# Data Admin Service

# Best Practices

| | |
|---|---|
| **Issue** | 01 |
| **Date** | 2025-09-15 |

**HUAWEI TECHNOLOGIES CO., LTD.**

## Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

## Notice

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

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

# Security Declaration

## Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

# Contents

# 1 How Do I Use DAS to Log In to an Instance Using a Read-Only Account?

System permission policies of Data Admin Service (DAS) do not support read-only accounts. However, you can create a custom policy on the IAM console and assign the read-only permission on DAS.

## Differences Between IAM Permissions and Database Permissions

As a management plane service, DAS does not allow users to add, delete, or modify instances. Only adding, deleting, and modifying instance login information are allowed on the DAS console.

IAM permission control applies to DAS only before you log in to an instance. After you have logged in to the instance, permissions are assigned by your database account.

You can use IAM **Permissions Management** to control whether IAM users can add, delete, and modify instance logins and whether they can log in to an instance. After a user logs in to an instance, only the database account controls whether the user can execute SQL statements.

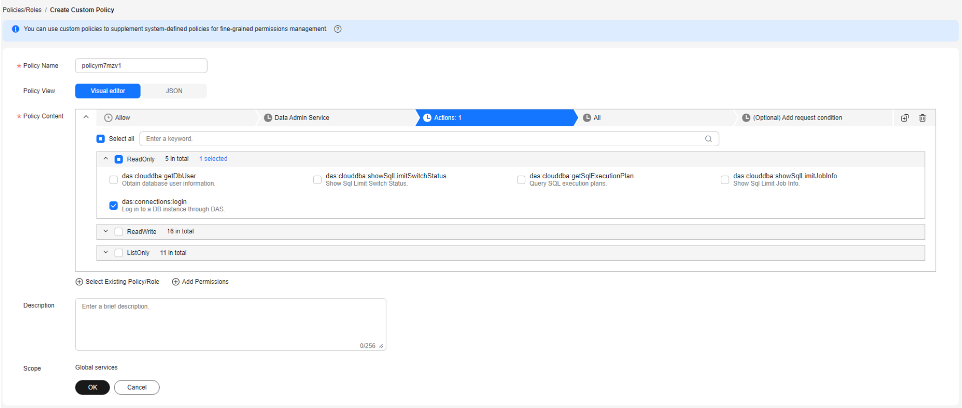
## Procedure

**Step 1** Log in to the IAM console using a Huawei ID account.

**Step 2** Create a custom permission policy.

1. In the navigation pane, choose **Permissions** > **Policies/Roles** and click **Create Custom Policy**.
2. Configure parameters.

**Figure 1-1** Configuring a custom permission policy



Policies/Roles / Create Custom Policy

You can use custom policies to supplement system-defined policies for fine-grained permissions management.

* Policy Name: policym7mzv1

Policy View: Visual editor | JSON

* Policy Content: Allow | Data Admin Service | Actions: 1 | All | (Optional) Add request condition

☑ Select all

☑ ReadOnly 5 in total 1 selected

- □ das:clouddba:getDbUser Obtain database user information.
- □ das:clouddba:showSqlLimitSwitchStatus Show Sql Limit Switch Status.
- □ das:clouddba:getSqlExecutionPlan Query SQL execution plans.
- □ das:clouddba:showSqlLimitJobInfo Show Sql Limit Job Info.
- ☑ das:connections:login Log in to a DB instance through DAS.

□ ReadWrite 16 in total

□ ListOnly 11 in total

Description: Enter a brief description. 0/256

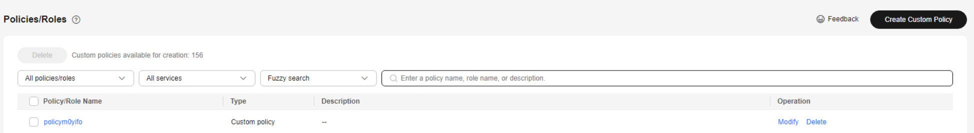
Scope: Global services

**Table 1-1** Parameter description

| Parameter | Description | Example |
|---|---|---|
| Policy Name | Customize a name. | das-log-database |
| Policy View | Select **Visual editor** or **JSON**. | Visual editor |
| Policy Content | Choose DAS and add the read-only permission as required.<br>Take the **das:connections:login** permission as an example. A user or user group with this permission can only log in to an instance using DAS. | das:connections:login |

