Data Admin Service

User Guide

Issue 01

Date 2025-01-07

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property 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 Cloud 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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Service Overview

1.1 What Is Data Admin Service?

Data Admin Service (DAS) is a web service that allows you to log in to and perform operations on databases.

- DAS provides a one-stop management platform for cloud database development, O&M, and intelligent diagnosis.
- DAS manages DB instances on a web console, making work easy, secure, and intelligent.

1.2 Basic Concepts

Metadata Collection

DAS originally allowed you to query metadata of databases, tables, and fields in each instance, but now it can also periodically collect metadata and store it in the DAS database.

Selecting an AZ

When determining whether to deploy resources in the same AZ, consider your applications' requirements on disaster recovery (DR) and network latency.

- For robust DR, deploy clusters in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

1.3 Advantages

Anytime, Anywhere

The DAS web console means there is no need to install clients locally and you can access your databases anytime, from anywhere.

Kernel Source Code Optimization

To address O&M pain points, the kernel is optimized and enhanced to support functions like Emergency Channel and SQL Explorer, allowing you to kill sessions that are not necessarily required in the case of an emergency and helping record and analyze all executed SQL statements.

Secure Operations

Built-in security systems protect your databases so you can worry less about security and stay focused on operations. For example, when you execute a slow SQL statement, DAS automatically triggers a timeout mechanism to protect databases from jitter.

Robust Features

DAS provides a wide range of features for you to choose from, such as SQL statement diagnosis, scheduled SQL execution, import and export of up to 1 GB of data, and cross-instance table structure synchronization.

Professional Database O&M Platform

DAS is a professional database O&M platform with SQL explorer, slow query logs, support for daily inspections, exception diagnosis, and real-time analysis. It also allows you to view performance trends and kill sessions as needed.

1.4 Permissions Management

If you need to assign different permissions to different employees in your enterprise to access your DAS resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control for your cloud resources.

With IAM, you can use your account to create IAM users, and assign permissions to the users to control their access to specific resources. For example, if you need software developers in your enterprise to be able to use DAS but not able to delete DAS resources or perform any high-risk operations, you can create IAM users for the developers and grant them only the permissions required for using DAS resources.

If your account does not require individual IAM users for permissions management, you can skip this section.

DAS Permissions

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups and attach permissions policies or roles to these groups. Users then inherit permissions from the groups they belong to and can perform specified operations on cloud services.

DAS is a project-level service deployed in specific physical regions. To assign DAS permissions to a user group, specify projects in specific regions where the

permissions will take effect. If you select **All projects**, the permissions will be granted to the user group in all projects. When accessing DAS, you need to switch to a region where you have been authorized to use this service.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization system that defines permissions related to users responsibilities. Only a limited number of service-level roles are available for authorization. When using roles to grant permissions, you may need to also assign other roles that the permissions depend on. Roles are not ideal for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization system that defines permissions required to perform operations on specific cloud resources under certain conditions. Policies are more flexible than roles, and they can ensure more secure access control. For example, you can grant IAM users only permissions for managing a certain type of database resource.

Table 1-1 lists all the system-defined roles and policies supported by DAS.

Table 1-1 DAS system permissions

Policy Name	Description	Type	Dependency
DAS Administrator	DAS administrator, who has full permissions for DAS.	System-defined role	This role depends on the **Tenant Guest** role. The **DAS Administrator** and **Tenant Guest** roles must be assigned in the same project.
DAS FullAccess	Full permissions for DAS	System-defined policy	None

NOTE

- DAS depends on other services to implement the management and O&M of databases.
- If you authorize IAM users in fine-grained mode and want to use DAS to manage DB instances, add the DAS FullAccess system policy during authorization.
- On the DAS console, you can view and manage the instances configured in the corresponding services.

By default, users with fine-grained authorization have permissions to view the database login list of Development Tool, delete database login information, and access Intelligent O&M on DAS. The instances visible to these users are the same as those configured in the corresponding services.

Table 1-2 describes the common operations supported by each system-defined policy or role of DAS. Select the policy or role you need based on the following tables.

Table 1-2 Common operations supported by each system-defined policy or role of DAS

Operation	DAS Administrator	DAS FullAccess
Logging in to a database	Supported	Supported
Adding a login	Supported	Supported
Modifying a login	Supported	Supported
Deleting a DB instance login	Supported	Supported
Viewing the login list in Development Tool	Supported	Supported
Executing a SQL diagnosis	Supported	Supported
Exporting SQL Explorer data	Supported	Supported
Subscribing to Daily Reports	Supported	Supported
Exporting slow query logs	Supported	Supported
Querying Full SQL Statements	Supported	Supported
Querying the Slow Query Log List	Supported	Supported
Viewing the Intelligent O&M page	Supported	Supported
Querying the Top SQL List	Supported	Supported
Querying the Daily Report List	Supported	Supported
Querying SQL execution plan	Supported	Supported

Table 1-3 Common DAS operations and supported actions

Operation	Action	Remarks
Logging in to a database	das:connections:login	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Obtaining the login information list	das:connections:list	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Deleting login information	das:connections:delete	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Adding a login	das:connections:create	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Modifying a database login	das:connections:modify	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Changing the payment mode of an instance on Intelligent O&M	das:clouddba:changePaymentMode	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;

Operation	Action	Remarks
Killing sessions on Intelligent O&M if necessary	das:clouddba:deleteProcess	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;
Executing a SQL diagnosis	das:clouddba:sqlDiagnosis	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;
Exporting SQL Explorer data	das:clouddba:fullSqlExport	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;
Subscribing to Daily Reports	das:clouddba:dailyReportsSubscribe	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;
Exporting slow query logs	das:clouddba:slowSqlExport	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;
Querying Full SQL Statements	das:clouddba:fullSqlList	Configure the permissions required to query other database instances based on the instance type. ● rds:instance:list; ● dds:instance:list; ● gaussdb:instance:list;

Operation	Action	Remarks
Querying the Slow Query Log List	das:clouddba:slowSqlList	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Viewing the Intelligent O&M page	das:clouddba:menuList	**NOTE** This permission is granted by IAM. After this permission is configured, you can view the Intelligent O&M page of DAS.
Querying the Top SQL List	das:clouddba:topSqlList	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Querying the Daily Report List	das:clouddba:dailyReportsList	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;
Querying SQL execution plan	das:clouddba:getSqlExecutionPlan	Configure the permissions required to query other database instances based on the instance type. • rds:instance:list; • dds:instance:list; • gaussdb:instance:list;

Table 1-4 Other permissions DAS depends on

Policy Name	Description	Type	Dependency
Tenant Administrator	Operation permissions: • All permissions on the account center, billing center, and resource center • All permissions on cloud resources owned by the account OBS policies are configured in the Global project.	System-defined role	None
OBS OperateAccess	Operation permissions: Users with this permission can view buckets, obtain basic bucket information, obtain bucket metadata, view objects, upload objects, download objects, delete objects, and obtain object ACLs. Configure the OBS policies globally.	System-defined policy	None

DAS import and export features require the usage of OBS buckets. You need to obtain required OBS permissions before using these features.

- Typically, it is recommended that you configure the Tenant Administrator policy that allows you to perform operations on OBS resources.
- If you do not want employees to have the permissions for creating and deleting buckets, you can configure the OBS OperateAccess policy for the employees so that they can use the DAS features but cannot create or delete OBS buckets.

1.5 DAS and Other Services

With DAS, you can access cloud databases with a few clicks instead of through clients.

- You can securely access data anytime and anywhere.
- You can directly manage and modify the data directory structure on the web-based console.

Relational Database Service (RDS)

DAS supports the management of RDS instances.

- You have the username and password for logging in to the target database.
- RDS instances and DAS are in the same region.

Table 1-5 DAS functions available to RDS instances

Module	MySQL	RDS for SQL Server	PostgreSQL
Database Management	√	√	√
SQL Window	√	√	√
SQL History	√	√	√
Import	√	√	√
Export	√	√	√
Table Structure Comparison and Synchronization	√	×	×
Data Generator	√	×	×
Task Scheduling	√	×	×
Real-Time Performance	√	×	×
Real-Time Sessions	√	√	×
SQL Diagnosis	√	×	×
Diagnosis Report	√	×	×
InnoDB Lock Query	√	×	×
User Management	√	√	×

Elastic Cloud Service (ECS)

DAS supports the management of ECS databases. To manage this type of databases, the following requirements must be met:

- You have the username, password, and port for logging in to the target database.
- ECSs and DAS are in the same region.
- The engine version of the managed MySQL instances can be 5.5, 5.6, 5.7, or 8.0. The instances are not deployed in HA clusters.

Table 1-6 DAS functions available for different ECS databases

Module	MySQL	RDS for SQL Server	PostgreSQL
Database Management	√	√	√

Module	MySQL	RDS for SQL Server	PostgreSQL
SQL Window	√	√	√
SQL History	√	√	√
Import	√	√	√
Export	√	√	√
Task Scheduling	√	×	×
Real-Time Performance	√	×	×
Real-Time Sessions	√	√	×
SQL Diagnosis	√	×	×
Diagnosis Report	√	×	×
InnoDB Lock Query	√	-	×
User Management	√	√	×

Document Database Service (DDS)

DAS supports the management of DDS DB instances. To manage DDS DB instances, the following requirements must be met:

- You have the username and password for logging in to the target database.
- DDS DB instances and DAS are in the same region.

Table 1-7 DAS functions available to DDS instances

Module	Function	DDS
Command Operation	To query commands.	√
	To display command execution records.	√
Database Management	To manage databases.	√
Collections	To manage database collections.	√
Views	To manage database views.	√
User Management	To create and manage users.	√
Role Management	To create and manage roles.	√

2 Getting Started

2.1 Logging In to a DB Instance

This section describes how to log in to a DB instance.

Usage Notes

The following DB instances are supported:

Table 2-1 Supported DB instances

DB Instance Source	Supported DB Engine
DB Instance	• RDS for MySQL • RDS for PostgreSQL • RDS for SQL Server • DDS • DDM
ECS-hosted DB Instance	• The engine version of the managed MySQL instances can be 5.5, 5.6, 5.7, or 8.0. The instances are not deployed in HA clusters. • Instances of PostgreSQL 9.4, 9.5, 9.6, 10, 11 and 12 are supported. • Instances of SQL Server 2008, 2012, 2014, 2016, or 2017 are supported, but HA cluster instances are not.

- The account used to create the current DB instance and the login account belong to the same account.
- The created DB instance and DAS must be in the same region.

Logging in to a DB Instance

This section describes how to log in to a DB instance. After a DB instance is created, DAS automatically creates login information for an administrator.

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the target DB instance and click **Log In** in the **Operation** column.

You need to enter the password at the first login. If **Remember Password** is selected at the first login, you do not need to enter the password again at subsequent logins.

----End

2.2 Developing or Maintaining Databases

DAS provides Development Tool and intelligent O&M to help you perform routine DB instance O&M and management.

Development Tool works an easy-to-use database client for developers. It provides diverse database development functions, including data and table structure synchronization, online editing, and intelligent prompts for SQL input.

Intelligent O&M is mainly designed for database administrators (DBAs) and provides the following database O&M functions: host and instance performance analysis, slow SQL and full SQL data analysis, real-time database performance analysis and diagnosis, and database history running data analysis, and more.

This section describes how to use DAS for RDS for MySQL instances.

Managing Databases

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Choose **RDS** and **MySQL** from the drop-down lists in the upper right corner.

Step 6 Locate the MySQL DB instance you want to log in to and click **Log In** in the **Operation** column.

Step 7 On the top menu bar, click **Database Management** and click **Change** to select a database you want to operate.

Step 8 Click the **Objects** tab to view objects such as tables, views, stored procedures, events, triggers, and functions.

----End

Database O&M

Intelligent O&M is available for paid and free instances. Paid instances can enjoy more value-added functions.

The following describes how to use Intelligent O&M to view performance trend of an RDS for MySQL instance.

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 On the **Performance** tab page, perform the following operations:

- View the trends in metrics in the same time range on different days.

  You can select **Select Date for Comparison**, and specify the target comparison date and metrics to view trends in the metrics at the same time on different days.

  You can place the pointer over a time point in the trend chart to view the metric at the time point on different days.

- View real-time performance of the instance.

  You can deselect **Select Date for Comparison**, set a time range or select **1h**, **3h**, or **12h** to view real-time metrics of the instance.

  You can place the pointer over a time point in the trend chart to view the metric at this time point.

- Customize the time range you wish to view.

  After clicking  , you can drag the mouse on the chart to select a period of time. Then, you can click **Analyze** to go to the **Slow SQL Logs** page and analyze slow query logs in the time period.

----End

3 User Guide

3.1 Permissions Management

3.1.1 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 this way, each IAM user has unique security credentials and can use DAS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or cloud service to perform professional and efficient O&M on your DAS resources.

If your account does not need individual IAM users, then you may skip over this section.

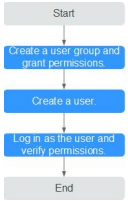
Figure 3-1 describes how to grant permissions to a user group.

Prerequisites

Before assigning permissions to user groups, you should learn about DAS system policies and select policies based on service requirements.

Process Flow

Figure 3-1 Flowchart 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 page, click **Add DB Instance Connection**. If a database connection can be created, the DAS permissions have taken effect.

3.1.2 DAS Custom Policies

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

There are two ways to create custom policies:

- 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"
      }
   ]
}
```

3.2 Logging in to a DB Instance

3.2.1 Logging in to a DB Instance

This section describes how to log in to a DB instance on DAS.

After a DB instance is created, DAS automatically creates login information for an administrator.

If you want to use a non-administrator account to log in to the DB instance, you need to create a DB instance login first. For details, see **Configuring Login Information**.

Usage Notes

- RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, RDS for MariaDB, DDS, GaussDB(for MySQL), GaussDB, and GeminiDB Cassandra instances are supported.
- The created DB instance and DAS must be in the same region.

Logging in to a DB Instance

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the target DB instance and click **Log In** in the **Operation** column.

You need to enter the password at the first login. If **Remember Password** is selected at the first login, you do not need to enter the password again at subsequent logins.

----End

Configuring Login Information

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 On the **My DB Instance Connections** tab page, click **Add DB Instance Connection**.

Step 6 Configure parameters for the new DB instance connection.

Table 3-1 Description

Parameter	Description	Example Value
DB Engine	DB engine for the instance that you want to add login information for.	MySQL

Parameter	Description	Example Value
Source Database	The source database depends on the DB engine that you select. After you select a source database, all instances of the current type under your account are displayed. You can select the instance that you want to add a login for. The following types of source databases are supported: • RDS • DDS • DDM	RDS
Login Username	Username for logging in to the selected DB instance. The user must have the permission to allow DAS to access the selected DB instance using an IP address.	root
Password	Password for logging in to the selected DB instance. You can select **Remember Password**. With this function enabled, you do not need to enter the password when you want to log in to the DB instance later.	-
Description	This parameter is optional.	-
Show Executed SQL Statements	If this function is enabled, all the SQL statements executed in the SQL window will be saved. You are advised to enable **Show Executed SQL Statements**. With it enabled, you can choose **SQL Operations** > **SQL History** to see what SQL statements you have executed and can re-execute any executed SQL statements rather than entering them again.	Enabled

Step 7 After the DB instance information is configured, click **Test Connection** on the right of the password text box.

- If the connection test is successful, go to **Step 8**.
- If a message is displayed indicating a connection test failure and failure causes, modify DB instance information based on the message and then go to **Step 8**.

Step 8 After the connection test is successful, click **OK**.

Step 9 On the **My DB Instance Connections** tab page, view information about the new DB instance connection.

You can also click **Modify** in the **Operation** column to modify the instance login information, or click **Delete** to delete unnecessary instance login information.

----End

3.2.2 Logging In to a Database Built On ECSs

3.2.2.1 Step 1: Adding DB Instance Login Information

Before logging in to a DB instance built on a ECS, you need to add a DB instance login. A created DB instance login can be modified or deleted.

Usage Notes

- If you have not logged in to a DB instance using a created login for over a year, the system will automatically delete all data of the login, including the saved password.
- You can log in to a DB instance either on the DAS console or using other clients. These two methods do not affect each other.

Adding Login Information

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Click **Add DB Instance Connection**.

Step 6 Configure parameters for the new DB instance connection.

Table 3-2 Parameter description

Parameter	Description	Example Value
DB Engine	DB engine for the instance that you want to add login information for.	MySQL
Source Database	The following types of databases are supported: • MySQL 5.5, 5.6, 5.7, and 8.0 instances. • PostgreSQL 9.4, 9.5, 9.6, 10, 11 and 12 instances. • SQL Server 2008, 2012, 2014, 2016, and 2017 instances are supported. HA clusters are not supported.	ECS
Port	When **Source Database** is set to **ECS**, you need to enter a port number for the instance that you select. Ensure that the security group of the ECS allows access to this port. To avoid conflicts with reserved ports, you are advised to enter a port number ranging from 1024 to 65535.	3306
Login Username	Username for logging in to the selected DB instance. The user must have the permission to allow DAS to access the selected DB instance using an IP address.	root
Password	Password for logging in to the selected DB instance. You can select **Remember Password**. With this function enabled, you do not need to enter the password when you want to log in to the DB instance later.	-
Description	This parameter is optional.	-

Parameter	Description	Example Value
Show Executed SQL Statements	If this function is enabled, all the SQL statements executed in the SQL window will be saved. You are advised to enable **Show Executed SQL Statements**. With it enabled, you can choose **SQL Operations** > **SQL History** to see what SQL statements you have executed and can re-execute any executed SQL statements rather than entering them again.	Enabled

Step 7 After the DB instance information is configured, click **Test Connection** on the right of the password text box.

- If the connection test is successful, go to **Step 8**.
- If a message is displayed indicating a connection test failure and failure causes, modify DB instance information based on the message and then go to **Step 8**.

Step 8 After the connection test is successful, click **OK**.

Step 9 On the **My DB Instance Connections** tab page, view information about the new DB instance connection.

You can also click **Modify** in the **Operation** column to modify the connection information or click **Delete** to delete unnecessary connection information.

----End

3.2.2.2 Step 2: Logging In to an ECS-Hosted DB Instance

After creating a DB instance login, you can log in to, operate, and manage the instance on the DAS console.

Prerequisites

You have created a DB instance login. For details, see **Step 1: Adding DB Instance Login Information**.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the target DB instance and click **Log In** in the **Operation** column.

----End

3.2.3 Logging in to a DB Instance Shared by Others

3.2.3.1 Logging in to a DB Instance Shared by Others

This section describes how to log in to a DB Instance shared by others.

Step 1 Log in to the management console as an IAM user.

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

Step 3 Click in the upper left corner, and under **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Click the **DB Instance Connections Shared by Others** tab.

Step 6 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

----End

3.3 MySQL

3.3.1 Account Management

3.3.1.1 Creating a Database User

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.

Hidden Accounts

DAS can hide your RDS for MySQL instance accounts in **Table 3-3** to ensure that your database runs normally and to prevent your account details from being deleted by mistake.

Table 3-3 Accounts that DAS can hide

Account	Description
mysql.session@localhost	Internal database account configured in MySQL 5.7 or later.

Account	Description
mysql.sys@localhost	Internal database account configured in MySQL 5.7 or later.
mysql.infoschema@localhost	Internal database account.
root@%	Account configured when you create a MySQL instance.
rdsAdmin@localhost	Management account, which has the highest permission and is used to query and modify instance information, rectify faults, as well as migrate and restore data.
rdsBackup@localhost	Backup account, which is used for backend backup.
rdsMetric	Metering account, which is used by watchdog to collect database status data.
rdsRepl	Replication account, which is used to synchronize data from primary instances to standby instances or read replicas.
rdsProxy	Proxy account, which is used for authentication when a database is connected using a 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 on DRS.
drsIncremental	Account added to the destination database when you start an incremental migration task on DRS.

Creating a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 Click **Create User**.

Configure required details, such as basic information, advanced settings, global permissions, object permissions, and roles by referring to **Table 3-4**.

Table 3-4 Parameter description

Type	Parameter	Configuration Description
Basic Information	Username	Name of the new user.
	Host	Whitelisted IP addresses allowed to access the database server. • If you leave this parameter blank or enter **%**, the user can access all IP addresses. • If you enter multiple IP addresses, 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.
	Password	Password of the user. The password: • Can include 8 to 32 characters. • Contains at least three types of the following characters: lowercase letters, uppercase letters, digits, and special characters ~!@#$%^*-_=+?,()&. | • Cannot be the username or the username spelled backwards. • Cannot be a weak password.
	Confirm Password	The value must be the same as the user password you set.
Advanced Settings	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 (of this user) to the DAS 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.
	SSL Type	Type of certificates for authenticating user requests sent to the server.
	Issuer	Authority that issues X.509 certificates for connecting the user to the server.

Type	Parameter	Configuration Description
	Subject	Subject of the X.509 certificates for connecting the user 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 current instance. For details about the global permissions that can be configured, see **Table 3-5**. For details about permission usages and explanations, visit the **official website**.	
Object Permissions	Permissions granted to the user on specific objects. For details about the object permissions that can be configured, see **Table 3-5**. For details about permission usages and explanations, visit the **official website**. Example: The following settings grant the user the permission to query the ID of table **user_test** in database **db_test**. On the **Object Permissions** tab, you can add or delete object permissions and click **Save**. In the displayed dialog box, click **OK**.	
Role	Role of the user. Roles are only supported for 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. **NOTE** To manage roles, the current account must have the **ROLE_ADMIN** permission. You can manually assign the permission. For example, you can run the **GRANT ROLE_ADMIN on** **.** **TO** *'root'@'%'* command to assign the **ROLE_ADMIN** permission to user **root**. Example: The following settings indicate that **user_2** can inherit permissions of **user@** and can grant permissions of **user@** to another user.	

Table 3-5 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.

Permission	Description
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 from a specified object.
RELOAD	Permission to execute **FLUSH**.
PROCESS	Permission to execute **SHOW PROCESSLIST** to view all the processes.
GRANT	Permission to grant a specified object the permissions of another account or reclaim permissions from other accounts.
REFERENCES	Permission to create foreign keys for a specified object.
INDEX	Permission to create or delete an index for a specified object.
ALTER	Permission to execute **ALTER TABLE** for a specified object.
SHOW DATABASES	Permission to run **SHOW DATABASES** 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 locations 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.
ALTER ROUTINE	Permission to alter or delete a stored procedure for a specified object.

Permission	Description
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.

Step 8 After configuring required parameters, click **Save**. In the preview dialog box, click **OK**.

Step 9 After the user is created, check whether the user is displayed in the user list.

----End

3.3.1.2 Editing a Database User

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

Usage Notes

Improper database user settings will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

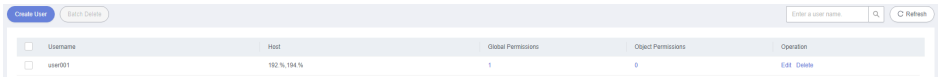
Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user whose information you want to edit and click **Edit** in the **Operation** column.

- 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 the user with the host address of 1192.% does not affect 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.%.



- For details about how to configure other parameters, see **Table 3-4** in **Creating a Database User**.

Step 8 After configuring required parameters, click **Save**. In the preview dialog box, click **OK**.

----End

3.3.1.3 Deleting a Database User

This section describes how to delete a user that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user that you want to delete and click **Delete** in the **Operation** column.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.3.2 Database Management

3.3.2.1 Creating a Database

This section describes how to create a database on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the home page, click **Create Database** above the database list.

Step 7 On the displayed page, enter a database name and select a character set.

Step 8 After settings are complete, click **OK**. View the created database in the database list.

----End

3.3.2.2 Dropping a Database

This section describes how to drop a database that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list, locate the database you want to drop and choose **More** > **Drop** in the **Operation** column.

Step 7 In the displayed dialog box, click **Yes**.

----End

3.3.2.3 Viewing Data Dictionary of a Database

A data dictionary is a collection of information that describes data, including the database name, database character set, table name, table size, number of records

in a table, table character set, table field, table index, table description, field type, field precision, and field description.

You can view data dictionary of a database on the DAS console or download the PDF file to your local PC.

Prerequisites

There are available tables in the current database.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list, locate the database whose data dictionary you want to view and click **Data Dictionary** in the **Operation** column.

Step 7 On the displayed page, view the content in **Tables** and **List of Table Schema** of the current database.

Step 8 Click **Export In PDF** to download it to your local PC if needed.

----End

3.3.3 SQL Operations

3.3.3.1 SQL Window

DAS allows you to run various SQL statements in the graphical SQL window, helping you easily manage databases.

Introduction to SQL Window

Table 3-6 Function description

Function	Description
Execute SQL	Executes SQL statements. SQL operations are performed on tables and views in a database.

Function	Description
Format SQL	Improves readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics. SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.
Execute SQL Plan	Reports execution of SQL statements to make troubleshooting easy and optimize SQL processing performance.
SQL Favorites	Allows you to add, view, and manage frequently-used SQL statements.
Save Executed SQL Statements	After this function is enabled, the system saves the recently executed SQL statements to the server.
SQL Input Prompt	Helps you quickly enter a specific database, table, or field name when you enter a SQL statement in the SQL window.
Full Screen	Shows SQL statements on a full screen.
Executed SQL Statements	Shows SQL execution details.
Messages	Shows the information returned after a SQL statement is executed.
Result Set	Shows SQL execution results. Allows you to view details about a single row, add a row, submit for editing, delete a row, export a row, copy a row, copy a column, set a column, and more.
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.

Opening the SQL Window

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

----End

3.3.3.2 Executing SQL Statements

This section describes how to execute various SQL statements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the upper left corner of the page, select a database from the drop-down list.

Step 8 Enter a SQL statement 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 when you enter a SQL statement in the SQL window.
- To protect query result sets from being displayed as garbled characters, select an encoding format other than the default encoding format, UTF-8.
- You can execute multiple SQL statements at a time. Separate them using semicolons (;). You can click **Full Screen** to view logics in SQL statements clearly.
  - To execute some of the SQL statements, select the statements before executing them.
  - To execute all SQL statements, do not select any SQL statements or select all SQL statements.

Step 9 View execution details of the current SQL statement and previously executed SQL statements in the lower part of the page.

Step 10 Click the **Messages** tab, view SQL execution details, including affected rows, progress, and time required.

Step 11 On the **Result Set** tab, view SQL execution results.

You can also perform the operations described in **Table 3-7** on result sets.

Table 3-7 Operations

Function	Description
Copy Row and Copy Column	Copies a row or column for reuse.
Column Settings	Customizes the display of columns when there are a number of columns in the query result.
Convert binary to hexadecimal	Converts binary data in the result set into hexadecimal data for display.
Refresh	Refreshes changed data.
Row Details	Displays the column field name, type, and data of the selected row.
Add Row	Adds an empty row to the result set.
Submit	Allows you to view the SQL statements to be modified. After you click **OK**, the result set is updated.
Delete Row	Deletes the selected row, including data.
Export	Allows you to export 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, data in the result set cannot be edited.
- 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, data in the result set cannot be edited.
- If only one table is in the result set and does not contain all primary key columns or any primary keys, you cannot edit the data.
- Data in virtual tables (for example, tables generated during execution of a stored procedure) cannot be edited.

----End

3.3.3.3 SQL Execution Plan

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Select a database in the upper left corner of the page.

Step 8 Enter a SQL statement in the SQL window and click **Execute SQL Plan**.

- On the **Executed SQL Statements** tab page, view SQL execution history.
- On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and time required.
- On the **Execution Plan** tab page, view SQL execution details. Execution plan details of SQL statements are displayed on different tabs.

----End

3.3.3.4 SQL Favorites

This section describes how to add frequently-used SQL statements and view and manage them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Select a database in the upper left corner of the page.

Step 8 Choose **SQL Favorites**.

- **Add SQL Statement**: allows you to add frequently-used SQL statements. Specifically, specify a title, select an application scope, enter SQL statements, and then click **Save**.
- **Insert SQL Statement**: allows you to view the SQL statements you have added. If there are no SQL statements stored in your account, this option will not be displayed.

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

----End

3.3.3.5 SQL History

After you toggle on **Save Executed SQL Statements**, DAS will save the SQL statements you executed in the SQL window for future view. You can also execute the SQL statements again with no need to enter them again in the SQL window.

Prerequisites

You have enabled **Save Executed SQL Statements** in the upper right corner of the **SQL Query** page. After this function is enabled, executed SQL statements will be saved on the DAS management host.

Constraints on Usage

SQL statement execution records can be stored for up to one year.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL History**.

Step 7 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 a SQL statement, click it in the **SQL Statement** column.
- To execute a SQL statement, click **Open in SQL window** in the **Operation** column.

----End

3.3.4 Table Management

3.3.4.1 Creating a Table

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Tables**.

Step 9 Click **Create Table**.

Step 10 Configure basic information as follows:

Enter a table name, specify the storage engine, character set, collation, and enter a comment in sequence. The table name is mandatory.

Step 11 (Optional) Configure advanced settings.

Specify parameters based on service requirements.

To learn more about partitioned tables, see the following note:

Table partitioning is used to divide a large table into several small tables based on conditions. Different rows in the table can be allocated to different physical partitions. Creating a partitioned table is 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**:

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)
);
```

