roles.

NOTE

The role function is available only to MySQL 8.0.

Procedure

Step 1 On the top menu bar, choose **Account Management > User Management**. In the user list, locate the target user and click **Edit** in the **Operation** column.

NOTE

- Modifying permissions for multiple users cannot be modified at the same time. For example, if two users with the same username and different host addresses of 192.% and 193.% are created, modifying permissions of one user with the host address of 192.% does not affect the permissions of the other user.
- When you edit a user, adding host addresses to the user does not create multiple users. For example, if a user already has a host address, 192.%, adding host address 194.% to it does not create another user with the host address of 194.%. This operation only changes the host address of the user from 192.% to 192.%,194.%.

<input type="checkbox"/>	Username	Host	Global Permissions	Object Permissions	Operation
<input type="checkbox"/>	user	193.%	0	0	Edit Delete
<input type="checkbox"/>	user	192.%,194.%	0	0	Edit Delete

Step 2 On the displayed page, edit user information, set global and object permissions, and click **Save**. In the displayed dialog box, click **OK**.

----End

9.8.4 Deleting a User

NOTICE

Deleted users cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 On the top menu bar, choose **Account Management > User Management**.

Step 2 In the user list, locate the user that you want to delete and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

----End

10 RDS for PostgreSQL Instance Management (Development Tool)

10.1 Homepage

10.1.1 Overview

Logging In to a DB Instance

Step 1 Obtain the login permission and log in to the DAS console.

Step 2 In the login list of Development Tool, locate the DB instance you want to log in to and click **Log In** in the **Operation** column.

----End

Function Overview

Table 10-1 Functions

Function	Description
Top menu bar	<ul style="list-style-type: none">• SQL Operations• Database Management• Import and Export• Account Management
Account information and logout	Displays the current account information and a logout button.
DB instance information	Displays the current database information, such as the DB engine logo, DB instance name, and DB engine version.

Function	Description
Database list	Displays the databases of the current DB instance, including user databases and system databases. Allows you to search for a database by name. Allows you to create databases.

10.1.2 Creating a Database

Procedure

Step 1 In the database list of the **Home** page, click **Create Database**.

Step 2 On the displayed page, enter a database name and select a character set.

NOTE

You can execute the following SQL statement to query system table **pg_collation** and view character sets and their corresponding collations and collation types:

```
select pg_encoding_to_char(collencoding) as encoding,collname,collcollate,collctype from  
pg_collation ;
```

Step 3 Click **OK**. The database you create appears in the database list.

----End

10.1.3 Dropping a Database

Procedure

Step 1 In the database list, locate the database you want to delete and click **Delete** in the **Operation** column.

Step 2 In the displayed dialog box, click **Yes**.

----End

10.2 SQL Operations

10.2.1 SQL Window

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**.

Step 2 In the navigation pane, select a database, and then specify a table or view.

- **Execute SQL:** executes SQL statements.
- **Execute SQL Plan:** reports the execution of SQL statements to facilitate troubleshooting and optimize SQL processing performance.

- **SQL Favorites:** allows you to add, view, and manage frequently used SQL statements.
- **Format SQL:** improves the readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics.
The formatting takes effect for all the SQL statements in the SQL window. You cannot format only the selected statements.
- **SQL Input Prompt:** helps you to select your desired database, table, or field name as prompted to quickly enter statements in the SQL window.
- **Overwrite/Append Mode:**
 - **Append Mode:** Each time a SQL statement is executed, the new result set is appended to the previous one.
 - **Overwrite Mode:** Each time a SQL statement is executed, the new result set overwrites the previous one.
- **Full Screen:** shows SQL statements on a full screen.

Step 3 After SQL statements are executed, view SQL execution status on the **Executed SQL Statements**, **Messages**, and **Result Set** tab pages.

----End

10.2.2 Executing SQL Statements

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**. In the SQL window, select a database and a schema from the navigation pane.

Step 2 Enter SQL statements in the SQL window and click **Execute SQL**.

- You are advised to enable **SQL Input Prompt**. It helps you to select your desired database, table, or field name as prompted to quickly enter statements in the SQL window.
- To execute multiple SQL statements at a time, separate them using semicolons (;). In this case, click **Full Screen** to view the logics in SQL statements clearly.
 - To execute some of the SQL statements, select the statements before executing them.
 - To execute all the SQL statements, do not select any SQL statements or select all of them.

Step 3 View the executed SQL statements and the current execution status in the lower part of the page. On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.

Step 4 On the **Result Sets** tab, view the execution details. You can perform the following operations on the result sets:

- **Copy Row/Copy Column:** copies a row or column for reuse.
- **Column Settings:** customizes the columns to display when there are a large number of columns in the query result.
- **Refresh:** refreshes the changed data.

- **Row Details:** displays the column field name, value, and type of the selected row.

----End

10.2.3 Executing the SQL Plan

Scenarios

This section describes how to analyze the execution efficiency of the current SQL statement.

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**. On the displayed page, select a database and a schema from the navigation pane.

Step 2 Enter SQL statements in the SQL window and click **Execute SQL Plan**.

- On the **Executed SQL Statements** tab page, view the historical SQL executions.
- On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.
- On the **Execution Plan** tab page, view the execution status for each SQL statement.

----End

10.2.4 SQL Favorites

Scenarios

DAS allows you to add frequently used SQL statements and view and manage SQL statements.

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**.

Step 2 In the SQL window, click **SQL Favorites**.

- **Add SQL Statement:** allows you to add your desired SQL statements. Specifically, specify the title and enter SQL statements as required, and then click **Save**.
- **Insert SQL Statement:** allows you to insert the SQL statements you have added. If there are no SQL statements stored, this option will not be displayed.
- **Manage SQL Statements:** allows you to manage frequently used SQL statements, including adding, editing, and deleting statements.

----End

10.2.5 SQL History

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL History**.
- Step 2** On the displayed page, search for execution information about the target SQL statement by time range, database name, or keyword.
- To access the **Database Management** page, click a database name.
 - To copy and use a SQL statement, click it in the **SQL Statement** column.
 - To execute a SQL statement, click **Open in SQL Window** in the **Operation** column.

 **NOTE**

After you enable **Save Executed SQL Statements**, executed SQL statements will be recorded in the DAS management host.

----End

10.3 Database Management

Function Overview

Table 10-2 Functions

Function	Description
Database information	Displays the name, IP address, character set, and SQL window of the current database.
Schema list	Allows you to manage data with different schemas in a database.
Object list	Consists of five types of objects, respectively tables, views, stored procedures, triggers, and sequences.
Metadata Collection	Allows DAS to automatically collect only the data such as database names, table names, and field names in an instance (user data in your tables is not included). NOTE <ul style="list-style-type: none">• If there are too many tables, the system will neither collect metadata nor display tables to minimize impacts on database performance.• To ensure stable performance, DAS will temporarily stop collecting data when the instance QPS is over 10,000 or TPS over 1,000.
List details	Displays the operation area for each type of object.

 NOTE

You can go to the **Database Management** page in either of the following ways:

- Choose **Database Management** on the top menu bar.
- In the database list of the **Home** page, locate the target database and click **Manage** in the **Operation** column.

Object List

You can manage tables, views, stored procedures, triggers, and sequences of PostgreSQL databases. Graphical windows for creating objects are not available. To create an object, execute SQL statements in the SQL window.

- For details on how to manage tables, see [Table Management](#).
- For details on how to manage views, see [View Management](#).
- For details on how to manage stored procedures, see [Stored Procedure Management](#).
- For details on how to manage triggers, see [Trigger Management](#).
- For details on how to manage sequences, see [Sequence Management](#).

10.3.1 Table Management

10.3.1.1 Opening a Table

Scenarios

Tables in PostgreSQL databases cannot be edited because the tables do not have primary keys.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** on the left, locate the target table, and click **Open** in the **Operation** column.

Step 2 On the table details page, specify the rows to be displayed. You can also copy and reuse rows and columns.

Table 10-3 Functions

Item	Description
Where Condition	Filters records.
Copy Row	Copies data of the selected row. Specifically, double-click the target row and then click Copy Row .
Copy Column	Allows you to copy all data in a column by selecting this column from the Copy Column drop-down list.
Column Settings	Allows you to set the columns you want to display.

Item	Description
Refresh	Allows you to update table data manually.
Row Details	Shows the detailed information of a specified row.

----End

10.3.1.2 Viewing Table Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Tables**, locate the target table, and click **View** in the **Operation** column.
- Step 3** In the displayed dialog box, view basic table information.

----End

10.3.2 View Management

10.3.2.1 Opening a View

Scenarios

Views in PostgreSQL databases cannot be edited because the views do not have primary keys.

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Views** on the left, locate the target view, and click **Open** in the **Operation** column.
- Step 2** Check the view information on the details page or click **Where Condition**.

Table 10-4 Functions

Item	Description
Where Condition	Filters records.
Copy Row	Copies data of the selected row. Specifically, double-click the target row and then click Copy Row .
Copy Column	Allows you to copy all data in a column by selecting this column from the Copy Column drop-down list.
Column Settings	Allows you to set the columns you want to display.

Item	Description
Refresh	Allows you to update table data manually.
Row Details	Shows the detailed information of a specified row.

----End

10.3.2.2 Obtaining View Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Views** on the left, locate the target view, and click **View Details** in the **Operation** column.
- Step 3** In the displayed dialog box, view the SQL statement used to create the view.