3. Click **OK**. You can then view the created custom permission policy on the **Policies/Roles** page.

**Figure 1-2** Viewing a Custom permission policy



Policies/Roles

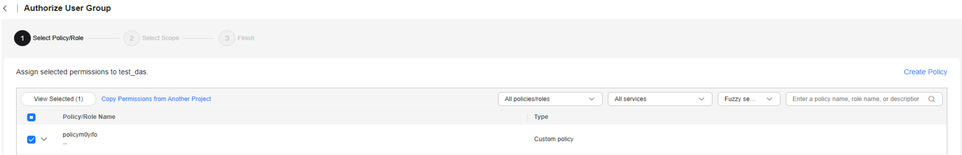
Custom policies available for creation: 156

| Policy/Role Name | Type | Description | Operation |
|---|---|---|---|
| policym0yifo | Custom policy | -- | Modify Delete |

**Step 3** Create a user group and assign the custom permission policy created in **Step 2** to the user group.

1. In the navigation pane on the left, choose **User Groups**. Then, click **Create User Group**. In the displayed dialog box, specify the user group name and click **OK**.
2. Locate the target user group and click **Authorize** in the **Operation** column. On the displayed page, select the custom policy created in **Step 2**.

**Figure 1-3** Authorization



Assign selected permissions to test_das.

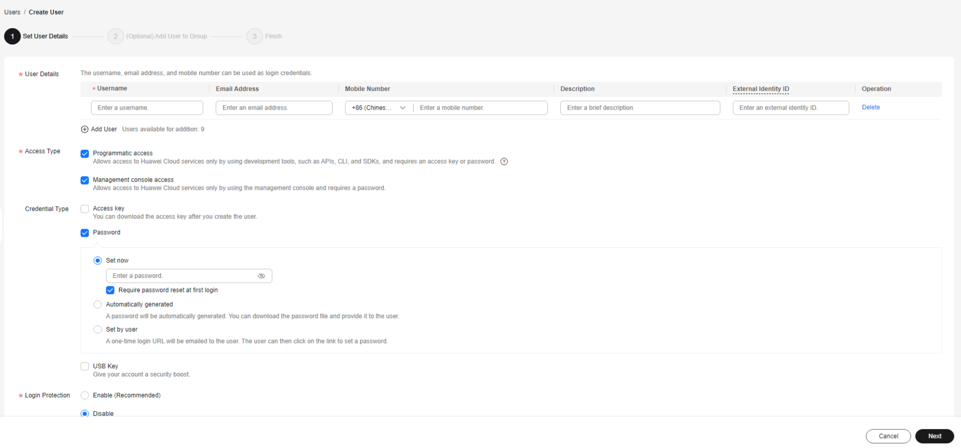
| | Policy/Role Name | Type |
|---|---|---|
| ☑ | policym0yifo<br>-- | Custom policy |

3. Click **Next**, select **All resources**, click **OK**, and complete subsequent operations.

**Step 4** Create a user and add it to the user group.

1. In the navigation pane, choose **Users** and click **Create**. On the displayed page, set user basic information.

**Figure 1-4** Setting user basic information



* User Details: The username, email address, and mobile number can be used as login credentials.

| * Username | Email Address | Mobile Number | Description | External Identity ID | Operation |
|---|---|---|---|---|---|
| Enter a username. | Enter an email address. | +86 (Chines... Enter a mobile number. | Enter a brief description. | Enter an external identity ID. | Delete |

Add User Users available for addition: 9

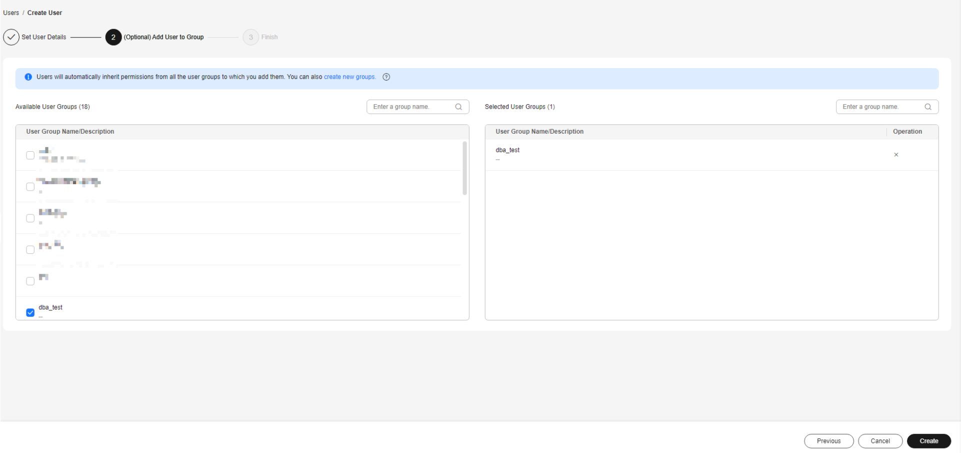
* Access Type:
☑ Programmatic access
Allows access to Huawei Cloud services only by using development tools, such as APIs, CLI, and SDKs, and requires an access key or password.
☑ Management console access
Allows access to Huawei Cloud services only by using the management console and requires a password.