Specifying a 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 12 Click **Next**.

Step 13 On the **Column** page, click **Add**.

- Set **Column Name**, **Type**, **Length**, **Nullable**, **Primary Key**, **Comment**, and **Extended Information** as needed.

  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, retain the default value. For some columns whose length is variable, you can change their lengths.
  - 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 no default values can be set.

- If you do not need to add generated columns, indexes, or foreign keys, click **Create**.

  NOTE

  - Only MySQL 5.6.5 and later versions support default DATETIME values.
  - In versions earlier than MySQL 5.6.5, leave default values blank. Otherwise, an error may occur.

- If you need to add generated columns, indexes, and foreign keys, click **Next** until all your desired parameters are specified. After the setting is complete, 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 14 In the **SQL Preview** dialog box, click **Execute**.

----End

3.3.4.2 Opening a Table

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table you want to open and click **Open** in the **Operation** column.

Step 10 On the displayed page, view table details.

On the table details page, click a cell to edit data. After adding or editing data, click **Submit** to save the changes.

You can also perform operations described in **Table 3-8** on the table.

Table 3-8 UI operations

Name	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.
Convert 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 details of a specified row.
Add Row	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.

Name	Description
Export	Exports a maximum of 10,000 data records to a CSV or SQL file.

----End

3.3.4.3 Viewing Details of a Table

This section describes how to view basic information of a table and the SQL statement for creating this table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table whose details you want to view and click **View** in the **Operation** column.

Step 10 In the displayed dialog box, view basic information of the table and SQL statements for creating the database.

----End

3.3.4.4 Altering a Table

After a table is created, you can alter information of the table, including basic information, columns, generated columns, indexes, and foreign keys.

Usage Notes

Improper alterations on a table will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table you want to alter and click **Alter** in the **Operation** column.

Step 10 Alter the table information as required. For details, see **Creating a Table**.

Step 11 After the alteration is complete, click **Alter**.

Step 12 In the **SQL Preview** dialog box, click **Execute**.

----End

3.3.4.5 Renaming a Table

This section describes how to rename a created table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table that you want to rename and click **Rename** in the **Operation** column.

Step 10 In the displayed dialog box, enter a new table name and click **OK**.

----End

3.3.4.6 Dropping a Table

This section describes how to drop a table that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table that you want to drop and choose **More** > **Drop** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.4.7 Clearing a Table

This section describes how to clear data in a table.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the table list, locate the table whose data you want to clear and choose **More** > **Clear** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.4.8 Maintaining a Table

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

Table 3-9 Function description

Function	Description
Check	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: **Check**, **Quick**, **Fast**, **Changed**, and **Extended**. The CHECK TABLE statement adds a read-only lock to the table. • **Check**: scans rows to verify that deleted links are valid. Alternatively, calculate a key checksum for the rows and verifies the validity using the obtained checksum. • **Quick**: checks only tables that have not been closed properly. • **Fast**: neither scans rows nor checks for incorrect links. • **Changed**: checks only the tables that have been changed since the last check and the tables that have not been closed properly. • **Extended**: searches for keywords in each row. This ensures that the table is 100% consistent, but takes a long time.

Function	Description
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**, **Quick**, and **Extended**. • **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.

Checking a Table

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 Locate the table you want to check and choose **More** > **Maintain** > **Check** in the **Operation** column.

Step 10 Select a check method as required.

You can check a table with any of the following methods: **Check**, **Quick**, **Fast**, **Changed**, and **Extended**.

Step 11 In the displayed dialog box, click **Yes**.

----End

Repairing a Table

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to maintain a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 Locate the table you want to repair and choose **More** > **Maintain** > **Repair** in the **Operation** column.

Step 10 Select a repair method as required.

You can repair tables using any of the following three methods: **Check**, **Quick**, and **Extended**.

Step 11 In the displayed dialog box, click **Yes**.

----End

3.3.5 View Management

3.3.5.1 Creating a View

This section describes how to create a view on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 In the upper left corner of the page, click **Create View**.

Step 10 On the displayed page, configure basic information.

Table 3-10 Parameter description

Parameter	Description
View Name	(Mandatory) Enter a view name.
Definer	Enter a definer.
Algorithm	(Optional) You can leave this parameter blank or set it to **MERGE**, **TEMPTABLE**, or **UNDEFINED**. • **UNDEFINED**: The required algorithm is automatically selected. • **MERGE**: A combination algorithm. Executing this algorithm will combine and execute SQL statements of the view and those of the external query view. • **EMPTABLE**: The result is stored in a temporary table for query.
Security	(Optional) You can leave this parameter blank or set it to **DEFINER** or **INVOKER**. • **DEFINER**: When the view is executed, the user account specified by DEFINER will be used to check access privileges for the view. • **INVOKER**: When the view is executed, the user account specified by INVOKER will be used to check access privileges for the view.
Check Option	(Optional) You can leave this parameter blank or set it to **LOCAL** or **CASCADED**. • If you select **CASCADED** for **Check Option**, the view that the current view depends on will also have the check option. • If you select **LOCAL** for **Check Option**, the system checks whether the view that the current view depends on has check options. If yes, the system checks the view that the current view depends on. If no, the system does not check the view.
View Definition Statement	Enter a SQL statement for creating the view. You only need to enter the SELECT part.

Step 11 Click **Create** at the bottom of the page.

Step 12 In the displayed dialog box, click **Execute**.

----End

3.3.5.2 Opening a View

This section describes how to view details of a view on the DAS console.

Usage Notes

Views do not have primary keys and their data can only be queried but not edited.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 Locate the view you want to open and click **Open** in the **Operation** column.

Step 10 On the displayed page, check view details and query view information using a Where condition.

You can also perform operations described in **Table 3-11** on the view.

Table 3-11 UI operations

Name	Description
Where Condition	Filters records.
Copy Row	Allows you to copy data of a selected row. Specifically, select the row you want to copy 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.
Convert 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 details of a specified row.

Name	Description
Add Row	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 to a CSV or SQL file.

----End

3.3.5.3 Altering a View

This section describes how to modify the definition, security, and algorithm of a view on the DAS console.

Usage Notes

Improper alterations on a view will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 Locate the view you want to alter and click **Alter** in the **Operation** column.

Step 10 On the displayed page, modify information of the view, including the security, algorithm, and view definition statement.

Table 3-12 Parameter description

Parameter	Description
Definer	Enter a definer.

Parameter	Description
Algorithm	(Optional) You can leave this parameter blank or set it to **MERGE**, **TEMPTABLE**, or **UNDEFINED**. • **UNDEFINED**: The required algorithm is automatically selected. • **MERGE**: A combination algorithm. Executing this algorithm will combine and execute SQL statements of the view and those of the external query view. • **EMPTABLE**: The result is stored in a temporary table for query.
Security	(Optional) You can leave this parameter blank or set it to **DEFINER** or **INVOKER**. • **DEFINER**: When the view is executed, the user account specified by DEFINER will be used to check access privileges for the view. • **INVOKER**: When the view is executed, the user account specified by INVOKER will be used to check access privileges for the view.
Check Option	(Optional) You can leave this parameter blank or set it to **LOCAL** or **CASCADED**. • If you select **CASCADED** for **Check Option**, the view that the current view depends on will also have the check option. • If you select **LOCAL** for **Check Option**, the system checks whether the view that the current view depends on has check options. If yes, the system checks the view that the current view depends on. If no, the system does not check the view.
View Definition Statement	Enter a SQL statement for creating the view. You only need to enter the SELECT part.

Step 11 Click **Alter** at the bottom of the page.

Step 12 In the displayed dialog box, click **Execute**.

----End

3.3.5.4 Dropping a View

This section describes how to drop a view that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 Locate the view you want to drop and click **Drop View** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.5.5 Viewing Details of a View

This section describes how to view details of a view, which show the SQL statement for creating it.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 Locate the view whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the view.

----End

3.3.6 Stored Procedure Management

3.3.6.1 Creating a Stored Procedure

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.

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 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Stored Procedures**.

Step 9 Click **Create Stored Procedure**.

Step 10 In the displayed dialog box, enter a stored procedure name and description.

Step 11 Click **OK**.

Step 12 Click **Option** in the upper left corner. In the displayed dialog box, set option parameters based on service requirements.

Step 13 After setting the option parameters, save or execute the stored procedure.

Step 14 View the execution status in the **Message** area.

----End

3.3.6.2 Altering or Executing a Stored Procedure

This section describes how to alter or execute a stored procedure on the DAS console.

Usage Notes

Improper alterations on a stored procedure will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Stored Procedures**.

Step 9 Locate the stored procedure you want to alter or execute and click **Alter or Execute** in the **Operation** column.

Step 10 On the displayed page, click **Option** to set parameters and click **OK**. Then, click **Save**.

Step 11 Execute the stored procedure and view the execution status in the **Message** area.

----End

3.3.6.3 Dropping a Stored Procedure

This section describes how to drop a stored procedure that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Stored Procedures**.

Step 9 Locate the stored procedure you want to drop and click **Drop** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.6.4 Viewing Details of a Stored Procedure

This section describes how to view details of a stored procedure on the DAS console. The details show the SQL statement for creating this stored procedure.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Stored Procedures**.

Step 9 Locate the stored procedure whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the stored procedure.

----End

3.3.7 Event Management

3.3.7.1 Creating an Event

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

If you set **event_scheduler** to **ON** and enable the event function, you can manage 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 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Events**.

Step 9 Click **Create Event**.

Step 10 Enter 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 11 In the displayed dialog box, click **Execute**.

If there are no errors reported, the event takes effect.

----End

3.3.7.2 Altering an Event

This section describes how to alter an event.

Usage Notes

Improper alterations on an event will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Events**.

Step 9 Locate the event you want to alter and click **Alter** in the **Operation** column.

Step 10 On the displayed page, modify the required information and click **Alter**.

Step 11 In the displayed dialog box, click **Execute**.

----End

3.3.7.3 Dropping an Event

This section describes how to drop an event that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Events**.

Step 9 Locate the event you want to drop and click **Drop** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.7.4 Viewing Details of an Event

This section describes how to view details of an event on the DAS console. The details show the SQL statement for creating the event.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Events**.

Step 9 Locate the event whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the event.

----End

3.3.8 Trigger Management

3.3.8.1 Creating a Trigger

This section describes how to create a trigger on the DAS console. A trigger is a set of SQL statements stored in the database catalog. Whenever an event

associated with a table occurs, a SQL trigger is executed or fired to insert, update, or delete data.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Triggers**.

Step 9 Click **Create Trigger**.

Step 10 On the displayed page, set required parameters and edit trigger definition statements. Then, click **Create**.

Step 11 In the displayed dialog box, click **Execute**.

----End

3.3.8.2 Altering a Trigger

This section describes how to alter the name, table, condition, and event of a trigger as well as the SQL statement for defining the trigger on the DAS console.

Usage Notes

Improper alterations on a trigger will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Triggers**.

Step 9 Locate the trigger you want to alter and click **Alter** in the **Operation** column.

Step 10 On the displayed page, modify the required parameters and click **Alter**.

Step 11 In the displayed dialog box, click **Execute**.

----End

3.3.8.3 Dropping a Trigger

This section describes how to drop a trigger that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Triggers**.

Step 9 Locate the trigger you want to drop and click **Drop Trigger** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.8.4 Viewing Details of a Trigger

This section describes how to view details of a trigger on the DAS console. The details show the SQL content for creating the trigger.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Triggers**.

Step 9 Locate the trigger whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the trigger.

----End

3.3.9 Function Management

3.3.9.1 Creating a Function

Functions can improve modularity of applications and code reusability. This section describes how to create a function on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Functions**.

Step 9 In the upper left corner of the page, click **Create Function**.

Step 10 In the displayed dialog box, enter a function name, returned value type, and description, and click **OK**.

Step 11 On the displayed page, edit the function body.

Step 12 In the upper left corner, click **Option**. In the displayed dialog box, set required parameters and click **OK**.

Step 13 Click **Execute**.

Step 14 After the function is successfully executed, view the execution status in the **Message** area.

----End

3.3.9.2 Altering or Executing a Function

This section describes how to alter or execute a function on the DAS console.

Usage Notes

Improper alterations on a function will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Functions**.

Step 9 Locate the function you want to alter or execute and click **Alter or Execute** in the **Operation** column.

Step 10 On the displayed page, alter the content as required.

Step 11 After the alteration is complete, click **Execute**.

----End

3.3.9.3 Dropping a Function

This section describes how to drop a function that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Functions**.

Step 9 Locate the function you want and click **Drop Function** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.3.9.4 Viewing Details of a Function

This section describes how to view details of a function on the DAS console. The details show the SQL statement for creating the function.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On the **Objects** tab page, choose **Functions**.

Step 9 Locate the function you want and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view function details and click **Close**.

----End

3.3.10 Data Import and Export

3.3.10.1 Importing Data

This section describes how to import data from your local PC or an OBS bucket for data backup and migration.

Usage Notes

- 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.
- Only one file that is no larger than 1 GB can be imported at a time.
- Only data files in the CSV or SQL format can be imported. If the number of MySQL 8.0 instance tables exceeds 100,000 (more than 10,000 in MySQL 5.7 and 5.6), the CSV format cannot be used.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- The size of a single SQL statement to be imported must be less than 100 MB.
- If a SQL file containing binary data is exported using the mysqldump tool, the file cannot be imported.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Import**.

Step 7 Click **Create Task**.

Table 3-13 Parameter description

Parameter	Description
Import Type	Set **Import Type** to **sql** or **CSV**.

Parameter	Description
File Source	Import a file from your local PC or an OBS bucket. • Upload file If you select **Upload file** for **File Source**, you need to set **Attachment Storage** and upload the required file. To keep your data secure, provide your own OBS bucket to store the file you uploaded. In this way, DAS automatically connects to your OBS bucket for in-memory reading. No data is stored on DAS. Creating OBS buckets is free of charge, but saving files will incur certain costs. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. • Choose from OBS If you select **Choose from OBS** for **File Source**, you need to select a file from the bucket. The file uploaded from an OBS bucket will not be deleted upon an import success.
Database	Select the database that you want to import the file to.
Charset	Select a charset as needed.
Options	If you select **Ignore errors, that is, skip the step where the SQL statement fails to be executed**, the system will skip any errors detected when SQL statements are being executed. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. This option is only available to the files uploaded from your local PC.
Remarks	Enter remarks as required.

Step 8 After setting import parameters, click **Create**.

Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 9 On the displayed page, view import progress in the task list.

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

----End

3.3.10.2 Exporting Data

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.

DAS allows you to export an entire database, some data tables, or result sets of SQL statements.

Usage Notes

- If you do not select **Generate a file for each table** when exporting data, the exported data file is in **.zip** format. Data files in this format cannot be directly imported. You need to decompress the file before importing it again.
- If **Generate a file for each table** is selected during data export, the exported data file is in **.sql** or **.csv** format. In this case, the exported data file can be directly imported again.
- If the **Exporting a Database** function is used to export over 100,000 MySQL 8.0 instance tables (more than 10,000 in MySQL 5.7 and 5.6), an error message will be displayed indicating that the number of tables is too large and data cannot be exported. In this case, use the **Exporting SQL Results** function instead.

Prerequisites

You have created a user database on the DAS console. For details, see **Creating a Database**.

Exporting a Database

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export Database**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

- If you select **Export all tables**, data in an entire database or in specific tables will be exported.
- 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 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.

- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created. You can specify the storage path.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Exporting SQL Results

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export SQL Result**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- To export multiple SQL result sets at a time, enter SQL statements in the SQL text box. Enter each SQL statement on a separate line and add a semicolon (;) at the end. After the export task is complete, SQL files are generated. One SQL statement corresponds to one file.
- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Downloading Data Files

Data exported using the data export function is stored in the OBS bucket you created. You can download exported data files in any of the following ways:

- Download on the DAS Console.
- Download on the OBS management console.

Quick Export (Not Promoted)

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Quick Export** and select the database that you want to export data from.

NOTE

A maximum of 200,000 rows can be quickly exported from a single table. To export more data, choose **Create Task** > **Export Database**.

Step 8 On the **Quick Export** page, select a storage path and click **OK**.

Step 9 In the task list, view the export task you created.

In the row that contains the export task, you can click **Details** in the **Operation** column to view execution details of the task and information about exported tables.

----End

3.3.11 Data Generator (Not Promoted)

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 can help you generate test data based on specific rules. This section describes how to generate test data on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Data Scheme** > **Data Generator**.

Step 7 On the displayed page, click **Create Task**.

Step 8 In the displayed dialog box, configure 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 Mode

  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 9 Click **Preview** to check whether the data that will be generated can meet your requirements. If not, adjust the generation rules.

Step 10 Click **Generate**.

Step 11 In the task list, locate the created task and click **Details**.

You can also delete the task as required.

----End

3.3.12 Tracking and Rolling Back Data (Not Promoted)

The data tracking and rollback function helps you track and roll back data changes. This section describes how to track and roll back data changes.

This function can be used to:

- 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 change history of a configuration table.
- Roll back data misoperations, recover the data deleted by mistake, and restore changed data to the original status. For example, the WHERE condition is not added when DBAs update data, configuration data is deleted by mistake, a large amount of dirty data and impacts are generated due to bugs.

Prerequisites

- You have enabled the data tracing and rollback function.
- The binlog function has been enabled for the destination database.

Data Tracking

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Data Scheme** > **Data Tracking and Rollback**.

- Search for a task by time range or database name.
- In the task list, locate the task and click its ID to view details.
- In the task list, click **View Task** in the **Operation** column. Then, you can view task information, search logs, and rollback tasks.
- If a task is not required any longer, you can select it and click delete in the upper lefter corner. Change history of the task will be deleted from the DAS storage.