----End

10.3.3 Stored Procedure Management

10.3.3.1 Viewing Stored Procedure Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Stored Procedures**.
- Step 2** In the right pane, click **View Details**.

----End

10.3.4 Trigger Management

10.3.4.1 Viewing Trigger Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Triggers**.
- Step 2** In the right pane, click **View Details** to view the SQL statement used to create the trigger.

----End

10.3.5 Sequence Management

10.3.5.1 Viewing Sequence Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Sequences**.
- Step 2** In the right pane, click **View Details**.
- End

10.4 Import and Export

10.4.1 Data Import

Scenarios

Import data into a table for backup or migration. If you import a CSV or SQL file, the file must have the same data type as the target table.

Constraints

- Only one file that is no larger than 1 GB can be imported at a time.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.

Procedure

- Step 1** On the top menu bar, choose **Import and Export > Import**.
- Step 2** Import a file from your local PC or an OBS bucket.

- From your local PC

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

For security purposes, imported files are stored in OBS buckets.

NOTE

- To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
- If you select **Delete the uploaded file upon an import success**, the file you upload will be automatically deleted from the OBS bucket after being imported to the destination database.
- From an OBS bucket

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

 **NOTE**

The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 4 View the import progress in the task list or check task details.

----End

10.4.2 Data Export

Scenarios

DAS allows you to export a large amount of data at a time to facilitate data query or to back up data for migration. This section describes how to export data.

Procedure

Step 1 On the top menu bar, choose **Import and Export > Export**.

Step 2 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Step 3 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

 **NOTE**

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- Databases are classified into user databases and system databases. System database cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.
- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 4 After settings are complete, click **OK**.

Step 5 In the task list, view the task ID, type, status, and progress.

Step 6 Click **Details** in the **Operation** column to view task details.

----End

10.5 Account Management

10.5.1 Role Management

Procedure

- Step 1** On the top menu bar, choose **Account Management > Role Management**.
- Step 2** On the role management page, click **Create Role** in the upper left corner.
- Step 3** On the **Basic Settings** tab page, enter a role name (mandatory) and specify other parameters as required. Ensure that all the settings are correct and click **Save**. In the displayed SQL preview window, confirm the settings again and click **OK** to create the role.
- Step 4** (Optional) On the **Role Groups** tab page, select the check boxes in columns **Grant** and **With Grant Option** as needed. Then click **Save** and confirm the settings.
- Step 5** (Optional) On the **Permissions** tab page, click **Add**, select the object type and corresponding column settings from the drop-down lists, and then click **Save** and confirm the settings.
- Step 6** On the role management page, edit existing roles if needed. For how to edit a role, see [3](#) to [5](#). Delete existing roles if needed. Deleted roles cannot be restored. Exercise caution when performing this operation.

----End

11 DDS DB Instance Management (Development Tool)

11.1 Homepage

11.1.1 Overview

Logging In to a DB Instance

Step 1 Obtain the login permission and log in to the DAS console.

Step 2 In the login list of Development Tool, choose **DDS** from the drop-down list, locate the DB instance you want to log in to, and click **Log In** in the **Operation** column.

----End

Function Overview

Table 11-1 Functions

Function	Description
Top menu bar	Provides entries to functional modules such as command operation, database management, and account management.
Account information and logout	Displays the current account information and a logout button.
DB instance information	Displays the current database information, such as the DB engine logo, DB instance name, and DB engine version.

Function	Description
Database list	Displays databases of the current DB instance, including user databases and system databases. <ul style="list-style-type: none">• Allows you to search for a database by name.• Allows you to create databases.

11.1.2 Creating a Database

Procedure

Step 1 In the database list of the **Home** page, click **Create Database**.

Step 2 On the displayed page, enter the database name and collection name (optional).

Step 3 Click **OK**. The database you create appears in the database list.

----End

11.1.3 Dropping a Database

Procedure

Step 1 In the database list, locate the database you want to delete and click **Drop** in the **Operation** column.

Step 2 In the displayed dialog box, click **Yes**.

----End

11.2 Command Operation

DAS manages database collections and views through SQL statements, facilitating user operations.

11.2.1 Command Window

Procedure

Step 1 On the top menu bar, choose **Command Operation > Command Window**.

Step 2 In the displayed text box, enter the commands you want to execute and click **Execute Command**.

Step 3 In the lower part of the page, view the execution results.

- On the **Executed Commands** tab page, view the details about the commands that have been executed.
- On the **Messages** tab page, view the execution results (successful or failed) and execution duration of current commands.

 NOTE

- Currently, the **db.collection.aggregate()** command is not supported. You can run the **db.runCommand** command instead. The following is an example about how to use syntax:

```
db.runCommand({
  aggregate: "collection",
  pipeline: [{
    "$match": {
      "time": {
        "$gte": 20.0
      }
    }
  ],
  {
    "$sort": {
      "time": -1.0
    }
  }
]
});
```

- You can execute SQL commands in either of the following ways:
 - **Append Mode:** Each time a command is executed, the new result set is appended to the previous one.
 - **Overwrite Mode:** Each time a command is executed, the new result set overwrites the previous one.

----End

11.2.2 Command History

Prerequisites

- You have run SQL commands in the command window.
- You have enabled **Save Executed Commands** in the command window.

Procedure

Step 1 On the top menu bar, choose **Command Operation > Command History**.

Step 2 In the command execution list, view the execution records.

- You can search execution records by time range, database name, or command keyword.
- **Database name:** Click a database name to go to the database management page.
- **SQL Statement:** Click a statement to view or copy it.
- **Open in command window:** Click it to view the executed commands in a window for easy view and use.

----End

11.3 Database Management

Function Overview

Table 11-2 Functions

Function	Description
Sidebar	Offers two types of objects, collections and views.
Database information	Displays the name, IP address, and command window of the current database.
Object list	Collections: list the validation rule, including the validation level and action. Views: allow different validations by setting pipeline phases and collations.

 NOTE

You can go to the **Database Management** page in either of the following ways:

- Choose **Database Management** on the top menu bar.
- In the database list of the **Home** page, locate the target database and click **Manage** in the **Operation** column.

11.3.1 Collection Management

11.3.1.1 Creating a Collection

Prerequisites

You have logged in to a DDS B instance and created a database.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the **Collections** tab page, click **New Collection**.
- Step 3** On the displayed page, enter the collection name. Select **Capped collection**, **Auto index**, and **No padding** as required.
 - **Capped collection:** Set the maximum number of bytes and the maximum number of documents. If the specified value is reached, an error is reported. The value specified must be greater than 0.
 - **Auto index:** Automatically create an index on the **_id** field.
 - **No padding:** A storage policy that does not reserve storage space for the growth of a record when inserting it.

Step 4 Enter the validation rule, and set **Validation Level** and **Validation Action**.

 **NOTE**

- Options for **Validation Level** include:
 - **strict**: a strict validation level.
 - **moderate**: a moderate validation level.
 - **off**: no validation level.
- Options for **Validation Action** include:
 - **error**
 - **warn**

Step 5 Ensure that all the settings are correct and click **OK**. The collection you create appears in the collection list, and you can perform operations on it.

----End

11.3.1.2 Opening a Collection

Prerequisites

You have logged in to a DDS DB instance, created a database, and created a collection in the database.

Procedure

Step 1 On the top menu bar, choose **Database Management**.

Step 2 On the displayed **Objects** tab page, choose **Collections**, locate the target collection, and click **Open Collection** in the **Operation** column.

Step 3 Set parameters **Query**, **Projection**, **Sort**, **Limit**, and **Skip** to query the documents in the current collection.

On the **JSON** tab page, view the filtered documents. To facilitate viewing, DAS allows you to view the documents in Table and Tree mode.

 **NOTE**

- **Query**: uses the query operator to specify the filter.
- **Projection**: specifies the fields that matched with the query filter in the document to return.
- **Sort**: defines the sorting order.
- **Limit**: specifies the maximum number of documents that can be returned.
- **Skip**: specifies the number of skipped documents.

Step 4 Click **Create Document**. On the displayed page, enter a valid JSON file and click **Save**.

----End

11.3.1.3 Altering Validator

Prerequisites

You have logged in to a DDS DB instance, created a database, and created a collection in the database.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
 - Step 2** On the displayed **Objects** tab, choose **Collections**, locate the target collection, and click **Alter Validator** in the **Operation** column.
 - Step 3** On the displayed **Alter Validator** page, confirm the validation rule, and set the validation level and action.
 - Step 4** Click **OK**.
- End

11.3.1.4 Renaming a Collection

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
 - Step 2** On the displayed **Objects** tab, choose **Collections**, locate the target collection, and click **Rename** in the **Operation** column.
 - Step 3** On the displayed page, set a new name for the collection and click **OK**.
- End

11.3.1.5 Clearing a Collection

NOTICE

Cleared collections cannot be recovered. Exercise caution when performing this operation.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
 - Step 2** On the displayed **Objects** tab, choose **Collections**, locate the target collection, and click **Clear** in the **Operation** column.
 - Step 3** In the displayed dialog box, click **Yes**.
- End

11.3.1.6 Dropping a Collection

NOTICE

Dropped collections cannot be recovered. Exercise caution when performing this operation.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Collections**, locate the target collection, and click **Drop** in the **Operation** column.
- Step 3** In the displayed dialog box, click **Yes**.