Credential Type:
□ Access key
You can download the access key after you create the user.
☑ Password
● Set now
Enter a password.
☑ Require password reset at first login
○ Automatically generated
A password will be automatically generated. You can download the password file and provide it to the user.
○ Set by user
A one-time login URL will be emailed to the user. The user can then click on the link to set a password.
□ USB Key
Give your account a security boost.

* Login Protection:
○ Enable (Recommended)
● Disable

2. Click **Next** to add the current user to the user group created in **Step 3**.

**Figure 1-5** Adding a user to a user group



Users will automatically inherit permissions from all the user groups to which you add them. You can also create new groups.

Available User Groups (18)

| | User Group Name/Description |
|---|---|
| □ | [redacted] |
| ☑ | dba_test<br>-- |

Selected User Groups (1)

| User Group Name/Description | Operation |
|---|---|
| dba_test<br>-- | × |

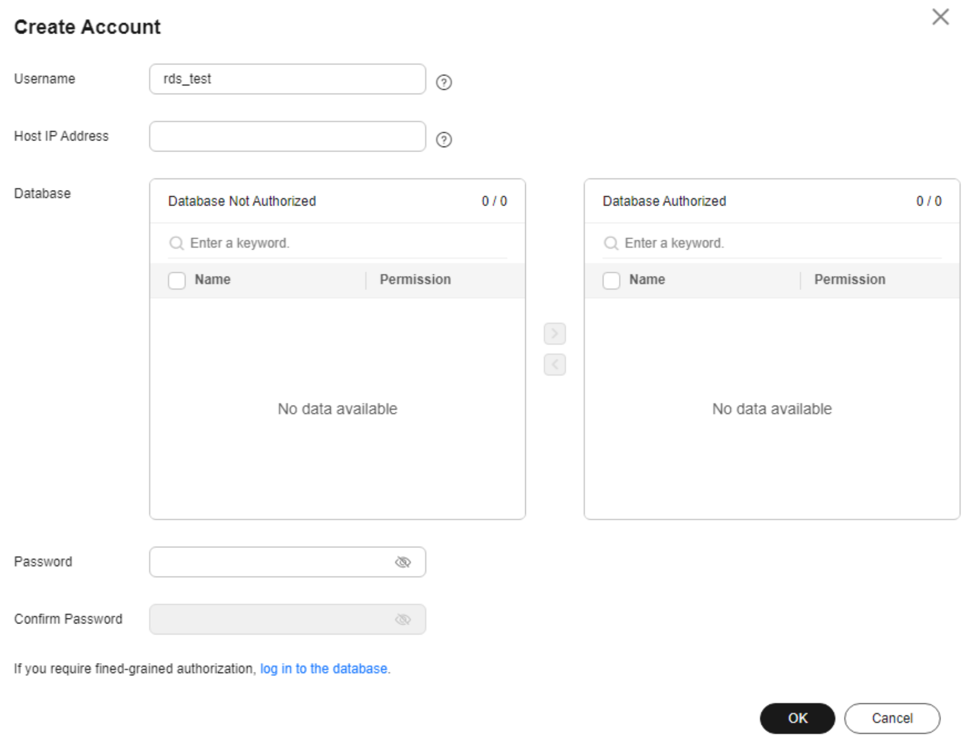
3. Click **Create User** to create an IAM user. The user has only the permission to log in to the instance on DAS.

**Step 5** Create a read-only account. An RDS for MySQL instance is used in this example.

1. Log in to the RDS console.
2. On the **Instances** page, locate the target instance and click its name.

3. In the navigation tree on the left, choose **Accounts**. On the displayed page, click **Create Account**.

**Figure 1-6** Creating a read-only account



Create Account

Username: rds_test

Host IP Address:

Database:

| Database Not Authorized | 0 / 0 |
|---|---|
| Name | Permission |
| No data available | |

| Database Authorized | 0 / 0 |
|---|---|
| Name | Permission |
| No data available | |

Password:

Confirm Password:

If you require fined-grained authorization, log in to the database.