NOTE

The default validity period of a data tracking task is 15 days. Once a task expires, the system automatically retrieves and deletes the changes in DAS.

Step 7 On the displayed page, click **Create Tracking Task**.

Step 8 In the displayed dialog box, configure required parameters.

Table 3-14 Parameters of a tracking task

Parameter	Description
Task Name	Task name, which is user-defined.
Time Range	Time range for data tracking. You are advised to select a time span less than 3 hours.
Database Name	Database whose data is to be tracked.
Table Name	Select a table to be searched. Multiple tables can be added.
Data Tracking Type	You can select multiple tracking operations. The following four options are supported: • **Update** • **Insert** • **Delete** • **DDL**

NOTE

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

Step 9 Click **Precheck** to search for binlog files based on the selected time range.

- RDS instances with the backup function enabled periodically back up binlog files and store them to your 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 10 After the precheck is successful, 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 11 After the logs are read, click **Log Search** to obtain details about change events.

- On the **Log Search** page, you can view information about the change events, such as event IDs, change time, change types, table names, impacted rows, and descriptions.
- You can also view data or rollback statements in the **Operation** column.

----End

Rolling Back Data

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Data Scheme** > **Data Tracking and Rollback**.

Step 7 Locate the task you want to view and click **View Task** in the **Operation** column.

Step 8 On the **Log Search** page, click **Create Rollback Task**. Alternatively, on the **Rollback Task List** page, click **Create Rollback Task**.

Step 9 In the displayed dialog box, configure required parameters.

Table 3-15 Parameters for creating a rollback task

Parameter	Description
Start Event ID	The start event ID is the one in the task list and must be entered in ascending order. You can view the event ID by referring to **Step 11**.
End Event ID	The end event ID is the one in the task list and must be entered in ascending order. You can view the event ID by referring to **Step 11**.
Rollback Statement Storage	Enter an OBS bucket. To keep your data secure, DAS stores rollback statements in the OBS bucket you provide. In this way, DAS automatically connects to your OBS bucket for in-memory reading. Your data will never be flushed to any storage media of DAS. Creating OBS buckets is free of charge, but saving files will incur certain costs.
Charset	Specify a character set.
Statement Type	• **Generate event rollback SQL statements**: generates SQL statements for rolling back 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.
Advanced Settings	You can also set **Data Tracking Type**, **Table Name**, **Field**, and **Field Content** under **Advanced Settings** as required.

NOTE

Changing a record (primary key) three consecutive times (for example, 1->2->3->4) equals the change of 1 to 4 (1->4).

Step 10 Click **OK**.

Step 11 In the rollback task list, view the created rollback task.

- Locate the rollback task you created and click **View Detail** in the **Operation** column.
- Click **Download** in the **Operation** column to download the original data before data change and event-based rollback SQL statements to your local PC.
- 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, sort out 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

3.3.13 Task Management

3.3.13.1 Scheduling Tasks (Not Promoted)

DAS allows you to execute SQL statements by scheduling tasks. Scheduling types include **immediate**, **scheduled**, and **periodic**. You can select a 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.

Concurrent tasks cannot be scheduled. To execute concurrent tasks, enable **Event Scheduler** to use the event capability provided by the database. For details, see .

Creating a Scheduling Task

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Background Tasks** > **Task Scheduling**.

NOTE

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

Step 7 On the **Scheduling Tasks** tab, click **Create Task**.

Step 8 On the displayed page, enter a task name and specify a scheduling type and execution time tolerance.

Table 3-16 Parameter description

Parameter	Description
Task Name	Task name, which is user-defined.

Parameter	Description
Type	The options can be **Immediate**, **Scheduled**, and **Periodic**. You can set a scheduling type as required. • **Immediate**: 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 (s)	Refers to how many seconds the system keep executing this task after the expected execution time comes. The default value is **3600**. The value ranges from **1** to **86,400**.

Step 9 Under **SQL Statements**, click **Add**. On the displayed page, set parameters as needed and click **Save**.

Table 3-17 Parameter description

Parameter	Description
Group Name	Group name, which is user-defined.
Instance	Target instance name.
Database	Associated database.
SQL Statements	You can manually input SQL statements or import existing SQL files.
Skip Errors	Configuring this parameter is recommended. After this parameter is configured, the system will skip any errors detected when SQL statements in the SQL group are being executed. If this function is disabled, the system will stop executing SQL statements.
Enable Transactions	Configuring this parameter is recommended. After this parameter is configured, SQL statements in the current SQL group will be executed as a transaction, and if a DML error occurs, a rollback will be performed. If this function is disabled, each SQL statement in the group is executed separately.
Dependent SQL Group	The system executes all SQL statements in the dependent SQL group first and then those statements in the current group.

Parameter	Description
Allow Concurrent Execution	Configuring this parameter is recommended. After this parameter is configured, the system will concurrently execute SQL statements in the current SQL group and other SQL groups, except for dependent SQL groups. **CAUTION** SQL statements in the current SQL group are executed still in serial mode.

Step 10 Click **Submit** at the bottom of the **Create Task** tab.

----End

Managing Scheduling Tasks

Tasks are displayed on the **Scheduling Tasks** and **Finished Tasks** tab pages by status.

- **Scheduling Tasks**: The scheduling tasks are periodic tasks that are being scheduled or paused.

  On the **Task Scheduling** page, click the **Scheduling Tasks** tab.

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

  You can also perform the following operations on scheduling tasks:

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

    Click the name of a scheduling task to go to the **Task Info** page and click **Manually Execute** to trigger a scheduling immediately. After the task is successfully executed, view execution details on the **Scheduling Execution Records** page.

    In the **SQL Groups** area, click **View Details**. On the displayed page, view, modify, copy, and delete SQL statements.
  - **View Execution Records**: Click it to view the task execution details and logs.
  - **View Log**: 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 **Pause**. You can also click **Resume** to restore the scheduling.
- **Finished Tasks**: Tasks in the **Finished Tasks** list are periodic tasks that have been terminated or immediate and scheduled tasks that have completed.

  On the **Task Scheduling** page, click the **Finished Tasks** tab.

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

  You can also perform the following operations on scheduling tasks:

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

    Click the name of a finished task that is periodically scheduled or starts immediately. On the displayed page, click **Manually Execute**. After the

execution completes, view execution details on the **Scheduling Execution Records** page.

In the **SQL Groups** area, click **View Details**. On the displayed page, view, modify, copy, and delete SQL statements.

- **View Execution Records**: Click it to view the task execution details, group execution status, SQL statements, and group logs.
- **View Log**: Click it to view log details.
- **Delete**: Click it to delete a task from the database.

3.3.13.2 Scheduled Backup (Not Promoted)

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

Configuring the AK/SK

Configure AK and SK before you periodically back up database tables or SQL result sets.

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Background Tasks** > **Scheduled Backup**.

Step 7 On the **Task List** tab, click **Set AK/SK** in the upper left corner.

Step 8 In the displayed dialog box, specify **Storage**, enter an AK and SK, and click **Test AK/SK**. After the test succeeds, click **OK**.

Table 3-18 Parameter description

Parameter	Description
Storage	The system stores encrypted files in OBS buckets.
AK and SK	**AK** and **SK**: Access key ID (AK) and secret access key (SK) are required credentials for you to access cloud services using development tools like APIs, CLI, and SDKs. The system uses AKs to identify users and SKs to verify encrypted signatures, ensuring that requests are secret, complete, and correct.

----End

Creating a DB Backup Task

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Background Tasks** > **Scheduled Backup**.

Step 7 On the task list page, click **Create DB Backup Task**.

Step 8 Enter a task name, select a task type, backup file format, database, and storage path, and enter remarks.

Step 9 Confirm the settings and click **OK**.

Step 10 View that the status of the current backup task is **Scheduling** in the task list.

You can also view task details and execution records and terminate or pause the task.

- You can click **Details** to view backup logs.
- You can click **View Execution Records** to view the backup status.
- You can click **Terminate** to end the backup task.
- You can click **Pause** suspend the backup task.

You can also click **Download** to download backup data.

----End

Creating a SQL Result Set Backup Task

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Background Tasks** > **Scheduled Backup**.

Step 7 On the task list tab, click **Create SQL Result Set Backup Task**.

Step 8 Enter a task name, select a task type, backup file format, database, and storage path, and enter SQL statements to be executed and remarks.

Step 9 Confirm the settings and click **OK**.

Step 10 View that the status of the current backup task is **Scheduling** in the task list.

You can also view details and execution records and terminate or suspend a scheduling task.

- You can click **Details** to view backup logs.
- You can click **View Execution Records** to view the backup status.
- You can click **Terminate** to end the backup task.
- You can click **Pause** suspend the backup task.

You can also click **Download** to download backup data.

----End

3.3.14 Table Structure Comparison and Synchronization (Not Promoted)

This section describes how to check the structure difference by comparing and synchronizing table structures when you perform a migration or verification.

Prerequisites

To create a table structure comparison and synchronization task, you need to toggle on **Save SQL** first so that DAS can store your data. If you do not toggle on **Save SQL**, you cannot create table structure comparison and synchronization tasks.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Structure Management** > **Table Structure Comparison and Synchronization**.

Step 7 On the displayed page, click **Create Task**.

Step 8 On the **Create** page, configure required parameters and click **Next**.

Table 3-19 Parameter description

Parameter	Description
Source Instance	The DB instance that you have logged in to by default.
Source Database	Select a source database.
Target Instance	Select **Current instance** or **Another instance**.
Target Database	Select a target database to be compared with.
Error Tolerance	Configuring this parameter is recommended. After this parameter is configured, if a comparison or synchronization error occurs during task execution, the system ignores the error and continues the execution.
Sync Type	• **Global**: synchronizes all tables with the same name from the source database to the target one. • **Specified**: synchronizes specific tables with the same name from the source database to the target one. • **One-to-one**: allows you to specify a source table and target table for synchronization. The two tables can have different names. • **One-to-more**: synchronizes the structure of a table in the source database to multiple tables in the target database. This option is usually selected when you want to change table structures during database and table sharding.

Step 9 Click **Next** to go to the **Compare** page, check task information and comparison items, and click **Compare**.

You can also select the items you want to skip and click **Skip** to cancel the comparison.

Step 10 View 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 11 After comparison is complete, click **Next**. On the **Synchronize** page, view basic information about this comparison task, such as the source instance, source database, target instance, target database, and synchronization type, and in the synchronization item list, specify the items to be synchronized.

Skip synchronization items that may cause high risks and then click **Synchronize**.

Step 12 After synchronization is complete, click **View Logs** in the **Operation** column to obtain comparison details.

You can download DDL as required.

----End

3.3.15 Intelligent O&M (New Version)

3.3.15.1 Dashboard

Dashboard helps you get knowledge of overall information about your instances, including instances by status, engine distribution, active alarms, slow query logs, monitoring dashboard, and resource risky instances. Intelligent Diagnosis checks instance health using instance operation data and intelligent algorithms and provides diagnosis results and suggestions.

Viewing Instances by Status

You can view the number of instances by status in the current region, including primary instances and read replicas.

Viewing Engine Distribution

You can view the distribution of engines in the current region, including engine types and abnormal engines.

Querying Active Alarms

You can view active alarms of all instances in the current region, including alarms in the Alarm (metric) and Triggered (event) statuses.

Viewing Slow Query Logs

You can view the number of slow query logs of all instances in the current region.

Monitoring Dashboards

You can view metrics of all instances in the current region.

Viewing Resource Risky Instances

You can view metric risks of all instances in the current region.

3.3.15.2 Viewing the Overall Status of an Instance

Dashboard shows the status of your instance, including alarms, resource usages, and key performance metrics. DAS diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.

Function Modules

Table 3-20 lists the function modules of the **Overview** page.

Table 3-20 Function description

Function	Description
Alarms	Shows alarms of different severities during instance running. After you click an alarm severity, the **Alarm Rules** page is displayed, showing all alarm rules of the severity.
Health	Diagnoses instance health using operational data analytics and intelligent algorithms.
Resources	Shows the CPU usage, memory usage, space usage, and disk IOPS of an instance.
Key Performance Metrics	Displays vCPU utilization & slow query logs, connections, memory utilization, and disk reads/writes in the last hour.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 On the **Dashboard** page, view status of the instance.

- Alarms

  In the **Alarms** area, view alarm information of the current instance. To view alarm details, click the number next to an alarm severity.

  To view metrics such as CPU usage, disk usage, and connection usage, click **Alarm Rules** to configure metrics and alarm policies.

- Health

  In the **Health** area, view diagnosis results of the instance.

  For example, if the vCPU utilization is high, click **Diagnose** to view related performance metrics and optimize service-related SQL statements or upgrade CPU specifications as suggested.

- Resources

  In the **Resources** area, view the resource usage of the current instance.

- Key performance metrics

In the **Key Performance Metrics** area, view key performance metrics in the last hour.

----End

3.3.15.3 Real-Time Performance

The **Real-Time Performance** tab displays key metrics of your instance and allows you to compare performance of your instance by date, helping you keep track of changes of your periodic services and metrics and detect exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 On the **Performance** tab page, perform the following operations:

- View the trends in metrics in the same time range on different days.

  You can select **Select Date for Comparison**, and specify the target comparison date and metrics to view trends in the metrics at the same time on different days.

  You can place the pointer over a time point in the trend chart to view the metric at the time point on different days.

- View real-time performance of the instance.

  You can deselect **Select Date for Comparison**, set a time range or select **1h**, **3h**, or **12h** to view real-time metrics of the instance.

  You can place the pointer over a time point in the trend chart to view the metric at this time point.

- Customize the time range you wish to view.

  After clicking  , you can drag the mouse on the chart to select a period of time. Then, you can click **Analyze** to go to the **Slow SQL Logs** page and analyze slow query logs in the time period.

----End

3.3.15.4 Real-Time Sessions

The real-time session function allows you to query session snapshots of your instance and can sort out and display only the required snapshots. You can quickly search for slow or active sessions in multiple dimensions like users, host IP

addresses, and database names. DAS allows you to kill sessions to restore your instance, securing your database availability.

Usage Notes

Killing a session may cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 On the **Sessions** tab page, perform the following operations:

- View session statistics.

  You can view statistics on the number of slow sessions, number of active sessions, and total number of sessions by user, access host, or database.

- Set a slow session threshold.

  Click **Set Slow Session Threshold**. In the displayed dialog box, set **Max. Execution Time for a Query (s)** and click **OK**. Sessions whose execution time exceeds the threshold are automatically displayed in the session list.

- Kill abnormal sessions if necessary.

  In the session list, you can view session details. You can also select the abnormal session you want to end based on session status and service requirements and click **Kill Session** to recover the database.

----End

3.3.15.5 SQL

3.3.15.5.1 Slow Query Logs

This section describes how to analyze slow query logs in a specified period. Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Viewing Slow Query Logs

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Slow Query Logs**.

If **Collect Slow Query Logs** is not enabled, enable it as prompted in the **Details** area.

You can also click **Log Settings** in the upper right corner of the page to enable **Collect Slow Query Logs**.

Step 8 On the **Slow Query Logs** tab page, perform the following operations:

- View slow queries over a specified period of time.

  Select a time range (last hour, last 3 hours, last 12 hours, or a user-defined time range which must be within three days) to view slow queries in the time range.

  In the chart, you can hover your mouse at any point in time to view slow query logs or CPU usage.

  Above the chart, you can switch to another instance or node to view slow queries by instance or by node.

- View slow query log details.

  View slow query log details in the lower part of the **Slow Query Logs** area. The details include the SQL statement, execution start time, database, client, user, execution duration, lock wait duration, and scanned and returned rows.

  Click Export to export slow query log details to a specified OBS bucket. A maximum of 100,000 records can be exported.

  After the log details are exported, you can click **View Export List** to view export records. You can also download the details to your local PC or click **Delete** to delete the slow query logs.

  Click **Concurrency Control** in the **Operation** column to create a concurrency control rule for the current SQL statement. For details, see **SQL Statement Concurrency Control**.

- View slow query log statistics.

  In the template statistics list, click **View Sample** in the **Operation** column to view the sample of a SQL statement template.

  Click Export to export a SQL template to a specified OBS bucket. A maximum of 100,000 records can be exported.

After the template is exported, you can click **View Export List** to view export records. You can also download the details to your local PC.

----End

Slow Query Log Storage and Archiving

- Slow query log storage

  Enabling **Collect Slow Query Logs** will allow DAS to store slow query logs for analysis.

- Log archiving

  Slow query logs are automatically archived every 3 minutes. Alternatively, you can click **Archive Now** and then view the latest statistics.

  Slow query logs generated in the last 10 days are archived.

  You can click **Archive History** to view archive records and click **Download** to download archived slow query logs to your local PC.

3.3.15.5.2 Top SQL

After collecting all SQL statements is enabled, Top SQL shows the SQL queries that have been contributing the most to DB load. You can sort them by multiple dimensions.

Usage Notes

- Top SQL is not available to free instances. To use this function, set free instances as paid instances or subscribe to **Intelligent O&M**.
- To use Top SQL, enable **Collect All SQL Statements** first, which is disabled by default. Collecting all SQL statements generates a performance loss of no more than 5%.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected and the collected SQL statements will be deleted.
- SQL Explorer cannot record all SQL data. As for RDS 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 removed by setting parameter **rds_sql_tracer_reserve_big_records** for RDS for MySQL 5.7.33.3 or later. You can set the parameter to **ON** by referring to , which means that SQL data records containing more than 4,096 bytes are stilled recorded.
- This function is supported only in DAS 5.6.51.1 and later, 5.7.29.2 and later, and 8.0.20.3 and later versions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Top SQL**.

Toggle on  on the right of **Collect All SQL Statements**.

You can also click **Log Settings** in the upper right corner of the page to enable **Collect All SQL Statements**.

NOTE

Collect All SQL Statements takes effect once it is enabled.

If you want to disable **Collect All SQL Statements**, click **Log Settings** in the upper right corner and toggle off the switch.

Step 8 On the **Top SQL** tab page, view execution duration of top SQL statements.

You can select **Last hour**, **Last 3 hours**, or **Last 6 hours** to view execution durations of top SQL statements. You can also set a time range that spans no more than one day.

The Execution Duration Distribution graph shows how many times SQL statements with different elapsed time lengths are executed over a specified time range. DAS collects SQL statistics of four elapsed time lengths.

- < 100 ms: number of SQL statement executions whose elapsed time is less than 100 ms
- 100 ms–500 ms: number of SQL statement executions whose elapsed time ranges from 100 ms to 500 ms
- 500 ms–1s: number of SQL statement executions whose elapsed time ranges from 500 ms to 1s
- > 1s: number of SQL statement executions whose elapsed time is longer than 1s

SQL statistics in the four elapsed time lengths are collected over the time range you specified.

- If you select **Last hour**, SQL statement executions are calculated every 10 seconds.
- If you select **Last 6 hours**, SQL statement executions are calculated every minute.
- If you select a time range longer than 6 hours, SQL statement executions are calculated every 5 minutes.

Step 9 Use either of the following methods 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 statements at that time.

- Specify a time range using your mouse on the graph and you will see top SQL statements during that time range.
- Click to export details about all top SQL templates in the list. This function is available only to paid instances.
- 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.
- Locate a SQL template and click **Concurrency Control** in the **Operation** column to create a concurrency control rule. For details, see **SQL Statement Concurrency Control**.
- Select **Date Comparison** in the upper left corner of the page to compare and analyze distributions of top SQL statements of the current instance in the same time range on different days.

----End

3.3.15.5.3 SQL Insights

The SQL Insights module allows you to query all executed SQL statements, as well as analyze and search for the tables that are accessed and updated most frequently, and the SQL statements that have the longest lock wait, helping you quickly identify exceptions and ensuring database stability.

Usage Notes

- You need to enable **Collect All SQL Statements** before using SQL Insights. Collecting all SQL statements generates a performance loss of no more than 5%.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected and the collected SQL statements will be deleted.
- 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 removed by setting parameter **rds_sql_tracer_reserve_big_records** for RDS for MySQL kernel versions 5.7.33.3 or later. RDS for MySQL 5.6 and 8.0 do not support this parameter. You can set the parameter to **ON** by referring to , which means that SQL data records containing more than 4,096 bytes are stilled recorded.
- Currently, SQL Explorer takes a period of time to consume and parse raw data. Data of TOP SQL and SQL Insights tasks has a latency of about 5 minutes.
- This function is supported only in DAS 5.6.51.1 and later, 5.7.29.2 and later, and 8.0.20.3 and later versions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Insights**.

Toggle on  on the right of **Collect All SQL Statements**.

You can also click **Log Settings** in the upper right corner of the page to enable **Collect All SQL Statements**.

NOTE

Collect All SQL Statements takes effect once it is enabled.

If you want to disable **Collect All SQL Statements**, click **Log Settings** in the upper right corner and toggle off the switch.

Step 8 Click **Create Task**. In the displayed dialog box, specify **Time Range**, **Dimension**, and other configuration items and click **OK**.

Step 9 In the task list, view the task you created.

- Locate the row containing your task and click **Details** in the **Operation** column to view task details.
- Locate your task and click **View Syn Tasks** in the **Operation** column to view synchronization task details.

----End

3.3.15.5.4 SQL Diagnosis

This function enables you to diagnose SQL statements and obtain optimization suggestions.

CAUTION

Diagnosing a large number of SQL statements will affect performance of your instance, so you are advised to use this function on an idle instance.

Usage Notes

- This function is available only to MySQL InnoDB.
- Only SELECT, INSERT, UPDATE, and DELETE statements can be diagnosed. An INSERT statement must contain a SELECT clause.
- Currently, SQL statements of databases information_schema, test, and mysql cannot be queried.

- SQL statements that contain views cannot be diagnosed.
- SQL Diagnosis obtains table structures and data distribution information (non-original). The obtained data is only for logic diagnosis, but not stored on the DAS server.
- Obtaining table structures and data distribution information may cause additional load to your instance, but has little impact on its performance.
- SQL diagnosis history is stored on the DAS server only. If you delete SQL diagnosis history on the console, it will be also deleted from the DAS server.
- SQL formatting 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.
- SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Diagnosis**. On the displayed dialog box, perform the following operations:

- Diagnose SQL statements.

  On the **SQL Diagnosis** tab page, select the database where you want to diagnose a SQL statement, enter a SQL statement, and click **Diagnose**. The diagnosis result page is displayed. You can optimize the SQL statement based on the diagnosis result.

- Execute SQL statements.

  Select a database where you want to execute a SQL statement, enter a SQL statement, and click **Execute**. The system executes the statement and displays a result set.

- Format a SQL statement.

  Select the target database, enter SQL statements, and click **Format**. The system formats all the SQL statements.

- View the SQL execution plan.

  Select the database where you want to view the SQL execution plan, enter SQL statements, and click **View Execution Plan**. The system executes all the SQL statements in sequence.

----End

3.3.15.5.5 SQL Statement Concurrency Control

Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.