----End

11.3.1.7 Creating an Index

Prerequisites

You have logged in to a DDS DB instance and created a database and a collection in the database.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** Choose **Collections** in the navigation pane. Under the collection list, click **New Index**.
- Step 3** On the displayed page, enter an index name, add an index key, and click **Save**.

NOTE

If you do not select **Background** for **Advanced Settings**, the following dialog box is displayed.


----End

11.3.1.8 Editing an Index

Prerequisites

You have logged in to a DDS DB instance and created a database, a collection, and an index.

Procedure

- Step 1** Choose **Collections** in the navigation pane. In the index list, locate the collection whose index you want to edit, and click  before the index name.

Step 2 Locate the index that you want to edit and click **Edit** in the **Operation** column.

Step 3 Click **Save**.

----End

11.3.1.9 Deleting an Index

Prerequisites

You have logged in to a DDS DB instance and created a database, a collection, and an index.

Procedure

Step 1 Choose **Collections** in the navigation pane. In the index list, locate the collection whose index you want to edit, and click **+** before the index name.

Step 2 Locate the index that you want to delete and click **Delete** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

11.3.2 View Management

11.3.2.1 Creating a View

Prerequisites

You have logged in to a DDS DB instance and created a database.

Procedure

Step 1 On the top menu bar, choose **Database Management**.

Step 2 On the displayed **Objects** tab, choose **Views**, and click **Create View**.

Step 3 On the displayed page, enter the view name and collection name.

Step 4 On the **Aggregation Pipeline** tab, click **Add Pipeline Stage**. On the displayed page, select an operator, edit the expression, and click **Save**.

The aggregation pipeline you create appears in the list, and you can edit, delete, move up, or move down it.

Step 5 On the **Collation** tab, click **Custom Collation** and set collation rules as needed.

Step 6 After setting the view information, click **Save**.

----End

11.3.2.2 Opening a View

Prerequisites

You have logged in to a DDS DB instance, created a database, and created a view in the database.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Views**, locate the target view, and click **Open** in the **Operation** column.
- Step 3** Set parameters **Query**, **Projection**, **Sort**, **Limit**, and **Skip** to query the documents in the current view.

On the **JSON** tab page, view the filtered documents.

NOTE

To facilitate viewing, DAS allows you to view documents in Table and Tree mode.

----End

11.3.2.3 Editing a View

Prerequisites

You have logged in to a DDS DB instance, created a database, and created a view in the database.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Views**, locate the target view, and click **Edit** in the **Operation** column.
- Step 3** On the displayed page, change the collection name.
- Step 4** Edit the aggregation pipeline and collation rule if need. For details, see [Creating a View](#).
- Step 5** After editing the view information, click **Save**.

----End

11.3.2.4 Dropping a View

Prerequisites

You have logged in to a DDS DB instance, created a database, and created a view in the database.

NOTICE

Dropped views cannot be recovered. Exercise caution when performing this operation.

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Views**, locate the target view, and click **Drop** in the **Operation** column.
- Step 3** In the displayed dialog box, click **Yes**.

----End

11.4 Account Management

11.4.1 User Management

Scenarios

You have logged in to a DDS DB instance.

Creating a User

- Step 1** On the top menu bar, choose **Account Management > User Management**.
- Step 2** Select the database for which a user is to be created and click **Create User**.
- Step 3** On the displayed page, enter the username and password, and confirm the password.
- Step 4** In the lower part of the page, click **Add Role**. On the displayed page, select roles for the database.

 **NOTE**

If you assign a role from another database to this created user, the user has the operation permissions on that database.

- Step 5** After setting required parameters, click **Save**. In the preview dialog box, click **OK**.

----End

Editing a User

The procedure for editing a user is the same as that for creating a user. For details, see [Creating a User](#).

Deleting a User

Step 1 On the top menu bar, choose **Account Management > User Management**.

Step 2 In the user list, locate the user you want to delete and click **Delete**.

Step 3 In the displayed dialog box, click **Yes**.

----End

11.4.2 Role Management

Scenarios

You have logged in to a DDS DB instance.

Procedure

Step 1 On the top menu bar, choose **Account Management > Role Management**.

Step 2 Select the database for which the roles are to be created and click **Add Roles**.

Step 3 Enter the role name. On the **Permission** tab, click **Add Privileges**. On the displayed page, specify required parameters and click **Add**.

Step 4 On the **Role** tab page, click **Add Roles**. On the displayed page, select the roles you want to add and click **Add**.

Step 5 After setting the roles, click **Save**.

NOTE

- If a role is created for the admin database, you can select other databases when adding privileges or roles, to grant the permissions of other databases to this role. Users with this role have the permissions to operate the selected database.

----End

12 Distributed DDM DB Instance Management (Development Tool)

12.1 Homepage

12.1.1 Overview

Logging In to a DB Instance

Step 1 Obtain the login permission and log in to the DAS console.

Step 2 In the login list of Development Tool, locate the DB instance you want to log in to and click **Log In** in the **Operation** column.

----End

Function Overview

Table 12-1 Functions

Function	Description
Top menu bar	<ul style="list-style-type: none">• SQL Operations• Database Management• Account Management
Account information and logout	Displays the current account information and a logout button.
DB instance information	Displays the current database information, such as the DB engine logo, DB instance name, and DB engine version.

Function	Description
Database list	Displays the databases of the current DB instance, including user databases and system databases. Allows you to search for a database by name. Allows you to create databases.

12.1.2 Creating and Deleting a Database

Database operations are currently not supported. To create or delete a database, see "DDM Schema Management" in the *Distributed Database Middleware User Guide*.

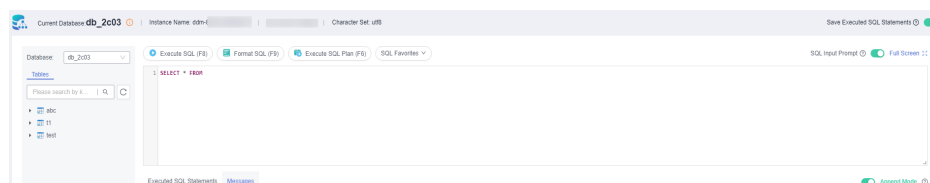
12.2 SQL Operations

12.2.1 SQL Window

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL Window**.
- Step 2** In the navigation pane, select a database, and then specify a table or view.

Figure 12-1 SQL operations



- **Execute SQL:** executes SQL statements.
- **Format SQL:** improves the readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics.
The formatting takes effect for all the SQL statements in the SQL window. You cannot format only the selected statements.
- **Execute SQL Plan:** reports the execution of SQL statements to facilitate troubleshooting and optimize SQL processing performance.
- **SQL Favorites:** allows you to add, view, and manage frequently used SQL statements.
- **SQL Input Prompt:** helps you to select your desired database, table, or field name as prompted to quickly enter statements in the SQL window.
- **Overwrite/Append Mode:**
 - **Append Mode:** Each time a SQL statement is executed, the new result set is appended to the previous one.

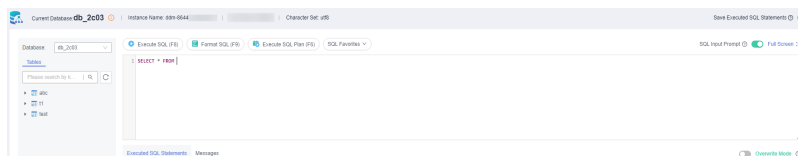
- **Overwrite Mode:** Each time a SQL statement is executed, the new result set overwrites the previous one.
 - **Full Screen:** shows SQL statements on a full screen.
- Step 3** After SQL statements are executed, view SQL execution status on the **Executed SQL Statements**, **Messages**, and **Result Set** tab pages.
- End

12.2.2 Executing SQL Statements

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL Window**. In the SQL window, select a database from the navigation pane.
- Step 2** Enter SQL statements in the SQL window and click **Execute SQL**.
- You are advised to enable **SQL Input Prompt**. It helps you to select your desired database, table, or field name as prompted to quickly enter statements in the SQL window.
 - To execute multiple SQL statements at a time, separate them using semicolons (;). In this case, click **Full Screen** to view the logics in SQL statements clearly.
 - To execute some of the SQL statements, select the statements before executing them.
 - To execute all the SQL statements, do not select any SQL statements or select all of them.

Figure 12-2 Executing SQL statements



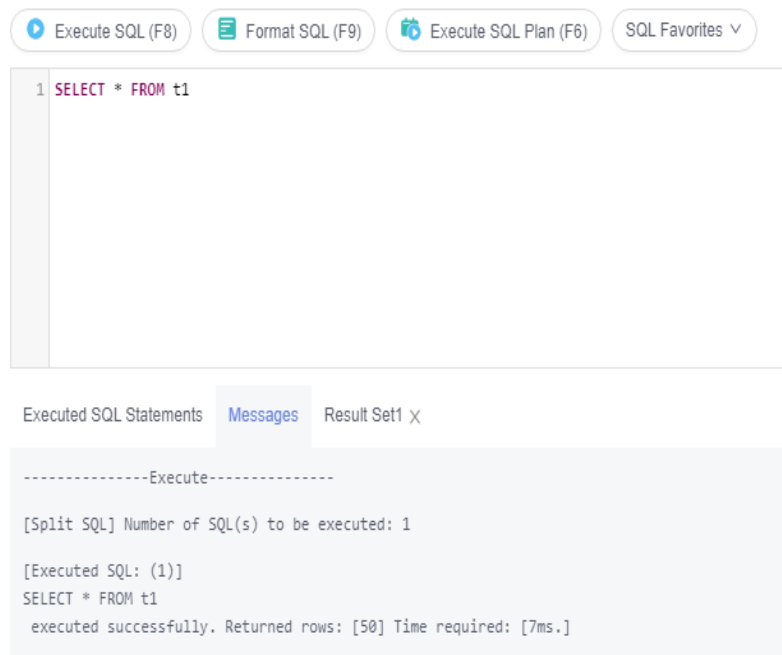
NOTE

```
"SELECT * FROM t1;"
```

student indicates the table name.