Usage Notes

- 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 concurrency rules can be added for RDS for MySQL read replicas of the kernel version 5.7.38-221000 or later.
- If a SQL statement matches multiple concurrency control 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, if one rule contains the keyword **a~and~b**, the system only matches **xxx a>1 and b>2**.
- 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.
- Each SQL statement concurrency control rule can contain a maximum of 128 keywords.
- Only SELECT, UPDATE, DELETE, and INSERT statements are supported for concurrency control.
- Empty characters before and after each keyword will be ignored, for example, spaces, \n, \r, and \t.
- Too many keywords or rules may affect performance. Retain only required concurrency control rules.
- Up to 100 concurrency control rules can be retained.
- If a SQL statement has been stored in query cache, it is not subject to concurrency control.
- SQL statement concurrency control does not limit concurrency for:
  - System tables
  - SQL statements used to query data, for example, **select sleep(xxx)**
  - User **root** in versions including 5.6, 5.7.41.230700 and earlier versions in 5.7, and 8.0.28.230701 and earlier versions in 8.0
  - Stored procedures, triggers, and functions

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Statement Concurrency Control**.

Step 8 On the displayed page, toggle on **Concurrency Control**.

Step 9 In the dialog box at the top of the page, click **View Versions**.

Step 10 Click **Add Rule**. In the displayed dialog box, specify **SQL Type**, **Keyword**, and **Max. Concurrency**.

- **Keyword**: You can enter keywords.

  NOTE

  - Keywords generated from an original SQL statement are only for reference.
  - Each SQL statement concurrency control rule can contain a maximum of 128 keywords.

  The following explains how a rule matches SQL statements based on keywords:

  For example, if you enter the keywords **select~id~name~from~t3~where~age~>~27** (the keywords are separated by a tilde (~)) for a rule, the system will match them from first to last and restrict the execution of any SQL statement that contains these keywords.

- **Max. Concurrency**: SQL statements that meet the specified SQL type and keywords and exceed the value of **Max. Concurrency** will not be executed.

  NOTE

  SQL statements that have been executed before a concurrency control rule is added are not counted.

  If you add a concurrency control rule and set **Max. Concurrency** to **0**, the concurrency of the 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 11 Confirm the settings and click **OK**.

If a SQL statement concurrency control rule is no longer needed, click **Delete** in the **Operation** column.

----End

3.3.15.5.6 Auto Flow Control

Auto Flow Control allows you to set criteria, such as the vCPU threshold and maximum number of active connections. When the criteria are met, flow control will be automatically triggered to ensure stability of your workloads.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Auto Flow Control**.

Auto flow control is not available to free instances. To use the function, set free instances as paid instances or subscribe to **Intelligent O&M**.

Step 8 Click **Auto Flow Control**.

Step 9 In the displayed dialog box, set the control range, time, and policy and click **OK**.

To set a custom time window for auto flow control, contact customer service to apply for required permissions. If you have obtained the permissions, you can set a time window from 00:00 to 23:59.

Step 10 After auto flow control is enabled, determine whether to kill sessions based on service requirements. For details, see **Real-Time Sessions**.

----End

3.3.15.6 Locks and Transactions

3.3.15.6.1 InnoDB Locks

InnoDB lock waits generated before DML operations are displayed in real time. 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 terminate the source session that holds locks to restore blocked operations.

NOTE

InnoDB Locks 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 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner, and under **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Locks and Transactions** > **InnoDB Locks** to check whether there are lock waits.

----End

3.3.15.6.2 Metadata Locks

Metadata locks (MDLs) are used to ensure consistency between DDL and DML operations. Executing DDL statements on a table generates metadata write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a connection backlog.

Metadata locks are displayed in real time. You can quickly identify problems and terminate the sessions with metadata locks to restore blocked operations.

Precautions

- DML locks are not included. You can view and analyze them on the **InnoDB Locks** page.
- Metadata locks are used only in 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 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner, and under **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Locks and Transactions** > **Metadata Locks**.

Step 8 Select a lock status and type, enter a database name, table name, and session ID as needed, and click **Query**.

Step 9 In the query result, check whether there are sessions that hold MDL locks. If yes, select the sessions as required and click **Kill**.

----End

3.3.15.7 Storage Analysis

Storage occupied by data and logs and historical changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DAS can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. , **Tablespaces**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.

Viewing Storage Space Usage

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **Storage Analysis**. In the **Overview** area, view the space usage.

- The following information is displayed:
  - Storage usage
  - Available and total storage
  - Average daily increase in last week
  - Available days of storage

    NOTE

    If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.
- If the space is insufficient, click **Scale** in the upper right corner.

----End

Tablespaces

This function helps you diagnose exceptions of tables, for example, counting tables with abnormal tablespace growth, tables without primary keys, and tables without indexes.

Both automatic diagnosis and manual diagnosis are supported.

NOTE

- Tablespaces function is not available to free instances. To use the function, set free instances as paid instances or subscribe to **Intelligent O&M**.
- If the number of tables exceeds 5,000, manual diagnosis cannot be used.
- If the CPU usage exceeds 90%, manual diagnosis cannot be used.
- Tables in the **Top 50 Tables** are automatically diagnosed. Scope for manual table diagnosis is not limited.

In the **Tablespaces** area, you can view tablespace diagnosis information.

- **Auto Diagnosis**

  The system automatically diagnoses tables at about 04:00 every day.

  In the left part of the **Tablespaces** area, you can view tables whose tablespace has grown abnormally in the past day. You can click the number to view the diagnosis details.

  Any table whose tablespace has grown by more than 10,240 MB in the past day is counted. You can also click  on the right of **Auto Diagnosis** to set the upper limit for daily tablespace increase.

- **Manual Diagnosis**

  Click **Re-diagnose** to manually trigger a diagnosis task. This operation can only be performed once every 5 minutes.

  After the diagnosis is successful, you can view the number of tables without primary keys and indexes. You can click a number to view the diagnosis details.

Viewing Disk Space Distribution

You can view the distribution and change trend of the storage space.

NOTE

If the total number of files in your disk space (including data space, binlog space, slow query log space, relay log space, audit log space, temporary space, and other space) exceeds 10,000, information about the files or display disk space distribution and usages over time will not be collected on the console. This prevents performance slowdowns caused by collecting statistics on too many files. If this happens, contact technical support.

- **Data space**: Disk space occupied by user data
- **Binlog**: Disk space occupied by binlogs
- **Slow query log**: Disk space occupied by slow logs
- **Relay log**: Disk space occupied by relay logs
- **Audit log**: Disk space occupied by audit logs
- **Temporary space**: Disk space occupied by temporary files
- **Other**: Disk space reserved by the operating system for system users. About 5% of the disk space is reserved in Linux.

Top Databases and Tables by Physical File Size

You can view the top 50 databases and tables by physical file size and identify the databases and tables with high usage based on storage space distribution.

Click **View Chart** to view data volume changes in the last 7 days, last 30 days, or a custom time period (spanning no more than 30 days).

NOTE

- DAS collects data at about 04:00 every day.
- Physical file sizes are precisely recorded, but other fields' values are estimated. If there is a large gap between a file size and another field, run ANALYZE TABLE on the table.
- A database or table whose name contains special characters, including slashes (/) and #p#p, is not counted.
- Top databases and tables are available only in RDS for MySQL 5.7 and 8.0.
- If there are more than 50,000 tables in your instance, to prevent data collection from affecting the instance performance, top databases and tables will not be counted.
- Some statistics may be missing because data of database or tables is fluctuating.

3.3.15.8 Daily Reports

The **Daily Reports** page provides overall information about your instance status of the previous day, including Slow Query Logs, SQL Explorer, performance, and disk analysis. You can download and subscribe to analysis reports. A daily diagnosis is recommended.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 On the displayed page, choose **Daily Reports**.

You can perform the following operations:

- Click **Download** to download a diagnosis report to your local PC.
- Click **Subscribe** to the subscribe to daily reports. After you subscribe to daily reports, your account will be emailed when risks are identified.

NOTE

Before subscribing to diagnosis reports, note that:

- The subscription function depends on the Simple Message Notification (SMN) service.
- You can subscribe to daily reports **by topic** or **by email**.

  A topic is used to publish messages and subscribe to notifications. If you select **by topic** and the required topic is unavailable, create one first and subscribe to the topic. For details, see . Only emails are available for subscribing to topics.

  If you select **by email**, you can enter up to 15 email addresses. Separate email addresses with a semicolon (;).
- After you subscribe to daily reports, diagnosis reports will be emailed to you after you manually diagnose your instance and the result is abnormal.

- Click **Start Diagnosis** to diagnose the current instance. After the diagnosis is complete, view the diagnosis result.

  In the **Inspection Scoring** area, you can view the inspection score and details of the current instance. For more information about inspection scoring, see **Inspection Score**.

  In the **Slow SQL Analysis** area, you can view the analysis result of slow query logs of the current instance.

  In the **All SQL Analysis** area, you can view analysis of all SQL statements of the current instance.

  In the **Performance & Storage** area, you can view performance metrics for high, medium, and low watermarks and the trend chart.
- Click **View History** to view historical reports.

----End

3.3.15.9 Inspection Score

3.3.15.9.1 Inspection Score

DAS allows you to score the health of instances. You can select an inspection period and manually start an inspection to learn statuses of your instances in a timely manner.

This section describes how to score the health of instances on the DAS console.

Usage Notes

- Only RDS for MySQL instances support this function.
- For details about the scoring rules supported by DAS, see **Scoring Rules**.

Prerequisites

An RDS for MySQL instance has been logged in to and is running normally.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Select the instance whose health you want to score and click **Details**.

Step 7 On the top menu bar, choose **Daily Reports**.

Step 8 Click **Start Diagnosis** in the upper right corner of the page and specify **Time Range for Diagnosis**. The time range cannot exceed one day.

Step 9 Click **OK**.

Step 10 After the diagnosis is complete, view the inspection score in the **Inspection Scoring** area.

You can also click **Deducted Points** in the **Operation** column to view score deduction items and the deducted points.

----End

3.3.15.9.2 Scoring Rules

This section describes inspection scoring rules.

Points are deducted based on scoring rules. You can view scores on the console. For details, see **Inspection Score**.

Scoring Rules

Table 3-21 Scoring rules of RDS for MySQL instances

Score Deduction Item	Description	Sub-item	Condition	Score Calculation Rule
Memory usage	Average memory usage per day	Major event	Memory > 90%	min[3 + (Memory - 0.9) × 50, 10]
		Warning	80% ≤ Memory ≤ 90%	1 + (Memory usage – 0.8) × 20

Score Deduction Item	Description	Sub-item	Condition	Score Calculation Rule
CPU usage	Average CPU usage per day **NOTE** Multi-core CPUs need to be converted to single-core CPUs for calculation.	Major event	CPU usage ≥ 80%	min[3 + (CPU usage - 0.8) × 30, 10]
		Warning	70% ≤ CPU usage ≤ 80%	1 + (CPU usage - 0.7) × 20
Space usage	(Average used space/Total space) x 100%	Major event	Space usage > 90%	min[3 + (Space usage - 0.9) × 40, 10]
		Warning	70% ≤ Space usage ≤ 90%	1 + (Space usage - 0.7) × 20
Connection usage	Average connections per day/ Maximum of connections allowed in a day	Major event	Connection usage > 80%	3
		Warning	70% ≤ Connection rate ≤ 80%	1
IOPS usage	(Average IOPS per day/ Maximum of IOPS allowed) x 100%	Major event	IOPS usage > 90%	5
		Warning	70% < IOPS usage < 90%	3
Active sessions	Active sessions generated in a day	Major event	Active sessions > min(4 × CPU cores + 8, 96)	9
		Warning	Active sessions > min(2 × CPU cores + 8, 64)	3
Slow SQL statements	Slow SQL statements executed per day	Major event	slowSqlCount >=500	18+ (slowSqlCount-10)/30
		Warning	100 ≤ Slow SQL statements < 500	4 + (Slow SQL statements - 100)/30

Score Deduction Item	Description	Sub-item	Condition	Score Calculation Rule
		Optimization	0 < Slow SQL statements < 100	1 + (Slow SQL statements - 10)/30

3.3.15.10 Managing Anomaly Snapshots

Scenarios

This function intelligently detects instance anomalies and records information about session, lock, and transaction snapshots to facilitate subsequent fault locating.

Constraints

- Enabling anomaly collection will cause about 5% of instance performance loss.
- Each anomaly snapshot can be retained for a maximum of seven days. A maximum of 10 anomaly snapshots can be retained for each node at the same time.
- Anomaly snapshots record long-running transactions.

Enabling Anomaly Collection

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Click the **Anomaly Snapshots** tab.

Step 8 On the displayed page, toggle on the **Anomaly Collection** switch.

----End

Viewing Anomaly Snapshots

Step 1 Click the **Anomaly Snapshots** tab.

Step 2 On the displayed page, view session snapshots, metadata lock snapshots, InnoDB lock snapshots, and transaction snapshots of the DB instance.

- To view anomaly causes, click **Diagnosis Details** in the **Operation** column.
- To view details about slow SQL statements, click **Slow SQL** in the **Operation** column.

----End

3.3.16 Intelligent O&M

3.3.16.1 Performance

3.3.16.1.1 Performance History

This section describes how to use the performance history function to view the running status of an instance in a specified period.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Performance** > **Performance History**.

- Click **Select Node** to select the node whose performance history you want to view.
- Click **Select Metric**, select the metrics to display, and click **OK**.

  Set a time range in the upper right corner so that you can view the node performance during this period. You can also click **Detail** on each metric performance graph to set a time range.
- Enable **Interactive Graph** to view other metrics at the same time.

----End

3.3.16.1.2 Real-Time Performance

The **Real-Time Performance** tab displays key metrics of your instance and allows you to compare performance of your instance by date, helping you keep track of changes of your periodic services and metrics and detect exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Performance** > **Real-Time Performance**.

- View the server uptime, connection information, and buffer pool information.
- Click **Set Metric**, select the metrics to display, and click **OK**.

Table 3-22 Real-time performance metrics

Metric	Description
Slow Sessions	Number of new slow sessions
QPS | TPS	**QPS**: SQL statements executed per second **TPS**: transactions executed per second
Connections	**Total**: enabled connections **Active**: active connections
DML Executions	**select**: SELECT statements executed per second **insert**: INSERT statements executed per second **update**: UPDATE statements executed per second **delete**: DELETE statements executed per second **insert_select**: INSERT statements executed 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 written from 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

Metric	Description
InnoDB Average Row Lock Duration (ms)	Average lock time (ms) of InnoDB row locks
InnoDB Row Lock Waits	**Average InnoDB Row Lock Waits**: average waits of InnoDB row locks **Total lock waits**: total lock waits
Temporary Tables | Files	**Temporary tables**: temporary tables automatically created when SQL statements are executed **Temporary files**: temporary files automatically created when SQL statements are executed
Network Traffic	**Total**: total incoming and outgoing traffic of an instance **Incoming**: incoming traffic of an instance **Outgoing**: outgoing traffic of an instance

- Adjust the time interval of collecting metrics. The interval ranges from 1 to 10 seconds.
- Pause or start the real-time performance function based on service requirements.

----End

3.3.16.1.3 Real-Time Diagnosis

DAS allows you to use the real-time diagnosis function to view overall performance of your instance and identify exceptions quickly.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Performance** > **Real-Time Diagnosis**.

Step 8 View CPU, memory, IOPS, storage space, and connection usages of your instance.

----End

3.3.16.1.4 Performance Trends Comparisons

DAS allows you to use the performance trends comparison function to view performance trends of your instance within the same time range on different days for comparison and analysis.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Performance** > **Performance Trends Comparisons**.

- Select a node whose performance trends you want to view.
- Click **Select Metric**, select the metrics to display, and click **OK**.

  Performance history graphs of the metrics you selected are displayed. Select the same time range on different days to view the graphs. You can also click **Detail** on each metric performance graph to set a time range.
- Enable **Interactive Graph** to view other metrics at the same time.

----End

3.3.16.1.5 Custom Graphs

This section describes how to create custom graphs based on your service requirements on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Performance** > **Custom Graphs**.

Step 8 Click **Create Graph**. On the displayed page, enter a graph name and specify monitoring metrics.

Step 9 Click **OK**. On the **Custom Graphs** page, view the created graph.

----End

3.3.16.2 Sessions

3.3.16.2.1 Sessions that Can Be Killed If Necessary

You can use this function in the following scenarios:

- If the number of sessions of an instance has reached the upper limit and the instance cannot be logged in to any longer, you can view and kill unnecessary sessions using the emergency channel function.
- You can view history logs to learn details of the kill operations that you performed using the emergency channel function.

Usage Notes

- This function is not available to ECS-hosted instances and instances in the creatingor abnormal state.
- The kernel version of RDS for MySQL 5.6 must be 5.6.43.3 or later, and the kernel version of RDS for MySQL 5.7 must be 5.7.25.3 or later.
- Use this function in urgent conditions. All your kill operations will be logged.
- This function is not recommended if you can log in to your instance on the DAS console. After logging in to the instance, you can perform required operations on the **Real-Time Sessions** page.
- Sessions of sensitive users such as **rdsadmin**, **rdsbackup**, **rdsmetric**, and **rdsRepl** cannot be killed.
- When the CPU usage or the number of connections reaches the upper limit, requests to kill sessions may time out. In this case, you have to try more than once.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Sessions** > **Sessions that Can Be Killed If Necessary** and select the sessions you want to kill.

Sessions are sorted in descending order by duration.

Step 8 Select the session you want to kill and click **Kill**.

Step 9 In the displayed dialog box, confirm the session information and click **Yes**.

Step 10 Click the **History Logs** tab and view the sessions killed.

----End

3.3.16.2.2 Real-Time Sessions

The real-time session function allows you to query session snapshots of your instance and can sort out and display only the required snapshots. You can quickly search for slow or active sessions in multiple dimensions like users, host IP addresses, and database names. DAS allows you to kill sessions to restore your instance, securing your database availability.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Sessions** > **Real-Time Sessions** to view session statistics, sessions, and slow sessions.

In the statistical item list, you can view total sessions, running sessions, and longest running session time of the current instance. You can also view total sessions and active sessions by user, host, or database. In the **All Sessions** area, you can view session details and perform the following operations:

- Select the abnormal session you want to end and click **Kill** for your database to recover.
- Toggle on or toggle off **Hide System Sessions** so that the system automatically filters or displays sessions.
- Enable **Auto Refresh**. The task list is automatically refreshed every 30 seconds.

In the **All Sessions** area, you can view slow session details and perform the following operations:

- Set a slow session threshold and click the refresh button to search for the slow sessions that last longer than this threshold.
- Toggle on or toggle off **Hide System Sessions** so that the system automatically filters or displays sessions.

- Enable **Auto Refresh**. The task list is automatically refreshed every 30 seconds.
- Select the session you want to end and click **Kill** for the databases to recover.

----End

3.3.16.3 SQL

3.3.16.3.1 Slow Query Logs

This section describes how to analyze slow query logs in a specified period. Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Usage Notes

To use the slow query log function, you need to toggle on **Collect Slow Query Logs**. Then DAS will store slow query logs for analysis.

Viewing slow query logs

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **SQL** > **Slow Query Logs**.

Step 8 Select a time range and view trends, statistics, and details of the slow query logs generated within the time range.

- Log trends

  In the trends graph, you can hover your mouse at any point in time to view slow query logs or CPU usage.

  Above the chart, you can switch to another instance or node to view slow queries by instance or by node.

- Details

  View slow query log details in the lower part of the **Slow Query Logs** area. The details include the SQL statement, execution time, database name, client, user, execution duration, lock wait duration, and scanned and returned rows.

  Click **Export** to export slow query log details to a specific OBS bucket. After the details are exported, you can click **Export History** to view export records. You can also download the details to your local PC.

Locate the SQL template you want to diagnose and click **Diagnose** in the **Operation** column.

- Slow query log statistics

  Click **Sample** in the **Operation** column to view the sample of the SQL statement template.

  On the **Statistics** page, click **Export** to export slow query logs to a specific OBS bucket. Then you can click **Export History** to view export records. You can also download the exported slow query logs to your local PC.

----End

Slow Query Log Storage and Archiving

- Slow query log storage

  After **Collect Slow Query Logs** is enabled, SQL text content will be stored in OBS for analysis.

- Log archiving

  Slow query logs are automatically archived every three minutes. Alternatively, you can click **Generate Latest Log File** and then view the latest statistics.

  Slow query logs generated in the last 10 days are archived.

  You can also download archived slow query logs to your local PC.

Deleting Slow Query Logs

On the **Slow Query Logs** tab page, you can click **Delete All** or **Delete** to delete all slow query logs or some slow query logs generated in a specific period.

NOTE

Deleted slow query logs cannot be recovered. Exercise caution when performing this operation.

3.3.16.3.2 SQL Explorer

After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.

SQL Explorer can record all SQL statements executed on your instance and enables you to analyze and search for the tables that are accessed and updated the most frequently and the SQL statements that have the longest lock wait.

Usage Notes

- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected and the collected SQL statements will be deleted.
- SQL Explorer cannot record all data. As for RDS 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 RDS for MySQL kernel versions

5.7.33.3 or later. RDS for MySQL 5.6 and 8.0 do not support this parameter. For example, you can set the parameter to **ON** on the page, indicating that SQL statements whose length exceeds 4,096 bytes are still recorded.

Execution Duration Distribution of Top SQL Templates

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Explorer**.

Step 8 Toggle on **Collect All SQL Statements** to obtain information about all SQL statements executed on the current instance.

NOTE

Collect All SQL Statements takes effect once it is enabled.

Step 9 On the **Top SQL** tab page, view execution duration of top SQL statements.

You can select **Last hour**, **Last 3 hours**, or **Last 6 hours** to view execution durations of top SQL statements. You can also set a time range that spans no more than one day.

The Execution Duration Distribution graph shows how many times SQL statements with different elapsed time lengths are executed over a specified time range. DAS collects SQL statistics of four elapsed time lengths.

- < 100 ms: number of SQL statement executions whose elapsed time is less than 100 ms
- 100 ms–500 ms: number of SQL statement executions whose elapsed time ranges from 100 ms to 500 ms
- 500 ms–1s: number of SQL statement executions whose elapsed time ranges from 500 ms to 1s
- > 1s: number of SQL statement executions whose elapsed time is longer than 1s

SQL statistics in the four elapsed time lengths are collected over the time range you specified.

- If you select **Last hour**, SQL statement executions are calculated every 10 seconds.
- If you select **Last 6 hours**, SQL statement executions are calculated every minute.

- If you select a time range longer than 6 hours, SQL statement executions are calculated every 5 minutes.

Select **By instance** or **By node** to view execution duration distribution. You can change node to view its execution duration distribution. The primary node is selected by default. If you want to view the execution duration distribution of other nodes, click **Change Node**.

Step 10 In the SQL template list, 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 statements at that time.
- Specify a time range using your mouse on the graph, and you will see top SQL statements during that time range.
- Click **Export** to export details about all top SQL templates in the list. This function is available only to paid instances.
- 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 more.
- In the SQL template list, locate a SQL template and click **Diagnose** to view details such as the execution plan, and suggestions for tuning indexes and SQL statements.

----End

SQL Audit

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Explorer**.

Step 8 Click the **SQL Audit** tab.

Step 9 On the **SQL Audit** tab page, view SQL audit tasks.

- Specify the start time, end time, task creation time, and node information to search for SQL audit tasks.
- Click **Add SQL Audit Task** and specify a time range to add a task. You can add an instance- or node-level task based on your service requirements. The added SQL audit task is displayed in the list below.
- Click **Details** in the **Operation** column to view task details. You can specify filters such as **Time Range**, **User**, **Keyword**, or **Database** and click **Query** to

search for the SQL statements executed on the current instance. The selected time range must be after the time when the SQL audit task is added.

----End

3.3.16.3.3 SQL Diagnosis

This function enables you to diagnose SQL statements and obtain optimization suggestions.

CAUTION

Diagnosing a large number of SQL statements will affect performance of your instance, so you are advised to use this function on an idle instance.

Usage Notes

- This function is available only to MySQL InnoDB.
- Only SELECT, INSERT, UPDATE, and DELETE statements can be diagnosed. An INSERT statement must contain a SELECT clause.
- Currently, SQL statements of databases information_schema, test, and mysql cannot be queried.
- SQL statements that contain views cannot be diagnosed.
- SQL Diagnosis obtains table structures and data distribution information (non-original). The obtained data is only for logic diagnosis, but not stored on the DAS server.
- Obtaining table structures and data distribution information may cause additional load to your instance, but has little impact on its performance.
- SQL diagnosis history is stored on the DAS server only. If you delete SQL diagnosis history on the console, it will be also deleted from the DAS server.
- SQL formatting improves readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics.
- SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Diagnosis**. On the displayed dialog box, perform the following operations:

- On the **SQL Diagnosis** tab page, select the database where you want to diagnose a SQL statement, enter a SQL statement, and click **Diagnose**. The diagnosis result page is displayed.
- Select a database where you want to execute a SQL statement, enter a SQL statement, and click **Execute**. The system executes the statement and displays a result set.
- Select the target database, enter SQL statements, and click **Format**. The system formats all the SQL statements.
- Select the database where you want to view the SQL execution plan, enter SQL statements, and click **View Execution Plan**. The system executes all the SQL statements in sequence.

----End

3.3.16.3.4 SQL Statement Concurrency Control

Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.

Usage Notes

- 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 concurrency rules can be added for RDS for MySQL read replicas of the kernel version 5.7.38-221000 or later.
- If a SQL statement matches multiple concurrency control 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, if one rule contains the keyword **a~and~b**, the system only matches **xxx a>1 and b>2**.
- 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.
- Each SQL statement concurrency control rule can contain a maximum of 128 keywords.
- Currently, SQL Statement Concurrency Control supports only keywords **SELECT**, **UPDATE**, and **DELETE**.
- Empty characters before and after each keyword will be ignored, for example, spaces, \n, \r, and \t.
- Too many keywords or rules may affect performance. Retain only required concurrency control rules.

- Up to 100 concurrency control rules can be retained.
- If a SQL statement has been stored in query cache, it is not subject to concurrency control.
- SQL statement concurrency control does not limit concurrency for:
  - System tables
  - SQL statements used to query data, for example, **select sleep(xxx)**
  - User **root** in versions including 5.6, 5.7.41.230700 and earlier versions in 5.7, and 8.0.28.230701 and earlier versions in 8.0
  - Stored procedures, triggers, and functions

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Statement Concurrency Control**.

Step 8 On the displayed page, toggle on **Concurrency Control**.

Step 9 Click **View Supported Versions**.

Step 10 Click **Add Concurrency Control Rule**. In the displayed dialog box, select a SQL statement type, enter keywords, and specify the maximum number of concurrent SQL statements.

- **Keyword**: You can enter keywords or copy an existing SQL statement to the text box and click **Generate Keyword**.

  NOTE

  - Keywords generated from an original SQL statement are only for reference.
  - Each SQL statement concurrency control rule can contain a maximum of 128 keywords.

  The following explains how a rule matches SQL statements based on keywords:

  For example, if you enter the keywords **select~id~name~from~t3~where~age~>~27** (the keywords are separated by a tilde (~)) for a rule, the system will match them from first to last and restrict the execution of any SQL statement that contains these keywords.
- **Max. Concurrency**: SQL statements that meet the specified SQL type and keywords and exceed the value of **Max. Concurrency** will not be executed.

NOTE

SQL statements that have been executed before a concurrency control rule is added are not counted.

If you add a concurrency control rule and set **Max. Concurrency** to **0**, the concurrency of the 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 11 Confirm the settings and click **OK**.

If a SQL statement concurrency control rule is no longer needed, click **Delete** in the **Operation** column.

----End

3.3.16.4 Locks and Transactions

3.3.16.4.1 InnoDB Locks

InnoDB lock waits generated before DML operations are displayed in real time. 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 terminate the source session that holds locks to restore blocked operations.

NOTE

InnoDB Locks does not support metadata locks (MDLs). You view and analyze them on the **Metadata Locks** page.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Locks and Transactions** > **InnoDB Locks** to check whether there are lock waits.

----End

3.3.16.4.2 Metadata Locks

Metadata locks (MDLs) are used to ensure consistency between DDL and DML operations. Executing DDL statements on a table generates metadata write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a connection backlog.

Metadata locks are displayed in real time. You can quickly identify problems and terminate the sessions with metadata locks to restore blocked operations.

Precautions

- This function does not support DML locks. You can view and analyze them on the **InnoDB Locks** page.
- Metadata locks are used only in 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 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner, and under **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 Choose **Locks and Transactions** > **Metadata Locks**.

Step 8 Select a lock status and type, enter a database name, table name, and session ID as needed, and click **Query**.

Step 9 In the query result, check whether there are sessions that hold MDL locks. If yes, select the sessions as required and click **Kill**.

----End

3.3.16.5 Daily Reports

The **Daily Reports** page provides overall information about your instance status of the previous day, including Slow Query Logs, SQL Explorer, performance, and disk analysis. You can download and subscribe to analysis reports. A daily diagnosis is recommended.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the box containing your target instance and click **Details**.

Step 7 On the displayed page, choose **Daily Reports**.

You can perform the following operations:

- Click **Download** to download a diagnosis report to your local PC.
- Click **Subscribe** and enter your email to have the diagnosis report emailed to you.

  NOTE

  Before subscribing to diagnosis reports, note that:

  – The subscription function depends on the Simple Message Notification (SMN) service.

  – After you subscribe to daily reports, diagnosis reports will be emailed to you after you manually diagnose your instance and the result is abnormal.
- Click **Diagnose** to diagnose the current instance.
- Click **View History Report** to view historical diagnosis reports.
- On the **Diagnosis Overview** page, view diagnosis results of the instance.
- On the **Analysis Dimensions** page, view diagnosis dimensions, including Slow SQL Analysis, Full SQL Analysis, and Performance & Disk Analysis.
- On the **Specified Analysis** page, view details in a report, including slow SQL analysis, full SQL analysis, and performance and disk analysis.

----End

3.4 RDS for SQL Server

3.4.1 Account Management

3.4.1.1 Creating a Login Name

DAS allows you to create login accounts with different permissions for accessing databases to enhance data management security.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **Login Name**.

Step 7 Click **Create Login Name**.

Step 8 Set parameters on the **Basic Settings**, **Server Roles**, **User Mapping**, **Securables**, and **Status** tab pages separately, and click **Save**.

Table 3-23 Login name information

Parameter	Description
Login Name	Name of the account for logging in to the instance.
Authentication Type	The default value is **RDS for SQL Server Authentication**.
Password	Password of the login account. The password: ● Can contain 8 to 128 characters. ● Must contain at least three types of lowercase letters, uppercase letters, digits, and the following special characters: ~!@#%^*-_=+?$ ● Cannot contain the login name. ● Cannot be a weak password.
Confirm Password	The confirm password must be the same as the password you set.
Enforce Password Policy	If you select this option, the password must meet security rules of the SQL Server instance.
Default Database	Database that is logged in by default.
Default Language	Default language of the database.
Server Roles	Configure server roles for the login name.
User Mapping	Configure specific permissions for the login name on specific databases. **NOTE** If you click **User Mapping** and select the required databases, the system will delete user information in the **User** column and create new users accordingly.

Parameter	Description
Securables	Configure server-level permissions for the login name.
Status	If you select **Enable**, you allow the login name to log in to the current instance. If you select **Disable**, you do not allow the login name to log in to the instance.

Step 9 Click **Back to Login Name List** and view the newly-added login name.

Step 10 In the login name list, edit, rename, or delete the created login names as required.

----End

3.4.1.2 Database User Management

Multiple users with different permissions can be created to access a database, but the permissions of these users must be within the range of the account permissions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **Database User Management**.

Step 7 On the displayed page, select a database and click **Create User**.

Step 8 On the displayed page, set parameters on the **Basic Settings**, **Owned Schemas**, **Membership**, and **Securables** tab pages separately, and click **Save**.

Table 3-24 Database user information

Parameter	Description
User Type	You can select **SQL user with login** or **SQL user without login**. Users without login names cannot log in, but they can be assigned permissions.
Username	Name of the database user.

Parameter	Description
Login Name	Login name associated with the database user. When the database user logs in to a database using the login name, the permissions of the login name are also the permissions of the database user.
Default Schema	The first schema that the server searches for when it resolves object names for this database user.
Owned Schemas	All schemas owned by the database user.
Membership	All roles owned by the database user. If a user has a role, it has all permissions of the role.
Securables	The user's specific permissions on database objects.

Step 9 After configuring required parameters, click **Save**. In the preview dialog box, click **OK**.

Step 10 In the user management list, edit, rename, or delete the created users.

----End

3.4.1.3 Database Role Management

This section describes how to create roles on the DAS console and manage these roles.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **Database Role Management**.

Step 7 On the displayed page, select a database and click **Create Role**.

Step 8 On the displayed page, set parameters on the **Basic Settings**, **Owned Schemas**, **Role Member**, and **Securables** tab pages.

Table 3-25 Database role information

Parameter	Description
Role Name	Name of the database role.
Owner	Owner of the role.
Owned Schemas	All schemas owned by the database role.
Role Member	Database users that own the role. If a user has a role, it has all permissions of the role.
Securables	The role's specific permissions on database objects.

Step 9 After configuring required parameters, click **Save**. In the preview dialog box, click **OK**.

Step 10 In the role management list, view information of database roles.

You can also edit, rename, and delete created database roles.

----End

3.4.2 Database Management

3.4.2.1 Creating a Database

This section describes how to create a database on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list of the **Home** page, click **Create Database**.

Step 7 On the displayed page, enter a database name and set information including recovery mode, compatibility level, and database read-only.

Table 3-26 Parameter description

Parameter	Description
Name	Enter a database name.
Recovery Mode	The value is **Full** by default, indicating a full recovery for the database.
Compatibility Level	Multiple versions of RDS for SQL Server are available. The compatibility varies with the database version. A later version is recommended.
Containment Type	The value can be **None** or **Partial**. This parameter is available only for RDS for SQL Server 2012 (11.x) and later.
Database Read-only	The value can be **False** or **True**. If **True** is selected, a read-only database is created.
Allow Snapshot Isolation	The value can be **False** or **True**. Value **True** indicates that snapshots are isolated in emergent scenarios to protect the database data.
Database Files	By default, row database files and log database files can be added. You can also modify the initial and autogrowth sizes of existing database files if needed.

Step 8 Confirm the settings and click **OK**. View the created database in the database list.

----End

3.4.2.2 Dropping a Database

This section describes how to drop a database that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list, locate the database you want to drop and choose **More** > **Drop** in the **Operation** column.

Step 7 In the displayed dialog box, click **Yes**.

----End

3.4.3 SQL Operations

3.4.3.1 SQL Window

DAS allows you to run various SQL statements in the graphical SQL window, helping you easily manage databases.

Introduction to SQL Window

Table 3-27 Function description

Function	Description
Execute SQL	Executes SQL statements. SQL operations are performed on tables and views in a database.
Format SQL	Improves readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logics and semantics. SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.
Execute SQL Plan	Reports execution of SQL statements to make troubleshooting easy and optimize SQL processing performance.
SQL Favorites	Allows you to add, view, and manage frequently-used SQL statements.
Save Executed SQL Statements	Saves the recently executed SQL statements to the server.
SQL Input Prompt	Helps you quickly enter a specific database, table, or field name when you enter a SQL statement in the SQL window.
Full Screen	Shows SQL statements on a full screen.
Executed SQL Statements	Shows the SQL execution details.
Messages	Shows the information returned after a SQL statement is executed.

Function	Description
Result Set	Shows SQL execution results. Allows you to view details about a single row, add a row, submit for editing, delete a row, export a row, copy a row, copy a column, set a column, and more.
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.

Opening the SQL Window

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

----End

3.4.3.2 Executing SQL Statements

This section describes how to execute various SQL statements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the upper left corner of the page, select a database and schema.

Step 8 Enter a SQL statement 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 when you enter a SQL statement in the SQL window.
- To protect query result sets from being displayed as garbled characters, select an encoding format other than the default encoding format, UTF-8.
- You can execute multiple SQL statements at a time. Separate them using semicolons (;). You can click **Full Screen** to view logics in SQL statements clearly.
  - To execute some of the SQL statements, select the statements before executing them.
  - To execute all SQL statements, do not select any SQL statements or select all SQL statements.

Step 9 View execution details of the current SQL statement and previously executed SQL statements in the lower part of the page.

Step 10 Click the **Messages** tab, view SQL execution details, including affected rows, progress, and time required.

Step 11 On the **Result Set** tab, view SQL execution results.

You can also perform the operations described in **Table 3-28** on result sets.

Table 3-28 Operations

Function	Description
Copy Row and Copy Column	Copies a row or column for reuse.
Column Settings	Customizes the display of columns when there are a number of columns in the query result.
Convert binary to hexadecimal	Converts binary data in the result set into hexadecimal data for display.
Refresh	Refreshes changed data.
Row Details	Displays the column field name, type, and data of the selected row.
Add Row	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 data.
Export	Allows you to export data in a SQL or CSV file. A maximum of 10,000 rows of data can be exported.

Function	Description
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, data in the result set cannot be edited.
- 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, data in the result set cannot be edited.
- If only one table is in the result set and does not contain all primary key columns or any primary keys, you cannot edit the data.
- Data in virtual tables (for example, tables generated during execution of a stored procedure) cannot be edited.

----End

3.4.3.3 Executing SQL Plan

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Enter a SQL statement in the SQL window and click **Execute SQL Plan**.

- On the **Executed SQL Statements** tab page, view SQL execution history.
- On the **Messages** tab page, view execution information, including SQL splitting, execution status, and elapsed time.
- On the **Execution Plan** tab page, view SQL execution details. Execution plan details of SQL statements are displayed on different tabs.

----End

3.4.3.4 SQL Favorites

This section describes how to add frequently-used SQL statements and view and manage them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the SQL window, click **SQL Favorites**.

- **Add SQL Statement**: allows you to add frequently-used SQL statements. Specifically, specify a title, select an application scope, enter SQL statements, and then click **Save**.
- **Insert SQL Statement**: allows you to view the SQL statements you have added. If there are no SQL statements stored in your account, this option will not be displayed.
- **Manage SQL Statements**: allows you to manage frequently-used SQL statements, including adding, editing, and deleting statements.

----End

3.4.3.5 SQL History

After you toggle on **Save Executed SQL Statements**, DAS will save the SQL statements you executed in the SQL window for future view. You can also execute the SQL statements again with no need to enter them again in the SQL window.

Prerequisites

You have enabled **Save Executed SQL Statements** in the upper right corner of the **SQL Query** page. After this function is enabled, executed SQL statements will be saved on the DAS management host.

Constraints on Usage

SQL statement execution records can be stored for up to one year.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 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 a SQL statement, click it in the **SQL Statement** column.
- To execute a SQL statement, click **Open in SQL window** in the **Operation** column.

----End

3.4.4 Table Management

3.4.4.1 Creating a Table

A data table consists of basic information, columns, indexes, foreign keys, and CHECK constraints. The indexes, foreign keys, and CHECK constraints are optional and can be configured based on service requirements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to create a table.

Step 8 On displayed page, click **Create Table**.

Step 9 On the displayed **Basic Information** page, enter a table name, specify the schema, table lock escalation, and enter comments.

Step 10 Click **Next**.

Step 11 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 setting is complete, click **Create**.

Step 12 In the **SQL Preview** dialog box, click **Execute**.

----End

3.4.4.2 Opening a Table

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to open a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left. In the table list, locate the table you want to open, and click **Open** in the **Operation** column.

Step 9 On the table details page, double-click a cell to edit data. After adding or editing data, submit and save the changes.

Table 3-29 UI operations

Name	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.
Refresh	Allows you to update table data manually.
Row Details	Shows details of a specified row.
Add Row	Allows you to add rows.

Name	Description
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 to a CSV or SQL file.

----End

3.4.4.3 Viewing Details of a Table

This section describes how to view basic information of a table and the SQL statement for creating this table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of current database to switch to the database where you want to view details of a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left. In the right pane, locate the table whose details you want to view and click **View** in the **Operation** column.

Step 9 In the displayed dialog box, view basic table information and DDL information.

----End

3.4.4.4 Altering a Table

After a table is created, you can alter information of the table, including basic information, columns, generated columns, indexes, and foreign keys.

Usage Notes

Improper alterations on a table will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to alter a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the right pane, locate the table that you want to alter and click **Alter** in the **Operation** column.

Step 10 Modify the table information as required. After the modification is complete, click **Alter**.

Step 11 In the **SQL Preview** dialog box, click **Execute**.

----End

3.4.4.5 Renaming a Table

This section describes how to rename a created table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to rename a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the right pane, locate the table that you want to rename and click **Rename** in the **Operation** column.

Step 10 In the displayed dialog box, enter a new table name and click **OK**.

----End

3.4.4.6 Clearing a Table

This section describes how to clear data in a table.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to clear a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the right pane, locate the table you want to clear, and choose **More** > **Clear** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.4.4.7 Dropping a Table

This section describes how to drop a table that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to delete a table.

Step 8 On the displayed **Objects** tab, choose **Tables** on the left.

Step 9 In the right pane, locate the table you want to drop and choose **More** > **Drop** in the **Operation** column.

Step 10 In the displayed dialog box, click **Yes**.

----End

3.4.5 View Management

3.4.5.1 Creating a View

This section describes how to create a view on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to create a view.

Step 8 On the displayed **Objects** page, select **Views**. In the upper left corner of the page, click **Create View**.

Step 9 On the displayed page, click **Set Template Parameter**. In the displayed dialog box, set parameters as required and click **Apply Template**.

Step 10 Click **Save**.

----End

3.4.5.2 Dropping a View

This section describes how to drop a view that is no longer needed.

Usage Notes

A dropped view cannot be recovered.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to delete a view.

Step 8 On the displayed **Objects** page, choose **Views**. In the right pane, locate the view that you want to drop and click **Drop View** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.4.5.3 Opening a View

This section describes how to view details of a view on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to open a view.

Step 8 On the displayed **Objects** page, choose **Views**. In the right pane, locate the view that you want to open and click **Open** in the **Operation** column.

Step 9 On the displayed page, check view details and query view information using a Where condition.

Table 3-30 UI operations

Name	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.
Refresh	Allows you to update table data manually.
Row Details	Shows details of a specified row.
Export	Exports a maximum of 10,000 data records to a CSV or SQL file. If you choose **Export** > **Export More**, the system directs you to the **Export** page, and you can create an export task or quickly export data.

NOTE

Views do not have primary keys and their data can only be queried.

----End

3.4.5.4 Altering a View

Usage Notes

Improper alterations on a view will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to alter a view.

Step 8 On the displayed **Objects** page, choose **Views**. In the right pane, locate the view that you want to alter and click **Alter** in the **Operation** column.

Step 9 Modify the view information as required and click **Save**.

----End

3.4.5.5 Viewing Details of a View

This section describes how to view details of a view, which show the SQL statement for creating it.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to view details of a view.

Step 8 On the **Objects** tab page, choose **Views**.

Step 9 Locate the view whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the view.

----End

3.4.6 Stored Procedure Management

3.4.6.1 Creating a Stored Procedure

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.

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 PROCEDURE permission.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to create a stored procedure.

Step 8 On the displayed **Objects** page, select **Stored Procedures**. In the upper left corner of the page, click **Create Stored Procedure**.

Step 9 On the displayed page, click **Set Template Parameter**. In the displayed dialog box, set parameters as required and click **OK**.

Step 10 On the page for creating a stored procedure, click **Execute**.

Step 11 In the displayed dialog box, set input parameters and click **Execute**.

Step 12 View the execution status on the **Message** and **Result Set** tab pages after the execution of the stored procedure.

----End

3.4.6.2 Altering or Executing a Stored Procedure

This section describes how to alter or execute a stored procedure on the DAS console.

Usage Notes

Improper alterations on a stored procedure will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to alter or execute a stored procedure.

Step 8 On the displayed **Objects** page, choose **Stored Procedures**. In the right pane, locate the stored procedure you want to alter or execute and click **Alter or Execute** in the **Operation** column.

Step 9 On the displayed page, edit the content and click **Save**.

Step 10 In the displayed dialog box, set input parameters and click **Execute**.

Step 11 View the execution status in the **Message** area.

----End

3.4.6.3 Dropping a Stored Procedure

This section describes how to drop a stored procedure that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to delete a stored procedure.

Step 8 On the displayed **Objects** page, choose **Stored Procedures**. In the right pane, locate the stored procedure you want to drop and click **Drop** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.4.6.4 Viewing Details of a Stored Procedure

This section describes how to view details of a stored procedure on the DAS console. The details show the SQL statement for creating this stored procedure.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to view details of a stored procedure.

Step 8 On the displayed **Objects** page, choose **Stored Procedures**. In the right pane, locate the stored procedure you want to view and click **View Details** in the **Operation** column.

Step 9 In the displayed dialog box, view the stored procedure details.

----End

3.4.7 Trigger Management

3.4.7.1 Creating a Trigger

This section describes how to create a trigger on the DAS console. A trigger is a set of SQL statements stored in the database catalog. Whenever an event associated with a table occurs, a SQL trigger is executed or fired to insert, update, or delete data.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to create a trigger.

Step 8 On the displayed **Objects** page, choose **Db Triggers**. In the upper left corner of the page, click **Create Trigger**.

Step 9 On the displayed page, click **Set Template Parameter**.

Step 10 In the displayed dialog box, set parameters and click **OK**.

Step 11 Edit the content and click **Save**.

----End

3.4.7.2 Altering a Trigger

This section describes how to alter the name, table, condition, and event of a trigger as well as the SQL statement for defining the trigger on the DAS console.

Usage Notes

Improper alterations on a trigger will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to alter a trigger.

Step 8 On the displayed **Objects** page, select **Db Triggers**. In the right pane, locate the trigger you want to alter and click **Alter** in the **Operation** column.

Step 9 On the displayed page, edit the content and click **Save**.

----End

3.4.7.3 Dropping a Trigger

This section describes how to drop a trigger that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to delete a trigger.

Step 8 On the displayed **Objects** page, select **Db Triggers**. In the right pane, locate the trigger you want to drop and click **Drop Trigger** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.4.7.4 Viewing Details of a Trigger

This section describes how to view details of a trigger on the DAS console. The details show the SQL content for creating the trigger.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to view details of a trigger.

Step 8 On the **Objects** tab page, choose **Triggers**.

Step 9 Locate the trigger whose details you want to view and click **View Details** in the **Operation** column.

Step 10 In the displayed dialog box, view the SQL statement for creating the trigger.

----End

3.4.8 Function Management

3.4.8.1 Creating a Function

Functions can improve modularity of applications and code reusability. This section describes how to create a function on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to create a function.

Step 8 On the displayed **Objects** page, select **Functions**, select a schema, and click **Create Function**.

Step 9 In the displayed dialog box, select a function type and click **OK**.

Step 10 On the displayed page, click **Set Template Parameter**. In the displayed dialog box, set parameters as required and click **OK**.

Step 11 Click **Save** to complete the creation, or click **Execute** to invoke the function and view the result in the **Message** area.