- Step 3** View the executed SQL statements and the current execution status in the lower part of the page. On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.

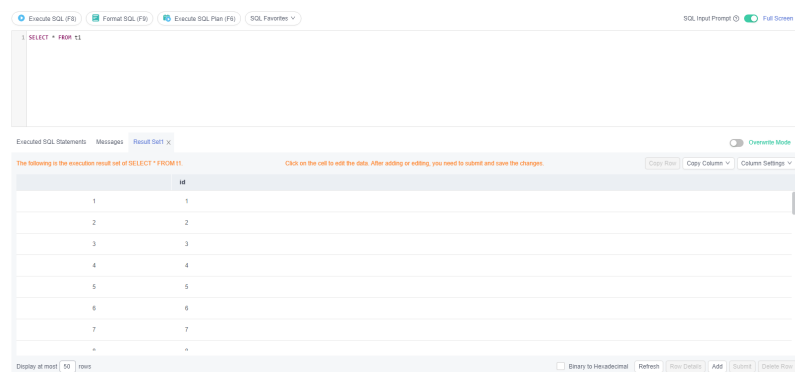
Figure 12-3 Viewing messages



Step 4 On the **Result Sets** tab, view the execution details. You can perform the following operations on the result sets:

- **Copy Row/Copy Column:** copies a row or column for reuse.
- **Column Settings:** customizes the columns to display when there are a large number of columns in the query result.
- **Refresh:** refreshes the changed data.
- **Row Details:** displays the column field name, value, and type of the selected row.
- **Add:** adds an empty row to the result set.
- **Submit:** views the SQL statements to be modified. After you click **OK**, the result set is updated to the latest.
- **Delete Row:** deletes the selected row, including the data.
- **Export:** allows you to export the data in a SQL or CSV file. A maximum of 10,000 rows of data can be exported.

Figure 12-4 Viewing result sets



NOTE

- If the type of the result set is metadata, the data cannot be edited or displayed on multiple pages.
- If the result set involves multiple tables, you cannot edit the data.
- If the result set involves only one table and does not include all the primary key columns of the table or no primary key is found, you cannot edit the data.
- Virtual tables (for example, tables generated during SQL execution) cannot be edited.

----End

12.2.3 Executing SQL Plan

Scenarios

This section describes how to analyze the execution efficiency of the current SQL statement.

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL Window**. On the displayed page, select a database from the navigation pane.
- Step 2** Enter SQL statements in the SQL window and click **Execute SQL Plan**.

Figure 12-5 Execute SQL Plan



- On the **Executed SQL Statements** tab page, view the historical SQL executions.
- On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.
- On the **Execution Plan** tab page, view the execution status for each SQL statement.

----End

12.2.4 SQL Favorites

Scenarios

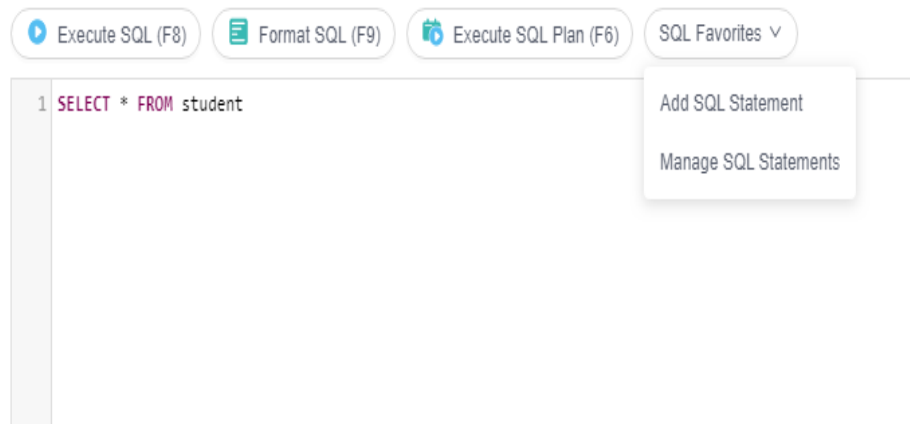
DAS allows you to add frequently used SQL statements and view and manage SQL statements.

Procedure

Step 1 On the top menu bar, choose **SQL Operations > SQL Window**.

Step 2 In the SQL window, click **SQL Favorites**.

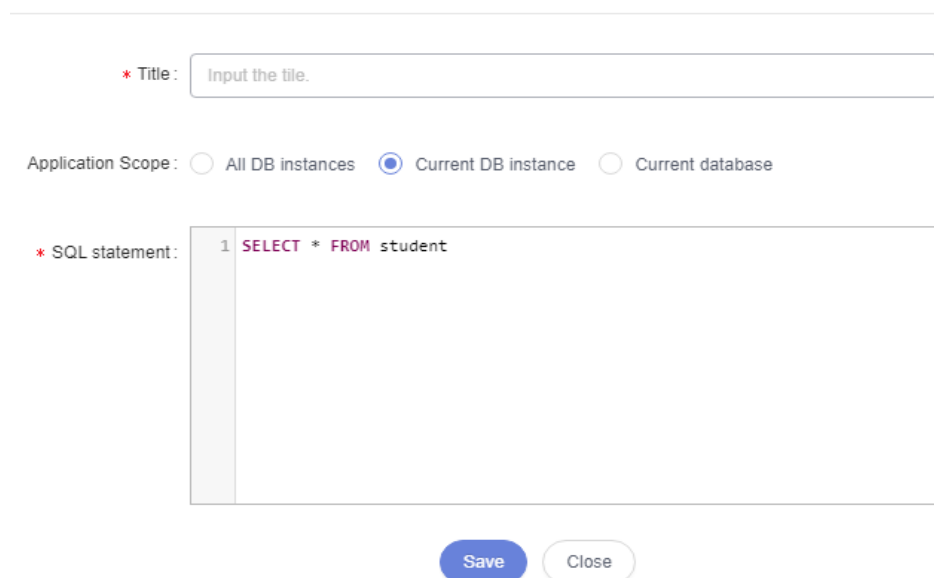
Figure 12-6 SQL Favorites



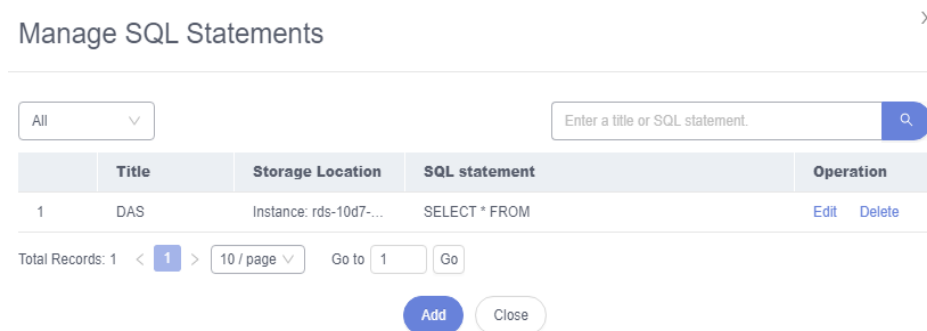
- **Add SQL Statement:** allows you to add your desired SQL statements. Specifically, specify the title and enter SQL statements as required, and then click **Save**.

Figure 12-7 Add SQL Statement

Add SQL Statement

The image shows a dialog box titled 'Add SQL Statement'. It has a text input field for the title with the placeholder text 'Input the title.'. Below the title field are three radio buttons for 'Application Scope': 'All DB instances', 'Current DB instance' (which is selected), and 'Current database'. Below the radio buttons is a text area for the SQL statement with the text '1 SELECT * FROM student'. At the bottom of the dialog are two buttons: 'Save' and 'Close'.

- **Manage SQL Statements:** allows you to manage frequently used SQL statements, including adding, editing, and deleting statements.

Figure 12-8 Manage SQL Statements

----End

12.2.5 SQL History

Prerequisites

You have logged in to a DB instance on the DAS console.

Procedure

- Step 1** On the top menu bar, choose **SQL Operations > SQL History**.
- Step 2** On the displayed page, search for execution information about the target SQL statement by time range, database name, or keyword.
- To access the **Database Management** page, click a database name.
 - To copy and use a SQL statement, click it in the **SQL Statement** column.
 - To directly execute a SQL statement, click **Open in SQL Window** in the **Operation** column.

NOTE

After you enable **Save Executed SQL Statements**, executed SQL statements will be recorded in the DAS management host.