----End

3.4.8.2 Altering or Executing a Function

This section describes how to alter or execute a function on the DAS console.

Usage Notes

Improper alterations on a function will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to alter or execute a function.

Step 8 On the **Objects** page, select **Functions**. In the right pane, locate the function you want to alter or execute and click **Alter or Execute** in the **Operation** column.

Step 9 On the displayed page, modify the content as required. Then, click **Save** to complete the modification, or click **Execute** to invoke the function.

Step 10 View the execution result.

----End

3.4.8.3 Dropping a Function

This section describes how to drop a function that is no longer needed.

Usage Notes

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to delete a function.

Step 8 On the **Objects** page, select **Functions**. In the right pane, locate the function you want to drop and click **Drop** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.4.8.4 Viewing Details of a Function

This section describes how to view details of a function on the DAS console. The details show the SQL statement for creating the function.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 In the upper area of the page, click **Change** on the right of the current database to switch to the database where you want to view details of a function.

Step 8 On the **Objects** page, select **Functions**. In the right pane, locate the function you want to view and click **View Function** in the **Operation** column.

Step 9 In the displayed dialog box, view the function details.

----End

3.4.9 Data Import and Export

3.4.9.1 Importing Data

This section describes how to import data from your local PC or an OBS bucket for data backup and migration.

Usage Notes

- 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.
- Only one file that is no larger than 1 GB can be imported at a time.

- Only data files in the CSV or SQL format can be imported. If the number of instance tables exceeds 10,000, the CSV format cannot be used.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- The size of a single SQL statement to be imported must be less than 100 MB.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Import**.

Step 7 Click **Create Task**.

Table 3-31 Parameter description

Parameter	Description
Import Type	Set **Import Type** to **sql** or **CSV**.
File Source	Import a file from your local PC or an OBS bucket. • Upload file If you select **Upload file** for **File Source**, you need to set **Attachment Storage** and upload the required file. To keep your data secure, provide your own OBS bucket to store the file you uploaded. In this way, DAS automatically connects to your OBS bucket for in-memory reading. No data is stored on DAS. Creating OBS buckets is free of charge, but saving files will incur certain costs. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. • Choose from OBS If you select **Choose from OBS** for **File Source**, you need to select a file from the bucket. The file uploaded from an OBS bucket will not be deleted upon an import success.
Database	Select the database that you want to import the file to.

Parameter	Description
Charset	Select a charset as needed.
Options	If you select **Ignore errors, that is, skip the step where the SQL statement fails to be executed**, the system will skip any errors detected when SQL statements are being executed. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. This option is only available to the files uploaded from your local PC.
Remarks	Enter remarks as required.

Step 8 After setting import parameters, click **Create**.

Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 9 On the displayed page, view import progress in the task list.

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

----End

3.4.9.2 Exporting Data

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.

DAS allows you to export an entire database, some data tables, or result sets of SQL statements.

Usage Notes

- If you do not select **Generate a file for each table** when exporting data, the exported data file is in **.zip** format. Data files in this format cannot be directly imported. You need to decompress the file before importing it again.
- If **Generate a file for each table** is selected during data export, the exported data file is in **.sql** or **.csv** format. In this case, the exported data file can be directly imported again.
- To export over 10,000 tables of RDS for SQL Server instances, **export SQL results** instead of **exporting a database**.

Exporting a Database

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export Database**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

- If you select **Export all tables**, data in an entire database or in specific tables will be exported.
- 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 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.
- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created. You can specify the storage path.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Exporting SQL Results

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export SQL Result**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- To export multiple SQL result sets at a time, enter SQL statements in the SQL text box. Enter each SQL statement on a separate line and add a semicolon (;) at the end. After the export task is complete, SQL files are generated. One SQL statement corresponds to one file.
- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Downloading Data Files

Data exported using the data export function is stored in the OBS bucket you created. You can download exported data files in any of the following ways:

- Download on the DAS Console.
- Download on the OBS management console.

Quick Export (Not Promoted)

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Quick Export** and select the database that you want to export data from.

NOTE

A maximum of 200,000 rows can be quickly exported from a single table. To export more data, choose **Create Task** > **Export Database**.

Step 8 On the **Quick Export** page, select a storage path and click **OK**.

Step 9 In the task list, view the export task you created.

In the row that contains the export task, you can click **Details** in the **Operation** column to view execution details of the task and information about exported tables.

----End

3.5 PostgreSQL

3.5.1 Account Management

3.5.1.1 Role Management

This section describes how to create one or more roles and grant different permissions to each role.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **Role Management**.

Step 7 On the role management page, click **Create Role** in the upper left corner.

Step 8 On the **Basic Settings** tab page, enter a role name (mandatory) and specify other parameters as required.

Step 9 (Optional) On the **Role Groups** tab page, select the check boxes in columns **Grant** and **With Grant Option** as needed.

Step 10 (Optional) On the **Permissions** tab page, click **Add**, specify a resource type, database, schema, and other column settings from the drop-down lists.

Step 11 Click **Save**. In the displayed **SQL Preview** dialog box, click **OK**.

Step 12 On the role management page, edit, rename, and delete existing roles as required.

----End

3.5.2 Database Management

3.5.2.1 Creating a Database

This section describes how to create a database on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list of the **Home** page, click **Create Database**.

Step 7 In the displayed dialog box, enter a database name, select a character set, and configure other parameters as required.

NOTE

You can execute the following SQL statement to query system table **pg_collation** and view the character set and its corresponding collation and character type as follows:

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

Step 8 Click **OK**. View the created database in the database list.

----End

3.5.2.2 Deleting a Database

This section describes how to drop a database that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list, locate the database you want to delete and click **Delete** in the **Operation** column.

Step 7 In the displayed dialog box, click **Yes**.

----End

3.5.3 SQL Operations

3.5.3.1 SQL Window

DAS allows you to run various SQL statements in the graphical SQL window, helping you easily manage databases.

Introduction to SQL Window

Table 3-32 Function description

Function	Description
Execute SQL	Executes SQL statements. SQL operations are performed on tables and views in a database.
Format SQL	Improves readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logics and semantics. SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.
Executing SQL Plan	Reports execution of SQL statements to make troubleshooting easy and optimize SQL processing performance.
SQL Favorites	Allows you to add, view, and manage frequently-used SQL statements.
Save Executed SQL Statements	Saves the recently executed SQL statements to the server.
SQL Input Prompt	Helps you quickly enter a specific database, table, or field name when you enter a SQL statement in the SQL window.
Full Screen	Shows SQL statements on a full screen.
Show Executed SQL Statements	Shows SQL execution details.
Messages	Shows the information returned after a SQL statement is executed.

Function	Description
Result Set	Shows SQL execution results. Allows you to view details about a single row, add a row, submit for editing, delete a row, export a row, copy a row, copy a column, set a column, and more.
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.

Opening the SQL Window

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

----End

3.5.3.2 Executing SQL Statements

This section describes how to execute various SQL statements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the upper left corner of the page, select a database and schema.

Step 8 Enter a SQL statement 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 when you enter a SQL statement in the SQL window.
- To protect query result sets from being displayed as garbled characters, select an encoding format other than the default encoding format, UTF-8.
- You can execute multiple SQL statements at a time. Separate them using semicolons (;). You can click **Full Screen** to view logics in SQL statements clearly.
  - To execute some of the SQL statements, select the statements before executing them.
  - To execute all SQL statements, do not select any SQL statements or select all SQL statements.

Step 9 View execution details of the current SQL statement and previously executed SQL statements in the lower part of the page.

Step 10 Click the **Messages** tab, view SQL execution details, including affected rows, progress, and time required.

Step 11 On the **Result Set** tab, view SQL execution results.

You can also perform the operations described in **Table 3-33** on result sets.

Table 3-33 Operations

Function	Description
Copy Row and Copy Column	Copies a row or column for reuse.
Column Settings	Customizes the display of columns when there are a number of columns in the query result.
Convert binary to hexadecimal	Converts binary data in the result set into hexadecimal data for display.
Refresh	Refreshes changed data.
Row Details	Displays the column field name, type, and data of the selected row.
Add Row	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.
Delete Row	Deletes the selected row, including data.
Export	Allows you to export data in a SQL or CSV file. A maximum of 10,000 rows of data can be exported.

Function	Description
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, data in the result set cannot be edited.
- 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, data in the result set cannot be edited.
- If only one table is in the result set and does not contain all primary key columns or any primary keys, you cannot edit the data.
- Data in virtual tables (for example, tables generated during execution of a stored procedure) cannot be edited.

----End

3.5.3.3 Executing SQL Plan

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the upper left corner of the page, select a database and schema.

Step 8 Enter a SQL statement in the SQL window and click **Execute SQL Plan**.

- On the **Executed SQL Statements** tab page, view SQL execution history.
- On the **Messages** tab page, view execution information, including SQL splitting, execution status, and elapsed time.
- On the **Execution Plan** tab page, view SQL execution details. Execution plan details of SQL statements are displayed on different tabs.

----End

3.5.3.4 SQL Favorites

This section describes how to add frequently-used SQL statements and view and manage them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 In the SQL window, click **SQL Favorites**.

- **Add SQL Statement**: allows you to add frequently-used SQL statements. Specifically, specify a title, select an application scope, enter SQL statements, and then click **Save**.
- **Insert SQL Statement**: allows you to view the SQL statements you have added. If there are no SQL statements stored in your account, this option will not be displayed.
- **Manage SQL Statements**: allows you to manage frequently-used SQL statements, including adding, editing, and deleting statements.

----End

3.5.3.5 SQL History

After you toggle on **Save Executed SQL Statements**, DAS will save the SQL statements you executed in the SQL window for future view. You can also execute the SQL statements again with no need to enter them again in the SQL window.

Prerequisites

You have enabled **Save Executed SQL Statements** in the upper right corner of the **SQL Query** page. After this function is enabled, executed SQL statements will be saved on the DAS management host.

Constraints on Usage

SQL statement execution records can be stored for up to one year.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL History**.

Step 7 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 a SQL statement, click it in the **SQL Statement** column.
- To execute a SQL statement, click **Open in SQL window** in the **Operation** column.

----End

3.5.4 Table Management

3.5.4.1 Opening a Table

The graphical window for creating a PostgreSQL database table is not available. To create a table, execute SQL statements in the SQL window.

After a table is created, you can open the table on the DAS console and view data in the table.

Usage Notes

Tables in PostgreSQL databases do not have primary keys and their data can only be queried.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left. In the table list, locate the table you want to open and click **Open** in the **Operation** column.

Step 8 On the table details page, specify the rows to be displayed.

You can also copy and reuse rows and columns.

Table 3-34 UI operations

Name	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.
Refresh	Allows you to update table data manually.
Row Details	Shows details of a specified row.

----End

3.5.4.2 Viewing Details of a Table

This section describes how to view basic information of a table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left. In the right pane, locate the table whose details you want to view and click **View** in the **Operation** column.

Step 8 In the displayed dialog box, view basic table information.

----End

3.5.5 View Management

3.5.5.1 Opening a View

Views in PostgreSQL databases do not have primary keys and their data can only be queried.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** page, choose **Views**. In the right pane, locate the view that you want to open and click **Open** in the **Operation** column.

Step 8 Check view information on the details page or click **Where Condition**.

Table 3-35 UI operations

Name	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.
Refresh	Allows you to update table data manually.
Row Details	Shows details of a specified row.

----End

3.5.5.2 Viewing Details of a View

This section describes how to view details of a view, which show the SQL statement for creating it.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Views** on the left. In the right pane, locate the view whose details you want to view and click **View Details** in the **Operation** column.

Step 8 In the displayed dialog box, view the SQL statement used to create the view.

----End

3.5.6 Stored Procedure Management

3.5.6.1 Viewing Details of a Stored Procedure

This section describes how to view details of a stored procedure on the DAS console. The details show the SQL statement for creating this stored procedure.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the **Objects** tab page, choose **Stored Procedures**.

Step 8 Locate the stored procedure whose details you want to view and click **View Details** in the **Operation** column.

Step 9 In the displayed dialog box, view the SQL statement used to create the stored procedure.

----End

3.5.7 Trigger Management

3.5.7.1 Viewing Details of a Trigger

This section describes how to view details of a trigger on the DAS console. The details show the SQL statement for creating the trigger.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the **Objects** tab page, choose **Triggers**.

Step 8 Locate the trigger whose details you want to view and click **View Details** to view the SQL statement used to create the trigger.

----End

3.5.8 Sequence Management

3.5.8.1 Viewing Details of a Sequence

This section describes how to view details of a sequence, which show the SQL statement for creating it.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the **Objects** tab page, choose **Sequences**.

Step 8 Locate the sequence whose details you want to view and click **View Details** to view the SQL statement used to create the sequence.

----End

3.5.9 Data Import and Export

3.5.9.1 Importing Data

This section describes how to import data from your local PC or an OBS bucket for data backup and migration.

Usage Notes

- 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.
- Only one file that is no larger than 1 GB can be imported at a time.
- Only data files in the CSV or SQL format can be imported. If the number of instance tables exceeds 100,000, the CSV format cannot be used.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- This function is not available to PostgreSQL Enhanced Edition instances.
- The size of a single SQL statement to be imported must be less than 100 MB.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Import**.

Step 7 Click **Create Task**.

Table 3-36 Parameter description

Parameter	Description
Import Type	Set **Import Type** to **sql** or **CSV**.

Parameter	Description
File Source	Import a file from your local PC or an OBS bucket. • Upload file If you select **Upload file** for **File Source**, you need to set **Attachment Storage** and upload the required file. To keep your data secure, provide your own OBS bucket to store the file you uploaded. In this way, DAS automatically connects to your OBS bucket for in-memory reading. No data is stored on DAS. Creating OBS buckets is free of charge, but saving files will incur certain costs. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. • Choose from OBS If you select **Choose from OBS** for **File Source**, you need to select a file from the bucket. The file uploaded from an OBS bucket will not be deleted upon an import success.
Database	Select the database that you want to import the file to.
Charset	Select a charset as needed.
Options	If you select **Ignore errors, that is, skip the step where the SQL statement fails to be executed**, the system will skip any errors detected when SQL statements are being executed. If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. This option is only available to the files uploaded from your local PC.
Remarks	Enter remarks as required.

Step 8 After setting import parameters, click **Create**.

Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 9 On the displayed page, view import progress in the task list.

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

----End

3.5.9.2 Exporting Data

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.

DAS allows you to export an entire database, some data tables, or result sets of SQL statements.

Usage Notes

- If you do not select **Generate a file for each table** when exporting data, the exported data file is in **.zip** format. Data files in this format cannot be directly imported. You need to decompress the file before importing it again.
- If **Generate a file for each table** is selected during data export, the exported data file is in **.sql** or **.csv** format. In this case, the exported data file can be directly imported again.
- If the **Exporting a Database** function is used to export over 100,000, PostgreSQL instance tables, an error message will be displayed indicating that the number of tables is too large and data cannot be exported. In this case, use the **Exporting SQL Results** function instead.
- This function is not available to PostgreSQL Enhanced Edition instances.

Exporting a Database

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export Database**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

NOTE

Functions listed in **Table 3-37** are automatically created by a PostgreSQL instance. When importing data on DAS, you need to manually delete the functions.

Table 3-37 Functions

1	public.create_plugin_pg_cron
2	public.drop_plugin_pg_cron
3	public.create_plugin_pg_repack
4	public.drop_plugin_pg_repack
5	public.create_plugin_dblink

6	public.drop_plugin_dblink
7	public.select_control_version
8	public.create_plugin_postgres_fdw
9	public.drop_plugin_postgres_fdw
10	public.create_plugin_pg_repack
11	public.drop_plugin_pg_repack
12	public.create_plugin_pg_stat_statements
13	public.drop_plugin_pg_stat_statements
14	public.create_plugin_pg_cron
15	public.drop_plugin_pg_cron
16	public.control_extension
17	public.control_tablespace

- If you select **Export all tables**, data in an entire database or in specific tables will be exported.
- 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 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.
- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created. You can specify the storage path.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Exporting SQL Results

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Create Task** and choose **Export SQL Result**.

Step 8 In the displayed dialog box, configure basic information and advanced settings as required.

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- To export multiple SQL result sets at a time, enter SQL statements in the SQL text box. Enter each SQL statement on a separate line and add a semicolon (;) at the end. After the export task is complete, SQL files are generated. One SQL statement corresponds to one file.
- DAS does not store any user data. The exported data files are stored in the OBS bucket that you have created.
- Creating OBS buckets is free of charge, but saving files will incur certain costs.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, locate the created task and view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

----End

Downloading Data Files

Data exported using the data export function is stored in the OBS bucket you created. You can download exported data files in any of the following ways:

- Download on the DAS Console.
- Download on the OBS management console.

Quick Export (Not Promoted)

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 In the upper left corner of the page, click **Quick Export** and select the database that you want to export data from.

NOTE

A maximum of 200,000 rows can be quickly exported from a single table. To export more data, choose **Create Task** > **Export Database**.

Step 8 On the **Quick Export** page, select a storage path and click **OK**.

Step 9 In the task list, view the export task you created.

In the row that contains the export task, you can click **Details** in the **Operation** column to view execution details of the task and information about exported tables.

----End

3.5.10 Intelligent O&M

3.5.10.1 Dashboard

Dashboard helps you get knowledge of overall information about your instances, including instances by status, engine distribution, active alarms, slow query logs, monitoring dashboard, and resource risky instances. Intelligent Diagnosis checks instance health using instance operation data and intelligent algorithms and provides diagnosis results and suggestions.

Viewing Instances by Status

You can view the number of instances by status in the current region, including primary instances and read replicas.

Viewing Engine Distribution

You can view the distribution of engines in the current region, including engine types and abnormal engines.

Querying Active Alarms

You can view active alarms of all instances in the current region, including alarms in the Alarm (metric) and Triggered (event) statuses.

Viewing Slow Query Logs

You can view the number of slow query logs of all instances in the current region.

Monitoring Dashboards

You can view metrics of all instances in the current region.

Viewing Resource Risky Instances

You can view metric risks of all instances in the current region.

3.5.10.2 SQL

3.5.10.2.1 Slow Query Logs

This section describes how to analyze slow query logs in a specified period. Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Viewing Slow Query Logs

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Slow Query Logs**.

Step 8 Select a time range and view trends, statistics, and details of the slow query logs generated within the time range.

- Log trends

  In the trends graph, you can hover your mouse at any point in time to view slow query logs.

- Log details

  View slow query log details in the lower part of the **Slow Query Logs** area. The details include the SQL statement, execution time, database name, client, user, execution duration, lock wait duration, and scanned and returned rows.

  Click **Export** to export slow query log details to a specific OBS bucket. After the details are exported, you can click **Export History** to view export records. You can also download the details to your local PC.

- Slow query log statistics

  Click **Sample** in the **Operation** column to view the sample of the SQL statement template.

  On the **Statistics** page, click **Export** to export slow query logs to a specific OBS bucket. Then you can click **Export History** to view export records. You can also download the exported slow query logs to your local PC.

----End

Slow Query Log Storage

After **Collect Slow Query Logs** is enabled, SQL text content will be stored in OBS for analysis.

3.5.10.2.2 SQL Explorer

After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.

SQL Explorer can record all SQL statements executed on your instance and enables you to analyze and search for the tables that are accessed and updated the most frequently and the SQL statements that have the longest lock wait.

Usage Notes

- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected and the collected SQL statements will be deleted.
- SQL Explorer cannot record all data. As for RDS for PostgreSQL, it has the following constraints:

  SQL Explorer is limited by the memory buffer, so some data may not be recorded when a buffer overrun occurs, for example, when the QPS is over 1000 or the usage of resources such as CPUs or memory becomes too high.

  Any SQL statement that exceeds 4,096 bytes by default will be discarded.

  SQL Explorer is available to only RDS for PostgreSQL instances of the following versions:

  - Major version: 14; minor version: 14.4 or later
  - Major version: 13; minor version: 13.6 or later (if the minor version is 13.2 or later but earlier than 13.6, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)
  - Major version: 12; minor version: 12.10 or later (if the minor version is 12.6 or later but earlier than 12.10, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)
  - Major version: 11; minor version: 11.15 or later (if the minor version is 11.11 or later but earlier than 11.15, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)
  - Major version: 10; minor version: 10.20 or later (if the minor version is 10.16 or later but earlier than 10.20, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)
  - Major version: 9.6; minor version: 9.6.24 or later (if the minor version is 9.6.21 or later but earlier than 9.6.24, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)
  - Major version: 9.5; minor version: 9.6.25 or later (for some instances with the minor version of 9.5.25, SQL statements executed using psql can be collected, but those executed by other drivers like JDBC may not.)

Execution Duration Distribution of Top SQL Templates

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Explorer**.

Step 8 Toggle on **Collect All SQL Statements** to obtain information about all SQL statements executed on the current instance.

NOTE

Collect All SQL Statements takes effect once it is enabled.

Step 9 On the **Top SQL** tab page, view execution duration of top SQL statements.

You can select **Last 1 hour**, **Last 3 hours**, or **Last 6 hours** based on service requirements to view the distribution. Or customize a time range that is no longer than one day.

The Execution Duration Distribution graph shows how many times SQL statements with different elapsed time lengths are executed over a specified time range. DAS collects SQL statistics of four elapsed time lengths.

- < 100 ms: number of SQL statement executions whose elapsed time is less than 100 ms
- 100 ms–500 ms: number of SQL statement executions whose elapsed time ranges from 100 ms to 500 ms
- 500 ms–1s: number of SQL statement executions whose elapsed time ranges from 500 ms to 1s
- > 1s: number of SQL statement executions whose elapsed time is longer than 1s

SQL statistics in the four elapsed time lengths are collected over the time range you specified.

- If you select **Last hour**, SQL statement executions are calculated every 10 seconds.
- If you select **Last 6 hours**, SQL statement executions are calculated every minute.
- If you select a time range longer than 6 hours, SQL statement executions are calculated every 5 minutes.

Select **By instance** or **By node** to view execution duration distribution. You can change node to view its execution duration distribution. The primary node is

selected by default. If you want to view the execution duration distribution of other nodes, click **Change Node**.

Step 10 Use either of the following methods 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 statements at that time.
- Specify a time range using your mouse on the graph, and you will see top SQL statements during that time range.
- Click **Export** to export details about all top SQL templates in the list. This function is available only to paid instances.
- 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.

----End

SQL Audit

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 On the **Overview** page, click **Go to Intelligent O&M**.

Step 5 Locate the target instance and click **Details**.

Step 6 Choose **SQL** > **SQL Explorer**.

Step 7 Click the **SQL Audit** tab.

Step 8 On the **SQL Audit** tab page, view SQL audit tasks.

- Specify the start time, end time, task creation time, and node information to search for SQL audit tasks.
- Click **Add SQL Audit Task** and specify a time range to add a task. You can add an instance- or node-level task based on your service requirements. The added SQL audit task is displayed in the list below.
- Click **Details** in the **Operation** column to view task details. You can specify filters such as **Time Range**, **User**, **Keyword**, or **Database** and click **Query** to search for the SQL statements executed on the current instance. The selected time range must be after the time when the SQL audit task is added.

----End

3.5.10.2.3 SQL Diagnosis

This function enables you to diagnose SQL statements and obtain optimization suggestions.

Usage Notes

- Only SELECT, INSERT, UPDATE, and DELETE statements can be diagnosed. An INSERT statement must contain a SELECT clause.
- SQL diagnosis history is stored on the DAS server only. If you delete SQL diagnosis history on the console, it will be also deleted from the DAS server.
- Well-formatted SQL statements are much easier to read. Only line breaks will be added, and there are no logic and semantic changes.
- All SQL statements in the SQL window will be formatted. You cannot format only one of them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP address.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Diagnosis**. On the displayed dialog box, perform the following operations:

- Diagnosing SQL Statements

  On the **SQL Diagnosis** tab page, select the database where you want to diagnose a SQL statement, enter a SQL statement, and click **Diagnose**. The diagnosis result page is displayed. You can optimize the SQL statement based on the diagnosis result.
- Executing SQL Statements

  Select the target database, enter a SQL statement, and click **Execute**. The system executes the statement and displays a result set.
- Formatting SQL Statements

  Select the target database, enter SQL statements, and click **Format**. The system formats all the SQL statements.
- Viewing the Execution Plan

  Select the target database, enter SQL statements, and click **View Execution Plan**. The system executes all the SQL statements in sequence.

----End

3.6 DDS

3.6.1 Account Management

3.6.1.1 User Management

Multiple users with different permissions can be created to access a DB instance or database, but the permissions of these users must be within a certain range.

Creating a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 Select the database for which a user is to be created and click **Create User**.

Step 8 On the displayed page, enter the username and password and confirm the password.

Step 9 In the lower part of the page, click **Add Role**. On the displayed page, select roles for the database.

NOTE

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

Step 10 After setting required parameters, click **Save**. In the preview dialog box, click **OK**.

----End

Editing a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user that you want to edit and click **Edit** in the **Operation** column.

Step 8 Edit user information, such as the password, associated role, and user-defined data.

Step 9 Click **Save**.

----End

Deleting a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user you want to delete and click **Delete**.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.6.1.2 Role Management

This section describes how to create one or more roles and grant different permissions to each role.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **Role Management**.

Step 7 Select the database for which the roles are to be created and click **Add Roles**.

Step 8 Enter a role name.

Step 9 On the **Permission** tab page, click **Add Privileges**. In the displayed dialog box, specify required parameters and click **Add**.

Step 10 (Optional) On the **Role** tab page, click **Add Roles**. On the displayed page, select the roles you want to add and click **Add**.

Step 11 After setting the roles, click **Save**.

NOTE

If a role is created for the admin database, you can select other databases when adding permissions 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

3.6.2 Database Management

3.6.2.1 Creating a Database

This section describes how to create a database on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list of the **Home** page, click **Create Database**.

Step 7 On the displayed page, enter a database name and collection name (optional).

Step 8 Click **OK**. View the created database in the database list.

----End

3.6.2.2 Dropping a Database

This section describes how to drop a database that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 In the database list, locate the database you want to delete and click **Drop** in the **Operation** column.

Step 7 In the displayed dialog box, click **Yes**.

----End

3.6.3 Command Operation

3.6.3.1 Command Query

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

Precautions

- The timestampe() command is not supported.
- 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
          }
       }
   ]
});
```

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Command Operation** > **Command Window**.

Step 7 In the displayed text box, enter the commands you want to execute and click **Execute Command**.

You can also perform the operations listed in **Table 3-38** on the command query page.

Table 3-38 Function description

Function	Description
Executing Command	Executes commands. Commands are performed on collections and views in a database.
Full Screen	Shows commands on a full screen.
Save Executed Commands	After this function is enabled, the system saves the recently executed commands to the server.
Executed Commands	Shows the command execution details.
Messages	Shows the command output.
Result Set	Shows the command execution result.
Overwrite/ Append Mode	• **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.
Export of the query result set	Used to export the query result set. A maximum of 1000 records can be exported at a time.

----End

3.6.3.2 Command History

After you toggle on **Save Executed Commands**, DAS will save the commands you executed in the SQL window for future view. You can also execute the commands again with no need to enter them again in the SQL window.

Prerequisites

You have enabled **Save Executed Commands**.

After you enable **Save Executed Commands**, executed commands will be recorded in the DAS management host.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Command Operation** > **Command History**.

Step 7 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

3.6.4 Collection Management

3.6.4.1 Creating a Collection

This section describes how to create a collection on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the **Collections** tab page, click **New Collection**.

Step 8 On the displayed page, enter a collection name. Select **Capped collection**, **Auto index_id**, 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 9 Enter a 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 10 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

3.6.4.2 Opening a Collection

This section describes how to view details about documents in a collection and create or delete documents on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab page, choose **Collections**, locate the target collection, and click **Open Collection** in the **Operation** column.

Step 8 On the displayed page, you can perform the following operations:

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

Table 3-39 Parameter description

Parameter	Description
Query	Uses a query operator to specify the filter.
Projection	Specifies fields that matched with the query filter in the document to return.
Sort	Defines the sorting rule.
Limit	Specifies the maximum number of documents that can be returned.
Skip	Specifies the number of skipped documents.

- On the **JSON** tab page, view filtered documents.
- To facilitate viewing, DAS allows you to view the documents in Table and Tree mode. You can view text files on the **Tree** and **Table** tab pages.
- You can create, delete, and view documents.

----End

3.6.4.3 Altering Validator

This section describes how to alter validation rules of a collection on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Collections**, locate your target collection and click **Alter Validator** in the **Operation** column.

Step 8 On the displayed **Alter Validator** page, confirm the validation rule, and set the validation level and action.

Step 9 Click **OK**.

----End

3.6.4.4 Renaming a Collection

This section describes how to rename a collection on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** page, select **Collections**, locate the collection that you want to rename, and click **Rename** in the **Operation** column.

Step 8 On the displayed page, set a new name for the collection and click **OK**.

----End

3.6.4.5 Clearing a Collection

This section describes how to clear data in a collection.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Collections**, locate the collection you want to clear, and click **Clear** in the **Operation** column.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.6.4.6 Dropping a Collection

This section describes how to drop a collection that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Collections**, locate the collection you want to drop and click **Drop** in the **Operation** column.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.6.4.7 Creating an Index

Indexes can greatly improve query efficiency. If there are no indexes, each file in the collection must be scanned and the records that meet the query conditions must be selected when data is read. If the amount of data is large, querying an entire collection may take dozens of seconds or even minutes.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed page, click + before the collection name.

Step 8 Click **New Index**.

Step 9 On the displayed page, configure basic parameters.

Table 3-40 Parameter description

Parameter	Description
Database	Selected database by default. You can also enter a database name. After the index is created, the system automatically creates a database with the name you entered.
Collection	Selected collection by default. You can also enter a collection name. After the index is created, the system automatically creates a collection with the name you entered.
Index	Index name, which is user-defined.
Advanced Settings	• **Background**: indicates the backend mode for creating indexes. Selecting **Background** is recommended. If this option is not selected, the database will be locked. • **Unique**: specifies whether the created index is unique. • **Sparse**: does not enable indexes for field data that does not exist in the document.
Index	Click **Add** and enter a name and type to add an index.

Step 10 (Optional) on the **Text Option** page, select information such as the index version and default language.

Step 11 (Optional) on the **Collation** page, set the sorting rule.

Step 12 After the configurations are complete, click **Save**.

----End

3.6.4.8 Editing an Index

This section describes how to alter the basic information, text options, and collation of an index on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Choose **Collections** in the navigation pane. In the index list, locate the index you want to edit and click + before the index name.

Step 8 Locate the index that you want to edit and click **Edit Index** in the **Operation** column.

Step 9 Click **Save**.

----End

3.6.4.9 Deleting an Index

This section describes how to drop an index that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 Choose **Collections** in the navigation pane. In the index list, locate the index you want to delete and click + before the index name.

Step 8 Locate the index that you want to delete and click **Delete** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.6.5 View Management

3.6.5.1 Creating a View

A view is a queryable object whose contents are defined by aggregation pipelines on other collections or views. This section describes how to create a view.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Views** and click **Create View**.

Step 8 On the displayed page, enter the view name and collection name.

Step 9 Click **Add Pipeline Stage**. In the displayed dialog box, select an operator and enter an expression.

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

Step 10 On the **Collation** tab, click **Custom Collation** and set collation rules as needed.

Step 11 After setting the view information, click **Save**.

----End

3.6.5.2 Opening a View

This section describes how to view details of a view on the DAS console.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** page, select **Views**, locate the view that you want to open and click **Open** in the **Operation** column.

Step 8 On the displayed page, you can perform the following operations:

- Set parameters **Query**, **Projection**, **Sort**, **Limit**, and **Skip**.
- On the **JSON** tab page, view filtered documents.
- DAS allows you to view text files in table and tree mode on the **Tree** and **Table** tab pages.

----End

3.6.5.3 Editing a View

This section describes how to alter collections, aggregation pipeline information, and validation rules that are associated with a collection.

Usage Notes

Improper alterations on a view will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Views**, locate the view you want to edit and click **Alter** in the **Operation** column.

Step 8 Altering view information

Step 9 After editing the view information, click **Save**.

----End

3.6.5.4 Dropping a View

This section describes how to drop a view that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Views**, locate the view you want to drop and click **Drop** in the **Operation** column.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.6.6 Intelligent O&M

3.6.6.1 Querying the Number of Connections to an Instance

DAS records client IP addresses and the connections corresponding to each of them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 On the **Connections** page, view connection details of the current instance, including the total connections, internal connections, and external connections.

You can also switch nodes to view the connections of a specific node.

----End

3.6.6.2 Sessions

You can manage sessions in the following scenarios:

- If the number of sessions of an instance has reached the upper limit and the instance cannot be logged in to any longer, you can view and kill unnecessary sessions using the emergency channel function.
- You can view history logs to learn details of the kill operations that you performed using the emergency channel function.

CAUTION

Killing a session may cause instance or service exceptions.

Usage Notes

- If it can be logged in to through DAS, log in to your instance on the DAS console and enter the commands to kill the sessions in question from there.
- Use this function in urgent conditions. All your kill operations will be logged.
- Only Community Edition 3.4 and 4.0 instances are supported.
- Instances in the creating, frozen or abnormal state are not supported.
- Killing inactive sessions is not allowed.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 Choose **Sessions**.

Step 6 On the displayed **Emergency Channel** page, view session statistics of the current instance node by overview, host, or namespace.

Step 7 In the session list in the lower part of the page, sessions are sorted by running time in descending order by default. Search for sessions by setting a running time threshold or namespace.

Step 8 Select the session that you want to kill and click **Kill Session**.

Step 9 In the **Kill Session** dialog box, confirm the session information and click **Yes**.

Step 10 Click **History Logs** to view the sessions killed through the emergency channel.

----End

3.6.6.3 Slow Query Logs

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, client IP address, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Slow Query Logs**.

You can click to enable **Collect Slow Query Logs**.

You can also click **Log Settings** in the upper right corner of the page to enable **Collect Slow Query Logs**.

On the **SQL Query Logs** tab page, you can select a time range (last hour, last 3 hours, last 12 hours, or a user-defined time range which must be within three days) to view slow queries in the time range. You can view slow query logs over time, as well as slow query log statistics and details.

- The **Slow Queries over Time** chart displays slow query logs in the specified time period.
- **Statistics** shows slow query logs by node, SQL statement type, database, data table, user, and client.
- Slow query log details include the execution completion time, SQL statement, SQL statement type, database, database table, client IP address, user, execution duration, lock wait time, scanned documents, returned documents, and scanned indexes.

Above the chart, you can switch to another instance or node to view slow query logs by instance or by node.

----End

3.7 DDM

3.7.1 Account Management

3.7.1.1 User Management

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 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 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.

Table 3-41 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.

Step 8 Ensure that the settings are correct and click **Save**.

----End

Editing a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user whose information you want to edit and click **Edit** in the **Operation** column.

Step 8 On the displayed page, edit basic information, basic permissions and extended permissions, and click **Save**.

----End

Deleting a User

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Account Management** > **User Management**.

Step 7 In the user list, locate the user that you want to delete and click **Delete** in the **Operation** column.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.7.2 Database Management

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

3.7.3 SQL Operations

3.7.3.1 SQL Window

DAS allows you to run various SQL statements in graphical SQL windows, helping you easily manage databases.

Introduction to SQL Window

Table 3-42 Function description

Function	Description
Execute SQL	Executes SQL statements.

Function	Description
Format SQL	Improves readability of SQL statements. Formatting SQL statements enables statements to be displayed in line break mode, but does not change their logic and semantics. SQL formatting takes effect for all the SQL statements in the SQL window. You cannot format only one selected statement.
Executing SQL Plan	Reports execution of SQL statements to make troubleshooting easy and optimize SQL processing performance.
SQL Favorites	Allows you to add, view, and manage frequently-used SQL statements.
Save Executed SQL Statements	After this function is enabled, the system saves the recently executed SQL statements to the server.
SQL Input Prompt	Helps you quickly enter a specific database, table, or field name when you enter a SQL statement in the SQL window.
Full Screen	Shows SQL statements on a full screen.
SQL History	Shows the SQL execution details.
Messages	Shows the information returned after a SQL statement is executed.
Result Set	Shows SQL execution results. Allows you to view details about a single row, copy a row, copy a column, set a column, and more.
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.

Opening the SQL Window

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

----End

3.7.3.2 Executing SQL Statements

This section describes how to execute various SQL statements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Select a database from the navigation tree on the left.

Step 8 Enter a SQL statement 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 when you enter a SQL statement in the SQL window.
- You can execute multiple SQL statements at a time. Separate them using semicolons (;). You can click **Full Screen** to view logics in SQL statements clearly.
  - To execute some of the SQL statements, select the statements before executing them.
  - To execute all SQL statements, do not select any SQL statements or select all SQL statements.

Step 9 View execution details of the current SQL statement and previously executed SQL statements in the lower part of the page.

Step 10 Click the **Messages** tab, view SQL execution details, including affected rows, progress, and time required.

Figure 3-2 Viewing messages

Step 11 On the **Result Set** tab, view SQL execution results. You can perform the following operations on the result sets:

- **Copy Row**/**Copy Column**: Copy a row or column for reuse.
- **Column Settings**: Customize columns to be displayed when there are a large number of columns in the query result.
- **Refresh**: Refresh changed data.
- **Row Details**: View the column field name, value, and type of the selected row.
- **Export**: Export the execution result of SQL statements.

----End

3.7.3.3 Executing SQL Plan

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Select a database from the navigation tree on the left.

Step 8 Enter a SQL statement in the SQL window and click **Execute SQL Plan**.

Figure 3-3 Executing SQL plan

- On the **Executed SQL Statements** tab page, view SQL execution history.
- On the **Messages** tab page, view the execution information, including SQL splitting, execution status, and elapsed time.
- On the **Execution Plan** tab page, view SQL execution details. Execution plan details of SQL statements are displayed on different tabs.

----End

3.7.3.4 SQL Favorites

This section describes how to add frequently-used SQL statements and view and manage them.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

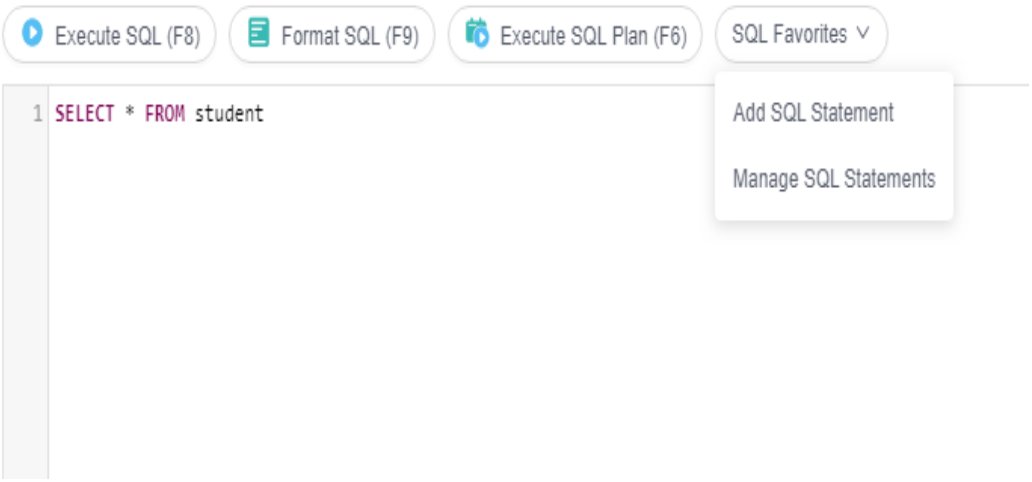
Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

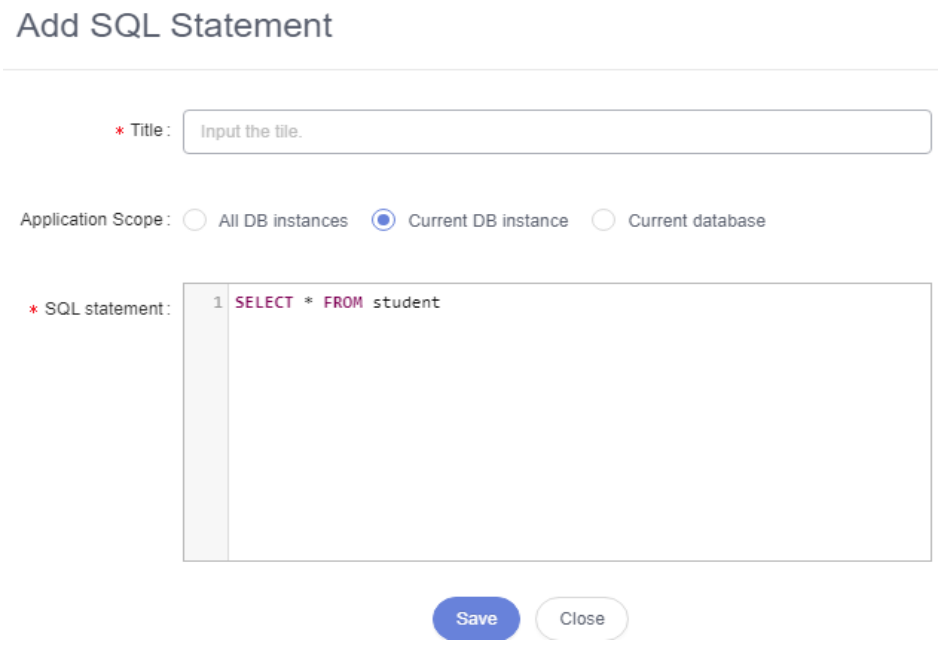
Step 7 In the SQL window, click **SQL Favorites**.

Figure 3-4 SQL favorites



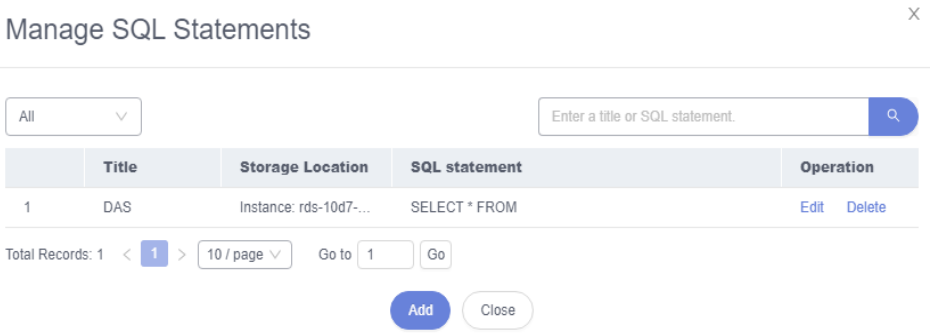
- **Add SQL Statement**: allows you to add frequently-used SQL statements. Specifically, specify a title, select an application scope, enter SQL statements, and then click **Save**.

  Figure 3-5 Adding a SQL statement

- **Insert SQL Statement**: allows you to view the SQL statements you have added. If there are no SQL statements stored in your account, this option will not be displayed.
- **Manage SQL Statements**: allows you to manage frequently used SQL statements, including adding, editing, and deleting statements.

Figure 3-6 Managing SQL statements



----End

3.7.3.5 SQL History

After you toggle on **Save Executed SQL Statements**, DAS will save the SQL statements you executed in the SQL window for future view. You can also execute the SQL statements again with no need to enter them again in the SQL window.

Prerequisites

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

Constraints on Usage

SQL statement execution records can be stored for up to one year.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL History**.

Step 7 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 a SQL statement, click it in the **SQL Statement** column.
- To execute a SQL statement, click **Open in SQL window** in the **Operation** column.

----End

3.7.4 Table Management

3.7.4.1 Creating a Table

A data table consists of basic information, fields, and indexes. As the supplement to table information, indexes are optional and can be configured based on service requirements.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** page, select **Tables**. In the upper left corner of the page, click **Create Table**.

Step 8 On the displayed page, specify required parameters.

Step 9 Click **Next**.

Step 10 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, click **Create**.
- If you need to add indexes, click **Next** until all your desired parameters are specified. After the setting is complete, click **Create**.

Step 11 In the **SQL Preview** dialog box, click **Execute**.

----End

3.7.4.2 Opening a Table

This section describes how to open a table on the DAS console and view data in the table.

Precautions

Currently, tables in a DDM database cannot be edited.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left. In the table list, locate the table you want to open, and click **Open** in the **Operation** column.

Step 8 On the table details page, view the table information.

You can also perform the operations listed in **Table 3-43**.

Table 3-43 UI operations

Name	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.
Convert 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 details of a specified row.

----End

3.7.4.3 Viewing Details of a Table

This section describes how to view table information and the SQL statement for creating this table.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left. In the table list, locate the table whose details you want to view and click **View** in the **Operation** column.

Step 8 In the displayed dialog box, view basic information of the table and the SQL statement for creating the database.

----End

3.7.4.4 Altering a Table

This section describes how to alter a table on the DAS console.

Usage Notes

Improper alterations on a table will cause instance or service exceptions.

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left.

Step 8 In the right pane, locate the table you want to alter and click **Alter** in the **Operation** column.

Step 9 On the displayed tab, modify the parameters as required and click **Save**.

Step 10 After the alteration is complete, click **Alter**.

Step 11 In the **SQL Preview** dialog box, click **Execute**.

----End

3.7.4.5 Clearing a Table

This section describes how to clear data in a table.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left.

Step 8 In the right pane, locate the table you want to clear and choose **More** > **Clear** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.7.4.6 Dropping a Table

This section describes how to drop a table that is no longer needed.

Precautions

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

Procedure

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane on the left, choose **Development Tool**.

Step 5 Locate the DB instance that you want to log in to and click **Log In** in the **Operation** column.

Step 6 On the top menu bar, choose **Database Management**.

Step 7 On the displayed **Objects** tab, choose **Tables** on the left.

Step 8 In the right pane, locate the table you want to drop and choose **More** > **Drop** in the **Operation** column.

Step 9 In the displayed dialog box, click **Yes**.

----End

3.7.5 Intelligent O&M

3.7.5.1 Dashboard

Dashboard helps you get knowledge of overall information about your instances, including instances by status, engine distribution, active alarms, slow query logs, monitoring dashboard, and resource risky instances. Intelligent Diagnosis checks instance health using instance operation data and intelligent algorithms and provides diagnosis results and suggestions.

Viewing Instances by Status

You can view the number of instances by status in the current region, including primary instances and read replicas.

Viewing Engine Distribution

You can view the distribution of engines in the current region, including engine types and abnormal engines.

Querying Active Alarms

You can view active alarms of all instances in the current region, including alarms in the Alarm (metric) and Triggered (event) statuses.

Viewing Slow Query Logs

You can view the number of slow query logs of all instances in the current region.

Monitoring Dashboards

You can view metrics of all instances in the current region.

Viewing Resource Risky Instances

You can view metric risks of all instances in the current region.

3.7.5.2 SQL

3.7.5.2.1 Slow Query Logs

This section describes how to analyze slow query logs in a specified period. Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Viewing Slow Query Logs

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **Slow Query Logs**.

Step 8 Select a time range and view trends, statistics, and details of the slow query logs generated within the time range.

- Log trends

  In the trends graph, you can hover your mouse at any point in time to view slow query logs.

- Log details

  View slow query log details in the lower part of the **Slow Query Logs** area. The details include the SQL statement, execution time, database name, client, user, execution duration, lock wait duration, and scanned and returned rows.

  Click **Export** to export slow query log details to a specific OBS bucket. After the details are exported, you can click **Export History** to view export records. You can also download the details to your local PC.

- Slow query log statistics

  Click **Sample** in the **Operation** column to view the sample of the SQL statement template.

  On the **Statistics** page, click **Export** to export slow query logs to a specific OBS bucket. Then you can click **Export History** to view export records. You can also download the exported slow query logs to your local PC.

----End

Slow Query Log Storage

After **Collect Slow Query Logs** is enabled, SQL text content will be stored in OBS for analysis.

Deleting Slow Query Logs

On the **Slow Query Logs** tab page, you can click **Delete All** or **Delete** to delete all slow query logs or some slow query logs generated in a specific period.

NOTE

Deleted slow query logs cannot be recovered. Exercise caution when performing this operation.

3.7.5.2.2 SQL Explorer

After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.

SQL Explorer can record all SQL statements executed on your instance and enables you to analyze and search for the tables that are accessed and updated the most frequently and the SQL statements that have the longest lock wait.

Usage Notes

- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected and the collected SQL statements will be deleted.

Execution Duration Distribution of Top SQL Templates

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 In the navigation pane, choose **Intelligent O&M** > **Instance List**.

Step 5 In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP.

Step 6 Locate the target instance and click **Details**.

Step 7 Choose **SQL** > **SQL Explorer**.

Step 8 Toggle on **Collect All SQL Statements** to obtain information about all SQL statements executed on the current instance.

NOTE

Collect All SQL Statements takes effect once it is enabled.

Step 9 On the **Top SQL** tab page, view execution duration of top SQL statements.

You can select **Last hour**, **Last 3 hours**, or **Last 6 hours** to view execution durations of top SQL statements. You can also set a time range that spans no more than one day.

The Execution Duration Distribution graph shows how many times SQL statements with different elapsed time lengths are executed over a specified time range. DAS collects SQL statistics of four elapsed time lengths.

- < 100 ms: number of SQL statement executions whose elapsed time is less than 100 ms
- 100 ms–500 ms: number of SQL statement executions whose elapsed time ranges from 100 ms to 500 ms
- 500 ms–1s: number of SQL statement executions whose elapsed time ranges from 500 ms to 1s

- > 1s: number of SQL statement executions whose elapsed time is longer than 1s

SQL statistics in the four elapsed time lengths are collected over the time range you specified.

- If you select **Last hour**, SQL statement executions are calculated every 10 seconds.
- If you select **Last 6 hours**, SQL statement executions are calculated every minute.
- If you select a time range longer than 6 hours, SQL statement executions are calculated every 5 minutes.

Step 10 Use either of the following methods 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 statements at that time.
- Specify a time range using your mouse on the graph, and you will see top SQL statements during that time range.
- Click **Export** to export details about all top SQL templates in the list.
- 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.

----End

SQL Audit

Step 1 Log in to the DAS 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 **Databases**, click **Data Admin Service**.

Step 4 On the **Overview** page, click **Go to Intelligent O&M**.

Step 5 Locate the target instance and click **Details**.

Step 6 Choose **SQL** > **SQL Explorer**.

Step 7 Click the **SQL Audit** tab.

Step 8 On the **SQL Audit** tab page, view SQL audit tasks.

- Specify the start time, end time, task creation time, and node information to search for SQL audit tasks.
- Click **Add SQL Audit Task** and specify a time range to add a task. You can add an instance- or node-level task based on your service requirements. The added SQL audit task is displayed in the list below.
- Click **Details** in the **Operation** column to view task details. You can specify filters such as **Time Range**, **User**, **Keyword**, or **Database** and click **Query** to

search for the SQL statements executed on the current instance. The selected time range must be after the time when the SQL audit task is added.

----End

3.8 Audit

3.8.1 DAS Key Operations Supported by CTS

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

Table 3-44 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	batchDeleteConnections
Logging in to a database instance	connection	databaseLoginActions
Logging in to a database instance	connection	dbsConnection
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
Editing result sets	connection	editDatas
Exporting result sets	connection	exportData
Closing the SQL window	connection	closeWindow
Obtaining autofill data	connection	autoFill

Operation	Resource Type	Trace Name
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	stopTestDataTaskCreate
Combining file fragments	connection	mergeFilePart
Uploading file fragments	connection	uploadFilePart
Executing a SQL diagnosis	connection	exeTuning
Creating an OBS bucket	connection	createBucket

3.8.2 Viewing 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 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

4 FAQs

4.1 Managing connections

4.1.1 What Should I Do If I Can't Connect to My RDS for MySQL Instance?

1. Error message: **Access denied for user 'user_name'@'100.xxx.xx.xx' (using password: YES)**

   a. Error cause: The username or password of the RDS instance is incorrect.

   Solution: Check whether the username and password are correct. If you are not sure, log in to the RDS console to reset the password.

   NOTICE

   Changing the password may affect services.

   If the username and password are correct, log in to the database using a client or CLI tool and run **select * from mysql.user where user = 'user_name'** to view the account. Make sure that the DAS CIDR block is within the CIDR block of the user. **user_name @ %** and **user_name @100.%** are two different users whose passwords and permissions are independent. Enter the password of **user user_name @100.%**.

   b. Error cause: The IP address of the DAS server is not in the whitelist of the login user.

   Solution: Log in to the database using the client or CLI tool, and create a user account that can be used to access the database through DAS.

   ```
   create user 'user_name'@'100.%' identified by 'password';
   grant select on *.* to 'user_name'@'100.%';
   ```


NOTE

- Ensure that the IP address of the DAS server is in a CIDR block starting with 100. Add the IP address to the whitelist of the login user.
- Grant permissions to user **user_name@100.%** based on service requirements.

c. Error cause: The SSL function is not enabled on the server.

Solution: Run the following statement to check whether the user is an SSL user. If yes, enable SSL on the RDS instance details page. The user is an SSL user if the **ssl_type** field has a value.