----End

12.3 Database Management

Function Overview

Table 12-2 Functions

Function	Description
Database information	Displays the name, IP address, character set, and SQL window of the current database.

Function	Description
Object list	Displays tables only.
Metadata collection	Allows DAS to automatically collect only the data such as database names, table names, and field names in an instance (user data in your tables is not included).
List details	Displays the operation area for each type of object.

 **NOTE**

You can go to the **Database Management** page in either of the following ways:

- Choose **Database Management** on the top menu bar.
- In the database list of the **Home** page, locate the target database and click **Manage** in the **Operation** column.

Object List

You can manage distributed DDM instance tables and perform operations including creating, querying, and modifying objects.

For details on how to manage tables, see [Table Management](#).

Metadata Collection

DAS periodically collects the metadata of DB instance databases, tables, and fields and stores the collected data in the DAS database. In addition, it collects only structural metadata, but not data in user tables, which ensures data security. Metadata collection delivers the following advantages:

- Reduces queries on user databases and the impact on the performance of user databases.
- Improves the search performance for DB instances that contain a large number of tables and supports pagination queries.

 **NOTE**

- If there are too many tables, the system will neither collect metadata nor display tables to minimize impacts on database performance.
- To ensure stable performance, DAS will temporarily stop collecting data when the instance QPS is over 10,000 or TPS over 1,000.

Procedure

- Step 1** On the **Database Management** page, enable **Auto Metadata Collection** in the upper right corner and click the **Metadata Collection** tab.

Step 2 On the displayed page, click **Collect Now** to start the collection. You can also stop the collection or view collection details.

- **Clear Collected Data:** Clears the collected data such as metadata, database structure, and table structure.
- **Delete Logs:** Deletes logs. Deleted logs cannot be restored. Exercise caution when performing this operation.

----End

12.3.1 Table Management

12.3.1.1 Creating a Table

Scenarios

A data table consists of basic information, fields, indexes, foreign keys, and CHECK constraints, among which indexes, foreign keys, and CHECK constraints are optional. Configure these items as required.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** and click **Create Table**.

Step 2 On the displayed **Basic Information** page, specify the table name, storage engine, and character set. Then, click **Next**.

Step 3 On the **Column** page, click **Add** and set **Column Name, Type, Nullable, Primary Key**, and **Extended Information** as needed.

- If you do not need to add indexes, foreign keys, or CHECK constraints, click **Create**.
- If you need to add indexes, foreign keys, or CHECK constraints, click **Next** until all your desired parameters are specified. After the settings are complete, click **Create**.

Step 4 In the **SQL Preview** dialog box, click **Execute** to create a table.

----End

12.3.1.2 Opening a Table

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** on the left, locate the target table, and click **Open** in the **Operation** column.

Step 2 View table information on the table details page. Table editing is currently not supported.

Table 12-3 Functions

Item	Description
Where Condition	Filters records.
Copy Row	Copies data of the selected row. Specifically, double-click the target row and then click Copy Row .
Delete Row	Deletes data of the selected row. Specifically, double-click the target row and then click Delete Row .
Copy Column	Allows you to copy all data in a column by selecting this column from the Copy Column drop-down list.
Column Settings	Allows you to set the columns you want to display.
Binary to Hexadecimal	Prevents binary data in columns from being displayed as garbled characters. If there is binary data in columns, you must select this option.
Add	Allows you to add data to a table.
Refresh	Allows you to update table data manually.
Submit	Allows you to edit data for the second time.
Row Details	Shows the detailed information of a specified row.

----End

12.3.1.3 Viewing Table Details

Procedure

- Step 1** On the top menu bar, choose **Database Management**.
- Step 2** On the displayed **Objects** tab, choose **Tables** on the left, locate the target table, and click **View** in the **Operation** column.
- Step 3** In the displayed dialog box, view basic table information and SQL statements for creating the database.

----End

12.3.1.4 Altering a Table

Procedure

- Step 1** On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** on the left.
- Step 2** In the right pane, locate the target table and click **Alter** in the **Operation** column.
- Step 3** Modify the table information as required. After the modification is complete, click **Alter**.

Step 4 In the **SQL Preview** dialog box, click **Execute**.

----End

12.3.1.5 Clearing a Table

NOTICE

Cleared tables cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** on the left.

Step 2 In the right pane, locate the target table, and choose **More > Clear** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

12.3.1.6 Dropping a Table

NOTICE

Dropped tables cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 On the top menu bar, choose **Database Management**. On the displayed **Objects** tab, choose **Tables** on the left.

Step 2 On the displayed **Objects** page, select **Tables**, locate the target table, and choose **More > Drop** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

----End

12.4 Account Management

12.4.1 User Management

Scenarios

Multiple users with different permissions can be created to access a DB instance or database, but the permissions of these users cannot exceed the operation permissions of the account.

Creating a User

- Step 1** On the top menu bar, choose **Account Management > User Management**.
- Step 2** Click **Create User**. On the displayed page, enter the user information, such as basic information, basic permissions, extended permissions, and schema (optional), and select the read policy type.
- Step 3** Ensure that the settings are correct and click **Save**.

Table 12-4 Basic permissions

Permission	Description
CREATE	Permission to create databases for a specified object.
DROP	Permission to delete databases for a specified object.
ALTER	Permission to modify databases for a specified object.
INDEX	Permission to query indexes of a specified object.
INSERT	Permission to insert data into a specified object.
DELETE	Permission to delete data from a specified object.
UPDATE	Permission to update data of a specified object.
SELECT	Permission to query data of a specified object.

----End

Editing a User

- Step 1** On the top menu bar, choose **Account Management > User Management**.
- Step 2** In the user list, locate the target user and click **Edit** in the **Operation** column.
- Step 3** On the displayed page, edit basic information, basic permissions and extended permissions, and click **Save**.

----End

Deleting a User

- Step 1** On the top menu bar, choose **Account Management > User Management**.

Step 2 In the user list, locate the target user and click **Delete** in the **Operation** column.
In the displayed dialog box, click **Yes**.

----End

13 Intelligent O&M

13.1 Overview

DAS Intelligent O&M provides an all-round intelligent monitoring, diagnosis, and tuning platform for databases, making it easy for you to perform operations.

 **NOTE**

Intelligent O&M is available in only regions CN-Hong Kong, CN East-Shanghai1, AP-Singapore, and LA-Sao Paulo1.

13.2 MySQL DB Instances

13.2.1 Function Overview

Procedure

- Step 1** On the DAS console, click **Intelligent O&M** to go to the instance overview page of Intelligent O&M.
- Step 2** In the **Instance Overview** area, click **Set Displayed Metrics** to specify the performance metrics to display.

Figure 13-1 Set Displayed Metrics

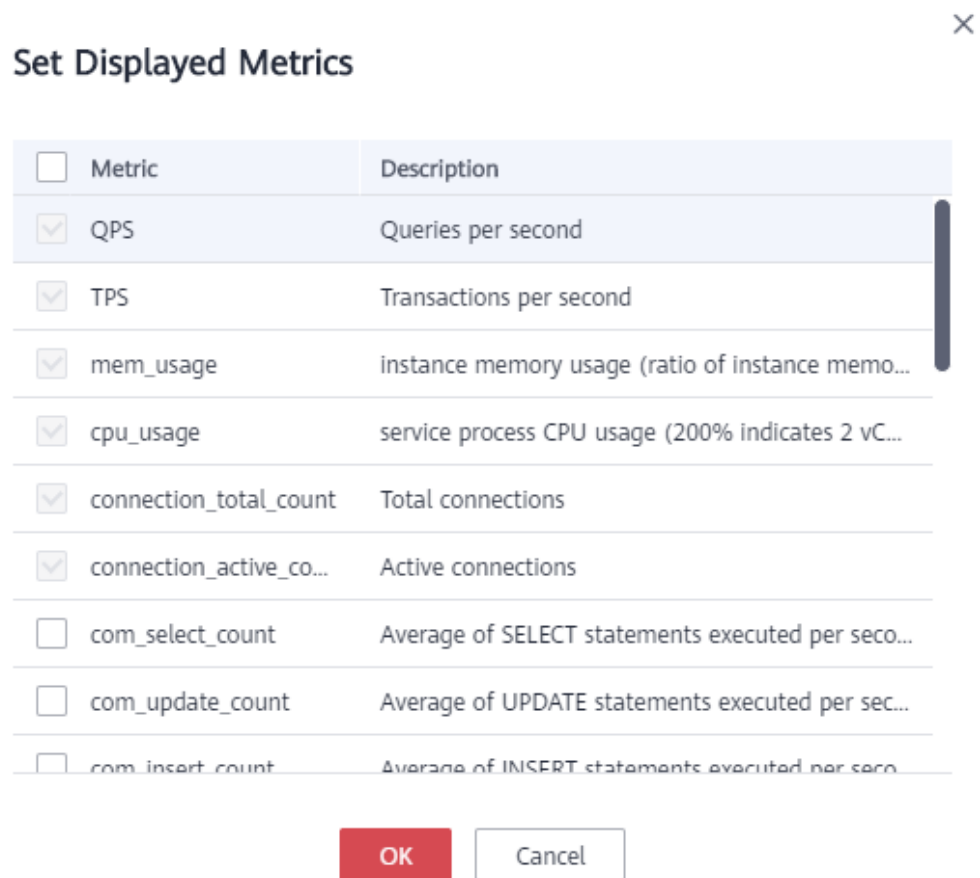


Table 13-1 Performance metrics

Metric	Description
QPS	Queries per second
TPS	Transactions per second
mem_usage	Instance memory usage (ratio of instance memory to OS memory)
cpu_usage	Service process CPU usage (200% indicates two vCPU cores are occupied)
connection_total_count	Total connections
connection_active_count	Active connections
com_select_count	Number of SELECT statement executions per second
com_update_count	Number of UPDATE statement executions per second

Metric	Description
com_insert_count	Number of INSERT statement executions per second
com_delete_count	Number of DELETE statement executions per second
com_replace_count	Number of REPLACE statement executions per second
innodb_rows_deleted	Number of rows deleted from an InnoDB table per second
innodb_rows_read	Number of rows read from an InnoDB table per second
innodb_rows_inserted	Number of rows inserted into an InnoDB table per second
innodb_rows_updated	Number of rows updated in an InnoDB table per second
bytes_received	Number of bytes received from all clients per second
bytes_sent	Number of bytes sent to all clients per second
innodb_data_read	Number of read bytes per second in an InnoDB buffer
innodb_data_written	Number of write bytes per second in an InnoDB buffer
innodb_buffer_pool_read_requests	Number of times that InnoDB reads pages (logical read) from a buffer pool per second
innodb_buffer_pool_write_requests	Number of times that InnoDB writes data to a buffer pool per second
innodb_buf_hit	InnoDB read cache hit ratio
innodb_buf_dirty	Dirty page ratio of the InnoDB buffer pool
innodb_buf_usage	Usage of the InnoDB buffer pool
created_tmp_disk_tables	Number of temporary tables automatically created on hard disks when SQL statements are executed
avg_disk_ms_per_read	Average time required for each disk read
avg_disk_ms_per_write	Average time required for each disk write
disk_read_throughput	Number of disk read bytes per second
disk_write_throughput	Number of disk write bytes per second
disk_size	<ul style="list-style-type: none"> disk_total_size: Total disk space disk_used_size: Used disk space

Metric	Description
innodb_log_writes_requests	Number of writes to the redo log file per second
innodb_log_writes	Number of physical writes to the redo log file per second
innodb_os_log_fsyncs	Number of fsync() writes on InnoDB log files per second
IOPS	Number of I/O requests processed per second

Step 3 After specifying the metrics, click **OK**.

Step 4 Click **Details** of an instance to go to the Intelligent O&M details page.

----End

13.2.2 Performance

13.2.2.1 Real-Time Diagnosis

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

Step 3 Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.

Step 4 Choose **Performance > Real-Time Diagnosis**.

Step 5 On the **Real-Time Diagnosis** tab page, view the current resource usages, real-time performance, and slow query logs of the instance.

----End

13.2.2.2 Real-Time Performance

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

Step 3 Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.

Step 4 On the **Real-Time Performance** tab page, view the server uptime, connection information, and buffer pool.

Step 5 Click **Start/Pause** or change the time interval (1 to 10 seconds) to adjust the performance monitoring. You can also click **Set Metric** to specify required metrics based on service requirements.

Table 13-2 Real-time performance metrics

Metric	Description
Slow Queries	Difference between two adjacent slow query collection points.
QPS TPS	QPS: Number of SQL statement executions per second. TPS: Transactions executed per second
Connections	Total: Number of enabled connections. Active: Active connections among the enabled ones.
DML Executions	select: SELECT statement executions per second. insert: INSERT statement executions per second. update: UPDATE statement executions per second. delete: DELETE statement executions per second. insert_select: INSERT and QUERY statement executions per second.
InnoDB Cache	Cache Hit Ratio: Cache hit ratio. Cache Usage: Cache usage.
InnoDB Accessed Rows	rows_read: Rows read from an InnoDB storage engine table. rows_inserted: Rows inserted into an InnoDB storage engine table. rows_updated: Rows updated in an InnoDB storage engine table. rows_deleted: Rows deleted from an InnoDB storage engine table.
Logical Reads Physical Reads	Logic Reads: Logical reads per second. Physical Reads: Physical reads per second.
InnoDB Average Row Lock Duration (ms)	Average lock time of InnoDB row locks. The unit is ms.
InnoDB Row Lock Waits	Average InnoDB Row Lock Waits: Average wait times of InnoDB row locks. Current InnoDB Row Lock Waits: Current waits of InnoDB row locks.
Temporary Tables Files	Temporary tables: Temporary tables that are automatically created during SQL statement execution. Temporary files: Temporary files that are automatically created during SQL statement execution.

Metric	Description
Network Traffic	Total: Total incoming and outgoing traffic of an instance. Incoming: Incoming traffic of an instance. Outgoing: Outgoing traffic of an instance.

----End

13.2.2.3 Performance History

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.
- Step 4** Choose **Performance > Performance History** to view performance graphs of instance metrics.
- Step 5** Click **Select Metric** to specify your desired metrics and click **OK**.
Set a time range in the upper right corner so that you can view the DB instance performance during this period. You can also click **Detail** on each metric performance graph to set the time range.
- Step 6** Enable **Interactive Graph** to view the performance of other metrics at the same time.

----End

13.2.2.4 Performance Trends Comparisons

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.
- Step 4** Choose **Performance > Performance Trends Comparisons**.
- Step 5** Click **Select Metric** to specify your desired metrics, and click **OK**.
- Step 6** Set the date and time period so that you can view the performance data during this period. Alternatively, click **Detail** on each metric performance graph to set the date and time range.

----End

13.2.2.5 Custom Graphs

Procedure

- Step 1** Log in to the DAS console.
 - Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
 - Step 3** Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.
 - Step 4** On the **Custom Graphs** tab page, customize the metrics you want to view.
 - Step 5** Click **Create Graph**, enter a graph name, select the required metrics, and click **OK** to view performance history in a graph.
- End

13.2.3 Sessions

13.2.3.1 Real-Time Sessions

Procedure

- Step 1** Log in to the DAS console.
 - Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
 - Step 3** Select the target instance and click **Details**.
 - Step 4** Choose **Sessions > Real-Time Sessions** to view the real-time session statistics collected by user, host, and database.
 - Step 5** On the **All Sessions** page, select the abnormal session you want to end and click **Kill** for the databases to recover.
 - Step 6** On the **Slow Sessions** tab page, specify **Slow Session Threshold** and click the refreshing button or enable **Auto Refresh** to search for the SQL statements whose execution time exceeds the threshold. Select the sessions you want to end and click **Kill** for the databases to recover.
- End

13.2.3.2 Emergency Channel

Scenarios

- Emergency Channel
 - If the maximum number of sessions for an instance has been reached and the instance cannot be logged in to, you can view and kill unnecessary sessions through the emergency channel.
 - Emergency channel is available to only RDS MySQL databases currently, but not to self-built databases on ECSs and instances in the creating, frozen, or abnormal state.

- Use this function in urgent conditions. All your kill operations will be logged.
- If your instance can be logged in on the DAS console, log in to the instance and perform desired operations through **Real-Time Sessions**.
- Sessions of sensitive users, like rdsadmin, rdsbackup, rdsmetric, and rdsRepl cannot be killed.
- History Logs
On the **History Logs** tab page, you can view all your sessions killed through the emergency channel.

Prerequisites

The DB engine must be MySQL. The MySQL 5.6 version must be 5.6.43.3 or later, and MySQL 5.7 version must be 5.7.25.3 or later.

Procedure

- Step 1** Log in to the DAS console.
 - Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
 - Step 3** Select the target instance and click **Details**.
 - Step 4** On the **Emergency Channel** tab page, select the sessions you want to kill. By default, the sessions are sorted in descending order by session duration.
 - Step 5** Click **Kill**. In the displayed dialog box, click **Yes**.
 - Step 6** On the **History Logs** tab page, view your sessions killed through the emergency channel.
- End

13.2.4 SQL

13.2.4.1 Slow Query Logs

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Select the target instance and click **Details**.
- Step 4** Choose **SQL > Slow Query Logs** to view log trends of the instance.
- Step 5** Set a time range or select **Last 1 hour**, **Last 6 hours**, or **Last 1 day** to view slow query logs, CPU usage, and archived slow query logs.
- Step 6** Toggle on **Enable DAS Slow Query Log**. In the displayed dialog box, read and agree to the security agreement, and click **OK**.
- Step 7** At the lower part of the **Log Trends** page, view statistics and details of slow query logs.

- Query and view slow query logs by database name.

Step 8 On the **Log Archiving** tab page, obtain the slow query log files of the current DB instance in either of the following ways: (The archiving function saves slow query log files generated in the last 10 days.)

- Click **Generate Latest Log File**.
- DAS automatically generates a slow query log every time the log data reaches 50 MB.

Step 9 In the slow query log list, locate the log that you want to view and click **Download** to download log data.

----End

13.2.4.2 SQL Explorer

Precautions

- After SQL Explorer is disabled, DAS will stop collecting new SQL statements and delete collected SQL data.
- SQL Explorer cannot record all data. As for MySQL, it has the following constraints:
 - Some data cannot be recorded if a buffer overrun occurs.
 - Any SQL statement that exceeds 4096 bytes is discarded by default.