```
select user, host, ssl_type from mysql.user where user = 'user_name';
```

2. Error message: **Trying to connect with ssl, but ssl not enabled in the server**

Error cause: The SSL function is not enabled on the server.

Solution: Run the following SQL statement to check whether the user is an SSL user. If yes, enable SSL on the RDS instance details page. The user is an SSL user if the **ssl_type** field has a value.

```
select user, host, ssl_type from mysql.user where user = 'user_name';
```

3. Error message: **Client does not support authentication protocol requested by server. plugin type was = 'sha256_password'**

a. Error cause: DAS does not allow you to connect to the database whose password is encrypted with SHA-256.

Solution: Execute the following SQL statements to change the password encryption method to mysql_native_password.

```
alter user 'user_name'@'%' identified with mysql_native_password by 'password';
```

b. Error cause: For MySQL 8.0, the IP address of the DAS server is not in the whitelist of the user.

Solution: Log in to the database using the client or CLI tool, and create a user that can be used to access the database through DAS.

4. Error message: **Communications link failure The last packet sent successfully to the server was 0 milliseconds ago.** The driver has not received any packets from the server

Error cause: The network between the DAS server and the target instance is disconnected.

5. Error message: **Instance connect timeout, please login again**

Error cause: The connection to the DAS server timed out.

6. Error information: **RSA public key is not available client side (option serverRsaPublicKeyFile not set).**

Error cause: The identity authentication mode of the database user has high requirements on password security. The password transmitted over the network during user authentication must be encrypted.

- If the connection is an SSL encrypted connection, the SSL certificate and key pair are used to exchange the symmetric encryption key pair (generated in the TSL handshake). Later, the symmetric encryption key pair is used to encrypt the password and data.
- For a non-SSL encrypted connection, the client uses the RSA public key of the MySQL server to encrypt the user password, and the server uses the RSA private key to decrypt and verify the password. This protects the password against snooping during network transmission.

Solution: Enable SSL for the instance or change the identity authentication mode of the database user.

4.1.2 What Should I Do If I Can't Connect to My ECS (MySQL) Instance?

1. Error message: **Access denied for user 'user_name'@'100.xxx.xx.xx' (using password: YES)**

   a. Error cause: The username or password of the self-built database on the ECS is incorrect.

      Solution: Check whether the username and password are correct. If the username and password are correct, log in to the database using a client or the CLI tool and run **select * from mysql.user where user = 'user_name'** to view the account. Make sure that the DAS CIDR block is within the CIDR block of the user. **user_name @ %** and **user_name @100.%** are two different users whose passwords and permissions are independent. Enter the password of **user user_name @100.%**.

   b. Error cause: The IP address of the DAS server is not in the whitelist of the login user.

      Solution: Log in to the database using the client or CLI tool, and create a user account that can be used to access the database through DAS.

      ```
      create user 'user_name'@'100.%' identified by 'password';
      grant all privileges on *.* to 'user_name'@'100.%';
      ```

      NOTE

      - Ensure that the IP address of the DAS server is in a CIDR block starting with 100. Add the IP address to the whitelist of the login user.
      - Grant permissions to user **user_name@100.%** based on service requirements.

   c. Error cause: The SSL function is not enabled on the server.

      Solution: Run the following statement to check whether the user is an SSL user. If yes, enable SSL on the RDS instance details page. The user is an SSL user if the **ssl_type** field has a value.

      ```
      select user, host, ssl_type from mysql.user where user = 'user_name';
      ```

2. Error message: **Host 'xxx.xxx.xx.xx' is not allowed to connect to this MySQL server**

   Error cause: The database username you entered does not support remote login. For example, if you enter username **root**, but only username **root@localhost** is configured in the **mysql.user** table, the specified user can only log in locally.

   Solution: Use a client or CLI tool to log in to the self-built database and create a user account that supports remote login.

   ```
   create user 'user_name'@'100.%' identified by 'password';
   grant all privileges on *.* to 'user_name'@'100.%';
   ```

   NOTE

   - Ensure that the IP address of the DAS server is in a CIDR block starting with 100. Add the IP address to the whitelist of the login user.
   - Grant permissions to user **user_name@100.%** based on service requirements.

3. Error message: **Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.**

   a. Error cause: The security group rules do not allow inbound traffic on the port.

      Solution: Modify the security group rules by referring to **How Do I View and Modify ECS Security Group Rules?**

   b. Error cause: The firewall policy does not allow inbound traffic on the port.

      Solution: Modify the firewall policy by referring to **How Do I View and Modify Firewall Rules?**

   c. Error cause: The remote login times out because the DNS resolution takes a long period of time.

      Solution: Rectify the fault by performing the following operations:

      i. Search for the configuration file of the self-built database in directory **/etc/my.cn**, enter the following content in **[mysqld]**, save the change and exit.

      ```
      skip-name-resolve
      ```

      NOTE

      The default location of the configuration file is **/etc/my.cnf**. If you store the file in the specified path, modify the directory accordingly.

      ii. Run **systemctl restart mysqld** to restart the database and log in again.

4. Error message: **Communications link failure The last packet sent successfully to the server was 0 milliseconds ago.** The driver has not received any packets from the server

   Error cause: The network between the DAS server and the target instance is disconnected.

   Solution: Check whether the firewall of the instance is correctly configured and whether the required port is enabled. If the firewall is abnormal or the port is not enabled, rectify the fault and try again. If the fault persists, .

5. Error message: **Instance connect timeout, please login again.**

   Error cause: The connection to the DAS server timed out.

   Solution: Rectify the fault by performing the following operations:

   a. Log in to a remote ECS and run the **iptables -S | grep input** command to view firewall configurations of the instance. If the self-built database port is not included in the firewall whitelist, add an iptables rule or run the **systemctl stop iptables** command to disable the firewall to allow traffic through this port, and try again.

   b. Log in to the ECS again and run the **ps -ef | grep mysql** command to check whether the database process is running. If processes **mysqld_safe** and **mysqld** are both running, the database process is normal. If the process is not running, run **systemctl start mysqld** to restart the database and try again.

   c. If the fault persists, .

4.1.3 What Should I Do If I Can't Connect to My RDS for PostgreSQL Instance?

Error message: **FATAL: Invalid username/password,login denied.**

Error cause: The username or password of the RDS DB instance is incorrect.

Solution: Check whether the username or password is correct. If you are not sure, view the username and reset the password on the RDS console.

NOTICE

Changing the password may affect services.

4.1.4 What Should I Do If I Can't Connect to My ECS (PostgreSQL) Instance?

Error message: **Connection refused (Connection refused).**

Error cause: The port number of the self-built database is incorrect, or the network is disconnected.

Solution: Ensure that the port number of the self-built database is correct and that the port is included in the security group rule and firewall whitelist. For details, see **How Do I View and Modify ECS Security Group Rules?** and **How Do I View and Modify Firewall Rules?**.

4.1.5 What Should I Do If I Can't Connect to My DDS Instance?

Error message: Command failed with error 18 (AuthenticationFailed): 'Authentication failed.' on server xxx.xxx.xx.xx:xxxx. The full response is { 'ok' : 0.0, 'errmsg' : "Authentication failed.", "code" : 18, "codeName" : "AuthenticationFailed" }

1. Error cause: The username or password of the DDS DB instance is incorrect.

   Solution: Check whether the username or password is correct. If you are not sure, view the username or reset the password on the DDS console.

   NOTICE

   Changing the password may affect services.

2. Error cause: The entered username does not have the permission to access the database.

   Solution: Check whether the username has the permission to access the database. If you are not sure, connect to the admin database as user **rwuser**. Then check whether the entered username has the required permission.

4.1.6 What Can I Do If a GaussDB(for MySQL) Instance Fails to Be Connected?

1. Error message: **Access denied for user 'user_name'@'100.xxx.xx.xx' (using password: YES).**

   a. Error cause: The username or password of a GaussDB(for MySQL) database is incorrect.

   Solution: Check whether the username and password are correct. If you are not sure, log in to the GaussDB console to view the username and reset the password.

   NOTICE

   Changing the password may affect services.

   If the username and password are correct, log in to the database using a client or CLI tool and run **select * from mysql.user where user = 'user_name'** to view the account. Make sure that the DAS CIDR block is within the CIDR block of the user. **user_name @ %** and **user_name @100.%** are two different users whose passwords and permissions are independent. Make sure to enter the password of **user user_name @100.%**.

   b. Error cause: The IP address of the DAS server is not in the whitelist of the login user.

   Solution: Log in to the database using the client or CLI tool, and create a user that can be used to access the database through DAS.

   ```
   create user 'user_name'@'100.%' identified by 'password';
   grant all privileges on *.* to 'user_name'@'100.%';
   ```

   NOTE

   1. Ensure that the IP address of the DAS server is in a CIDR block starting with 100. Add the IP address to the whitelist of the login user.
   2. Grant permissions to user **user_name@100.%** based on service requirements.

2. Error message: **Trying to connect with ssl, but ssl not enabled in the server**

   Error cause: The SSL function is not enabled on the server.

   Solution: Run the following statement to check an SSL user is used. If yes, enable SSL on the GaussDB(for MySQL) instance details page. The user is an SSL user if the **ssl_type** field has a value.

   ```
   select user, host, ssl_type from mysql.user where user = 'user_name';
   ```

3. Error message: **Client does not support authentication protocol requested by server.** plugin type was = 'sha256_password'

   Error cause: DAS does not allow you to connect to the database whose password is encrypted with SHA-256.

   Solution: Execute the following SQL statements to change the password encryption method to mysql_native_password.

   ```
   alter user 'user_name'@'%' identified with mysql_native_password by 'password';
   ```


4. Error message: **Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.**

   Error cause: The network between the DAS server and the instance is disconnected.

   Solution: Contact technical support.

4.1.7 What Should I Do If I Can't Connect to My GaussDB Instance?

Error message: **FATAL: Invalid username/password,login denied.**

Error cause: The username or password of the GaussDB instance is incorrect.

Check whether the username or password is correct. If you are not sure, view the username and reset the password on the GaussDB console.

NOTICE

Changing the password may affect services.

4.1.8 What Should I Do If I Can't Connect to My DDM Instance?

Error message: **User has no databases**

Error cause: The DDM account has not been associated with any schema.

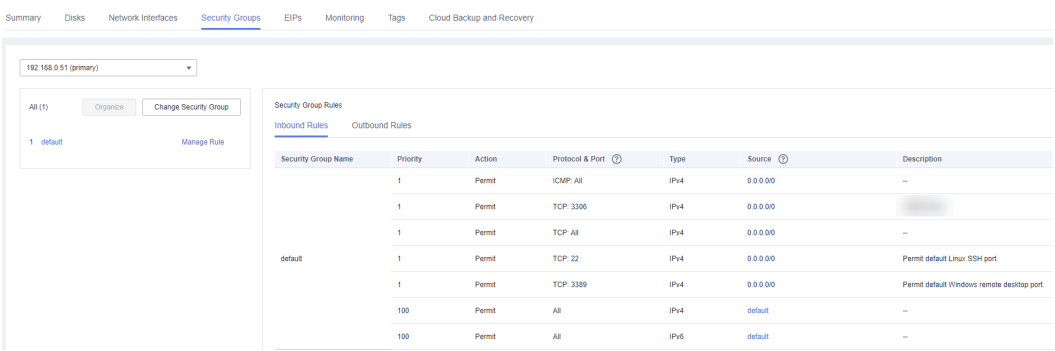
Solution: On the DDM instance basic information page, choose **Accounts** in the navigation pane on the left and associate the DDM account with the required schema.

4.1.9 How Do I View and Modify ECS Security Group Rules?

To enable DAS to access the self-built DB instances on ECSs, you need to add an inbound rule with the port set to 3306 (example) and source to 100.125.0.0/16 and 100.79.0.0/16.

Step 1 On the ECS details page, click the **Security Groups** tab and view security group rules.

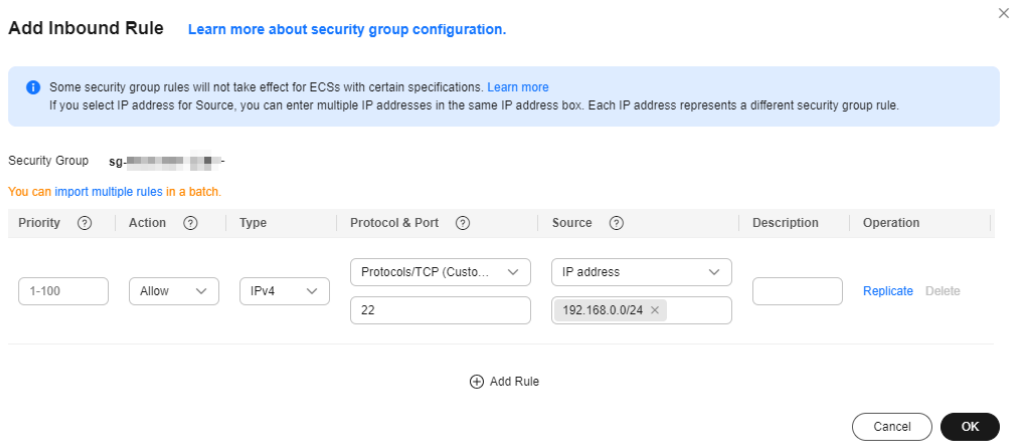
Figure 4-1 Security group rules



Step 2 Click **Manage Rule** on the left.

Step 3 On the **Inbound Rules** tab page, click **Add Inbound Rule**. .

Figure 4-2 Adding an inbound rule

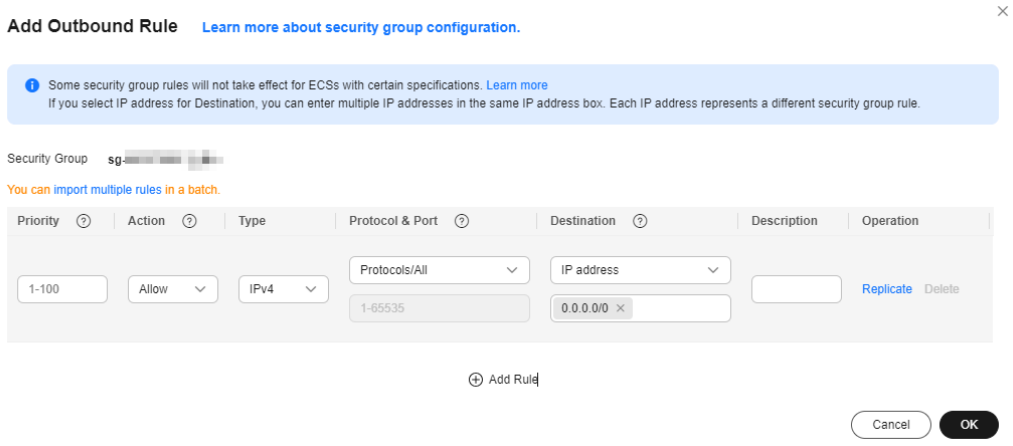


NOTE

Recommended configuration: Select **TCP** for **Protocols & TCP (Custom ports)**, enter the port number of the self-built database, and set the source to 100.125.0.0/16 and 100.79.0.0/16 respectively.

Step 4 On the **Outbound Rules** tab page, click **Add Outbound Rule**.

Figure 4-3 Adding an outbound rule



NOTE

Recommended configuration: Select **TCP (Custom ports)** for **Protocol & Port**, enter the port number of the self-built database, and set the source to 100.125.0.0/16 and 100.79.0.0/16 respectively.

----End

4.1.10 How Do I View and Modify Firewall Rules?

Step 1 In the ECS list, locate the required ECS and click **Remote Login**.

Step 2 Enter the username and password. After the login is successful, run the following command to check the iptables configuration:

```
iptables -S
```

NOTE

- The port next to **--dport** indicates the port that can be accessed.
- Perform the following operations to ensure that the port can be accessed:
  - Add an iptables rule to allow access to the port.
  - Run the following command to disable the firewall:
    ```
    systemctl stop iptables
    ```

----End

4.1.11 What Should I Do If My Connection Fails?

Error information: The connection does not exist.

Error cause: The shared connection is associated with the project of a shared user, which does not match the login project of the shared connection used by the current user.

Solution: In the upper left corner on the console, select another project in the current region and log in to the DB instance again.

4.2 Usage

4.2.1 What Can I Do If Garbled Characters Are Displayed in the Exported Database Result Set?

CSV files exported from DAS are encoded in UTF-8.

Excel files are encoded in ANSI. Garbled characters are caused by inconsistent encoding modes.

You are advised to open the CSV file using a text editor and save the file in ANSI encoding.

4.2.2 How Do I Modify the Collation?

DAS does not support modifications of SQL Server on the GUI. You can run commands instead.

Go to the SQL Window page of the database and run the following commands:

In this example, the character set of the **test** database is set to **SQL_Latin1_General_CP1_CI_AS**.


```
use root
go
ALTER DATABASE test SQL_Latin1_General_CP1_CI_AS
```


A Change History

Released On	Description
2025-01-07	This issue is the first official release.