This constraint can be specified by setting parameter **rds_sql_tracer_reserve_big_records** in MySQL 5.7.33.3 or later. For example, you can set the parameter to **ON** on the RDS instance parameter setting page, indicating that SQL statements whose length exceeds 4096 bytes are still recorded.

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

Step 3 Select the required instance and click **Details**.

Step 4 Choose **SQL > SQL Explorer** to view SQL Explorer details of the instance.

Step 5 Toggle on **Enable DAS SQL Explorer** to obtain information about all SQL statements executed on the current instance.

NOTE

- SQL Explorer takes effect once it is enabled.

Step 6 Click the **TOP SQL** tab, select **Last 1 hour**, **Last 3 hours**, or **Last 6 hours** or specify a time range spanning no more than one day, to view the top SQL statements executed.

 **NOTE**

Execution Duration Distribution displays the distribution of the time required to execute all SQL statements in a specific time period. There are four execution duration ranges. The system calculates execution times of the SQL statements in each execution time range. If you select **Last 1 hour**, the execution durations are calculated every 10 seconds. If you select **Last 6 hours**, the execution durations are calculated every 1 minute. If you select a time range longer than 6 hours, the execution durations are calculated every 5 minutes. The execution duration ranges of SQL templates are as follows:

- < 100 ms
- 100 ms—500 ms
- 500 ms—1s
- > 1s

Step 7 Select **By instance** or **By node**. If you select **By node**, the execution duration distribution of the primary node is displayed by default. To view the duration distribution of other nodes, click **Change Node**.

Step 8 Use either of the following method to view execution duration details, such as average execution duration, total duration, average lock wait duration, and average scanned rows.

- Hover your mouse at any point in time on the graph to view top SQL templates at that time.
- Specify a time range using your mouse on the graph, and you will see top SQL templates used during that time range.

Step 9 In the SQL template list, locate a SQL template and click **Details** to view total executions, average rows scanned, average execution duration, and the like.

Step 10 In the SQL template list, locate a SQL template and click **Diagnose** to view details such as the execution plan, suggestions for tuning indexes and SQL statements.

Step 11 On the **SQL Audit** page, specify data collection time, task creation time, and node information to search for SQL audit tasks.

Step 12 Click **Add SQL Audit Task** and specify a time range to add a task. You can add instance- or node-level task based on your service requirements.

Step 13 Locate the added task in the lower part.

Step 14 Click **View Detail** in the **Operation** column to view task details.

Step 15 Specify conditions such as **Time Range**, **User**, **Keyword**, **Database** and click **Query**. The selected time range must be after the time when the SQL audit task is added.

----End

13.2.4.3 SQL Diagnosis

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

- Step 3** Select the target instance and click **Details**.
- Step 4** Choose **SQL > SQL Diagnosis**. On the displayed page, select your desired database, enter the SQL statements to be diagnosed, and click **Diagnose**. The diagnosis result page is displayed.
- Step 5** On the **SQL Diagnosis** page, enter SQL statements and click **Execute**. The system automatically executes the statements and displays a result set.
- Step 6** On the **SQL Diagnosis** page, enter multiple SQL statements and click **Format**. The system automatically formats all SQL statements.
- Step 7** On the **SQL Diagnosis** page, enter multiple SQL statements and click **View Execution Plan**. The system automatically executes all SQL statements in sequence.

 **NOTE**

- Only MySQL InnoDB is supported.
- Only the diagnosis of the SELECT, INSERT, UPDATE, and DELETE statements is supported. An INSERT statement must contain a SELECT clause.
- SQL statements for querying system databases like **information_schema**, **test**, and **mysql** are not supported.
- View statements are not supported.
- The SQL tuning function obtains the table structure and data distribution information (non-original). The obtained data is only for logic diagnosis, but not stored on the DAS server.
- Obtaining table structure and data distribution information may add additional loads to the DB instance, but has little impact on its performance.
- Only the SQL diagnosis history is stored on the DAS server. You can delete it from the server permanently.
- **Format SQL**: improves the readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics.

The formatting takes effect for all the SQL statements in the SQL window. You cannot format only the selected statements.

----End

13.2.4.4 SQL Statement Concurrency Control

Scenarios

SQL statement concurrency control aims to keep MySQL instances running stably even if there is a sudden increase in concurrent SQL statements.

Prerequisites

- The kernel version of primary RDS for MySQL instances must be 5.6.50.3, 5.7.31.4, 8.0.25.1, or later.
- The kernel version of RDS for MySQL read replicas must be 5.6.51.6, 5.7.37.1, 8.0.25.1, or later.
- Separate SQL statement concurrency rules can be added for RDS for MySQL read replicas of the kernel version 5.7.38-221000 or later.

Precautions

- If a SQL statement matches multiple concurrency rules, only the latest rule takes effect.
- Keywords in a concurrency control rule are sorted in a specific order, and the system will match them from first to last. For example, "a" and "b" are two keywords contained in a concurrency control rule and in the order "xxx a>1 and b>2". The keywords are separated by a tilde (~). In this example, the system matches only "xxx a>1~b>2", rather than "xxx b>2~a>1".
- If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.
- If you replicate data from an instance to a third-party instance based on binlogs, deleting or adding SQL statement concurrency rules from or to the instance will interrupt the replication process.
- SQL statement concurrency control does not limit concurrency for:
 - System tables
 - SQL statements used to query data, for example, **select sleep(xxx)**
 - Certain RDS for MySQL versions when the **root** account is used
 - SQL statement concurrency control is not available to stored procedures, triggers, and functions.
- Empty characters before and after each keyword will be ignored, for example, spaces, \n, \r, and \t.

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Select the required instance and click **Details**.
- Step 4** Choose **SQL > SQL Statement Concurrency Control**.
- Step 5** On the displayed page, enable **SQL Statement Concurrency Control**.
- Step 6** Click **View supported versions**.
- Step 7** Click **Add Concurrency Control Rule**. In the displayed dialog box, select the SQL statement type, enter keywords, and specify the maximum number of concurrent SQL statements. You can enter keywords or copy an existing SQL statement to the text box and click **Generate Keyword**.

 NOTE

Keyword: For example, if you set **Keyword** to **select~id~name~from~t3~where~age~>~27**, the created concurrency control rule takes effect for any SQL statement that contains keywords **select, id, name, from, t3, where, age, >**, and **27** that are separated by ~ and in the preset sequence.

Max. Concurrency: Maximum concurrent statements meeting the same rule. If the current statements exceed the upper limit, they will not be executed.

If you add a concurrency control rule and set **Max. Concurrency** to **0**, the concurrency of following SQL statements will be controlled:

"select id, name from t3 where age > 27"

"select id, name, age from t3 where age > 27"

"select id, name from t3 where age > 5 and id < 27"

But the concurrency of the following SQL statement will not be controlled:

"select name, age from t3 where age > 27"

Step 8 Confirm the settings and click **OK**.

Step 9 If a rule is no longer needed, locate the rule, click **Delete** in the **Operation** column, and click **Yes**.

----End

13.2.5 Daily Reports

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

Step 3 Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.

Step 4 On the displayed page, choose **Daily Reports > Diagnosis Overview** and click **Diagnose** to diagnose the current instance.

You can also click **View History Report** to view historical diagnosis reports.

Step 5 On the **Analysis Dimensions** page, view report details.

Step 6 Click **Specified Analysis** to view the details in a report, including slow SQL analysis, full SQL analysis, SQL type analysis, performance and disk analysis, performance trend graphs, tablespace analysis, and event analysis.

 NOTE

- Click **Download** to the upper right corner of the **Today's Report** page to download the diagnosis report to your local PC.
- Click **Subscribe** and enter your email to have the diagnosis report emailed to you. Pay attention to the following precautions before subscribing to diagnosis reports:
 - The subscription function depends on the SMN service.
 - Diagnosis reports will be emailed to you only after you have subscribed to diagnosis reports and the diagnosis result is abnormal.

----End

13.2.6 Locks and Transactions

13.2.6.1 InnoDB Lock Query

Overview

With this feature, the system displays the lock waits generated before the DML operations on the database in real time. With these waits, you can quickly locate the session waits and blocks that happened when multiple sessions update the same piece of data at the same time, and can quickly terminate the source session that holds locks to restore blocked operations.

 **NOTE**

This feature does not support DDL locks, also called metadata locks (MDLs). You view and analyze them on the **Metadata Locks** page.

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Locate your desired instance and click **Details** to go to the Intelligent O&M overview page.
- Step 4** Choose **Locks and Transactions > InnoDB Locks** to check whether there are lock waits.

----End

13.2.6.2 Metadata Locks

Overview

- Metadata locks (MDLs) are used to ensure consistency between DDL and DML operations. Usually, DDL operations require MDL write locks. Once an MDL lock occurs, it can have a significant impact on your database because all subsequent operations (SELECT, DML, and DDL operations) on the target table will be blocked, causing a backlog of connections.
- The system displays database MDL locks in real time. With these locks, you can quickly locate MDL problems and terminate the sessions holding MDL locks, so that blocked operations can be restored.

 **NOTE**

- This feature does not support DML locks. You can view and analyze them on the **InnoDB Locks** page.
- Currently, this feature is available only for MySQL 5.6 and 5.7.
- A maximum of 1,000 records can be displayed.

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Select the target instance and click **Details**.
- Step 4** Choose **Locks and Transactions > Metadata Locks**.
- Step 5** Select a lock status and type, and enter a database name, table name, and session ID as needed. Then, click **Query**.
- Step 6** In the query result, check whether there are sessions that hold MDL locks. If yes, select the sessions as required and click **Kill Session**.

----End

13.3 DDM Instances

13.3.1 Function Overview

Procedure

- Step 1** On the **Overview** page, click **Go to Intelligent O&M**.

Table 13-3 Function description

No.	Function	Description
1	Intelligent O&M	Allows you to monitor, diagnose, and tune databases.
2	Instance Overview	Displays metrics of DB instances in real time. You can specify which metrics to display.
3	Fuzzy Search	Allows you to view DB instances by DB engine, instance name, or IP address.

- Step 2** In the **Instance Overview** area, click **Set Displayed Metrics** to specify the performance metrics to display.

Table 13-4 Performance metrics

Metric	Description
QPS	Queries per second
TPS	Transactions per second
mem_usage	Instance memory usage (ratio of instance memory to OS memory)

Metric	Description
cpu_usage	Service process CPU usage (200% indicates two vCPU cores are occupied)
connection_total_count	Total connections
connection_active_count	Active connections
com_select_count	SELECT statement executions per second
com_update_count	UPDATE statement executions per second
com_insert_count	INSERT statement executions per second
com_delete_count	DELETE statement executions per second
com_replace_count	REPLACE statement executions per second
innodb_rows_deleted	Rows deleted from an InnoDB table per second
innodb_rows_read	Rows read from an InnoDB table per second
innodb_rows_inserted	Rows inserted into an InnoDB table per second
innodb_rows_updated	Rows updated in an InnoDB table per second
bytes_received	Bytes received from all clients per second
bytes_sent	Bytes sent to all clients per second
innodb_data_read	Read bytes per second in an InnoDB buffer
innodb_data_written	Write bytes per second in an InnoDB table
innodb_buffer_pool_read_requests	Times that InnoDB reads pages (logical read) from a buffer pool per second
innodb_buffer_pool_write_requests	Times that InnoDB writes data to a buffer pool per second
innodb_buf_hit	InnoDB read cache hit ratio
innodb_buf_dirty	Dirty page ratio of the InnoDB buffer pool
innodb_buf_usage	Usage of the InnoDB buffer pool
created_tmp_disk_tables	Temporary tables automatically created on hard disks when a SQL statement is executed
avg_disk_ms_per_read	Average time required for each disk read
avg_disk_ms_per_write	Average time required for each disk write
disk_read_throughput	Disk read bytes per second

Metric	Description
disk_write_throughput	Disk write bytes per second
disk_size	<ul style="list-style-type: none">• disk_total_size: Total disk space• disk_used_size: Used disk space
innodb_log_writes_requests	Writes to the redo log file per second
innodb_log_writes	Physical writes to the redo log file per second
innodb_os_log_fsyncs	fsync() writes on InnoDB log files per second
IOPS	I/O requests processed per second

Step 3 After specifying the metrics, click **OK**.

Step 4 Click **Details** of an instance to go to the Intelligent O&M details page.

----End

13.3.2 SQL

13.3.2.1 Slow Query Logs

Procedure

Step 1 Log in to the DAS console.

Step 2 On the **Overview** page, click **Go to Intelligent O&M**.

Step 3 Select the required instance and click **Details**.

Step 4 Choose **SQL > Slow Query Logs** to view log trends of the instance.

Step 5 Specify a time range or select **Last 1 hour**, **Last 6 hours**, or **Last 1 day** to view the corresponding slow query logs and CPU usage.

Step 6 Toggle on **Enable DAS Slow Query Log**. In the displayed dialog box, read and agree to the security agreement, and click **OK**.

Step 7 At the lower part of the **Log Trends** page, view statistics and details of slow query logs.

- Query and view slow query logs by database name.

Step 8 In the slow query log list, locate the log that you want to view and click **Download** to download log data.

----End

13.3.2.2 SQL Explorer

Precautions

After SQL Explorer is disabled, DAS will stop collecting new SQL statements and delete collected SQL data.

Procedure

- Step 1** Log in to the DAS console.
- Step 2** On the **Overview** page, click **Go to Intelligent O&M**.
- Step 3** Select the required instance and click **Details**.
- Step 4** Choose **SQL > SQL Explorer** to view SQL Explorer details of the instance.
- Step 5** Toggle on **Enable DAS SQL Explorer** to obtain information about all SQL statements executed on the current instance.

NOTE

- SQL Explorer takes effect once it is enabled.

- Step 6** Click the **TOP SQL** tab, select **Last 1 hour**, **Last 3 hours**, or **Last 6 hours** or specify a time range spanning no more than one day, to view the top SQL statements executed.

Figure 13-2 TOP SQL



 **NOTE**

Execution Duration Distribution displays the distribution of execution durations for all SQL templates in a specific period of time. There are four execution duration ranges. The system calculates how many times the SQL templates are executed in each execution duration range. If you select **Last 1 hour**, the execution durations are calculated every 10 seconds. If you select **Last 6 hours**, the execution durations are calculated every 1 minute. If you select a time range longer than 6 hours, the execution durations are calculated every 5 minutes. The execution duration ranges of SQL templates are as follows:

- < 100 ms
- 100 ms—500 ms
- 500 ms—1s
- > 1s

Step 7 Use either of the following method to view execution duration details, such as average execution duration, total duration, average lock wait duration, and average scanned rows.

- Hover your mouse at any point in time on the graph to view top SQL templates at that time.
- Specify a time range using your mouse on the graph, and you will see top SQL templates used during that time range.

Step 8 In the SQL template list, locate a SQL template and click **Details** to view the total execution times, average rows scanned, average execution duration, and the like.

Step 9 On the **SQL Audit** page, specify data collection time, task creation time, and node information to search for SQL audit tasks.

Step 10 Click **Add SQL Audit Task** and specify a time range to add a task. You can add instance- or node-level task based on your service requirements.

Step 11 Locate the added task in the lower part.

Step 12 Click **View Detail** in the **Operation** column to view task details.

Step 13 Specify conditions such as **Time Range**, **User**, **Keyword**, **Database** and click **Query**. The selected time range must be after the time when the SQL audit task is added.

----End

14 Audit

14.1 DAS Key Operations Supported by CTS

With CTS, you can record DAS key operations for later query, audit, and backtracking.

Table 14-1 Key operations on DAS

Operation	Resource Type	Trace Name
Adding a database login	instance	createConnection
Testing a database login	connection	testConnection
Verifying a database login	connection	verifyConnection
Modifying a database login	connection	modifyConnection
Deleting a database login	connection	deleteConnection
Logging in to a database instance	connection	databaseLoginActions
Logging in to a database instance	connection	dbConnection
Executing a SQL statement	connection	executeSql
Canceling the execution of a SQL statement	connection	cancelExecute
Obtaining result sets	connection	getExecuteResult
Splitting a SQL statement into small SQL statements	connection	splitSql
Formatting a SQL statement	connection	formatSql
Obtaining the SQL execution plan	connection	getPlan

Operation	Resource Type	Trace Name
Editing result sets	connection	editDatas
Exporting result sets	connection	exportData
Closing the SQL window	connection	closeWindow
Obtaining autofill data	connection	autoFill
Deleting executed SQL statements	connection	deleteExecSqlRecord
Creating a task for exporting SQL result sets	connection	createExportSqlResultsetTask
Creating a database export task	connection	createExportDatabaseTask
Creating a quick export task	connection	createExportQuickDatabaseTask
Deleting an export task	connection	deleteExportTask
Stopping an export task	connection	stopExportTask
Creating a task for importing a SQL file	connection	createImportSqlFileTask
Creating a task for importing a CSV file	connection	createImportCsvFileTask
Deleting an import task	connection	deleteImportTask
Stopping an import task	connection	stopImportTask
Creating a test data generation task	connection	createTestDataTask
Deleting a test data generation task	connection	deleteTestDataTask
Stopping a test data generation task	connection	stopTestDataTask-Create
Combining file fragments	connection	mergeFilePart
Uploading file fragments	connection	uploadFilePart
Executing a SQL diagnosis	connection	exeTuning
Creating an OBS bucket	connection	createBucket


14.2 Querying Traces

Scenarios

After CTS is enabled, the tracker starts recording operations on cloud resources. The last 7 days of operation records are stored on the CTS console.

This section describes how to query the last 7 days of operation records on the CTS console.

Procedure

- Step 1** Log in to the management console.
- Step 2** Under **Management & Governance**, click **Cloud Trace Service**.
- Step 3** Choose **Trace List** in the navigation pane on the left.
- Step 4** Specify filter criteria to search for the required traces. The following filters are available:
 - **Trace Type, Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
 - **Operator:** Select a specific operator from the drop-down list.
 - **Trace Status:** Available options include **All trace statuses, Normal, Warning, and Incident**. You can only select one of them.
 - **Time range:** In the upper right corner of the page, specify a time range for querying traces.
- Step 5** Click **Query**.
- Step 6** Locate the required trace and click  on the left of the trace to view its details.
- Step 7** Click **View Trace** in the **Operation** column. In the displayed dialog box, the trace structure details are displayed.
- Step 8** Click **Export** in the upper right corner of the trace list, and CTS exports traces collected in the past 7 days to a CSV file.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

A Change History

Released On	Description
2023-03-02	This issue is the first official release.