NOTE

You can also log in to the RDS for MySQL instance and run the following commands to create a read-only account:

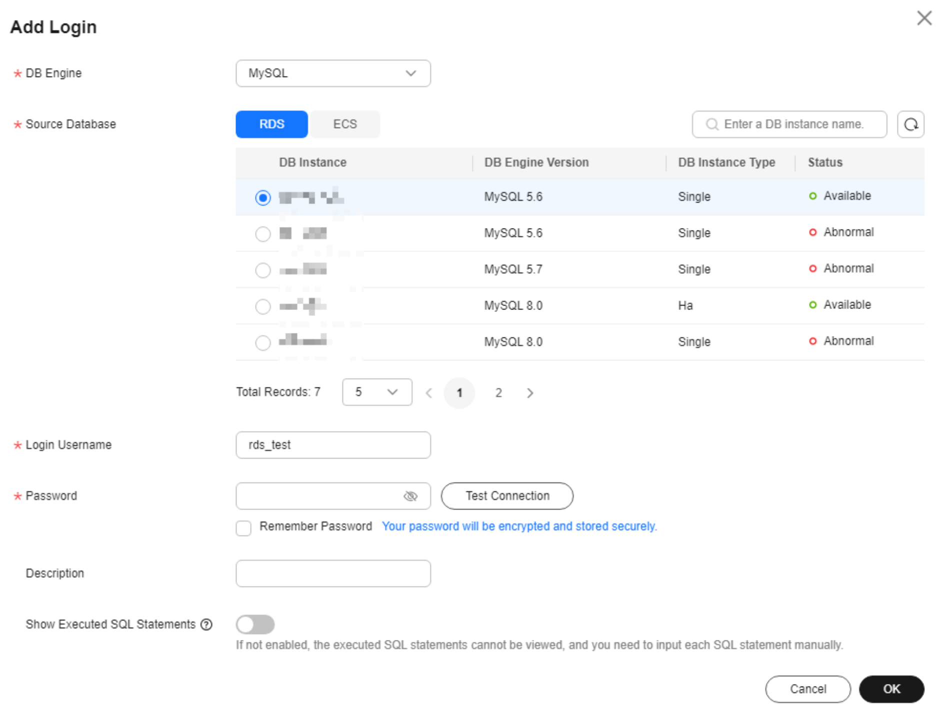
```
CREATE USER 'db_read_only'@'%' IDENTIFIED BY '**********';
GRANT SELECT ON *.* TO 'db_read_only'@'%';
FLUSH PRIVILEGES;
```

**Step 6** Authorize the read-only permission to the IAM user.

1. Log in to the DAS console using the Huawei ID account.
2. Use the read-only account to add a login.

   In the navigation tree on the left, choose **Development Tool**. On the **My DB Instance Logins** tab page, click **Add Login**.

**Figure 1-7** Adding a login instance



Add Login

* DB Engine: MySQL

* Source Database: RDS | ECS — Enter a DB instance name.

| | DB Instance | DB Engine Version | DB Instance Type | Status |
|---|---|---|---|---|
| ● | [redacted] | MySQL 5.6 | Single | Available |
| ○ | [redacted] | MySQL 5.6 | Single | Abnormal |
| ○ | [redacted] | MySQL 5.7 | Single | Abnormal |
| ○ | [redacted] | MySQL 8.0 | Ha | Available |
| ○ | [redacted] | MySQL 8.0 | Single | Abnormal |

Total Records: 7 — 5 — 1 2

* Login Username: rds_test

* Password: — Test Connection

□ Remember Password Your password will be encrypted and stored securely.

Description:

Show Executed SQL Statements

If not enabled, the executed SQL statements cannot be viewed, and you need to input each SQL statement manually.

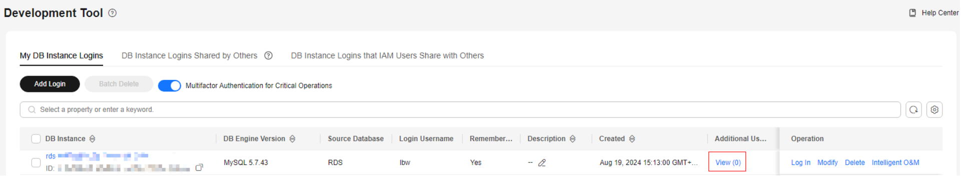
Cancel OK

**The login username is the read-only account created in Step 5.**

3. Share the login information of the read-only account with the IAM user.

   Locate the target instance and click the number in the **Additional Users** column.

**Figure 1-8** Sharing a login with an IAM user



Development Tool — Help Center

My DB Instance Logins | DB Instance Logins Shared by Others | DB Instance Logins that IAM Users Share with Others

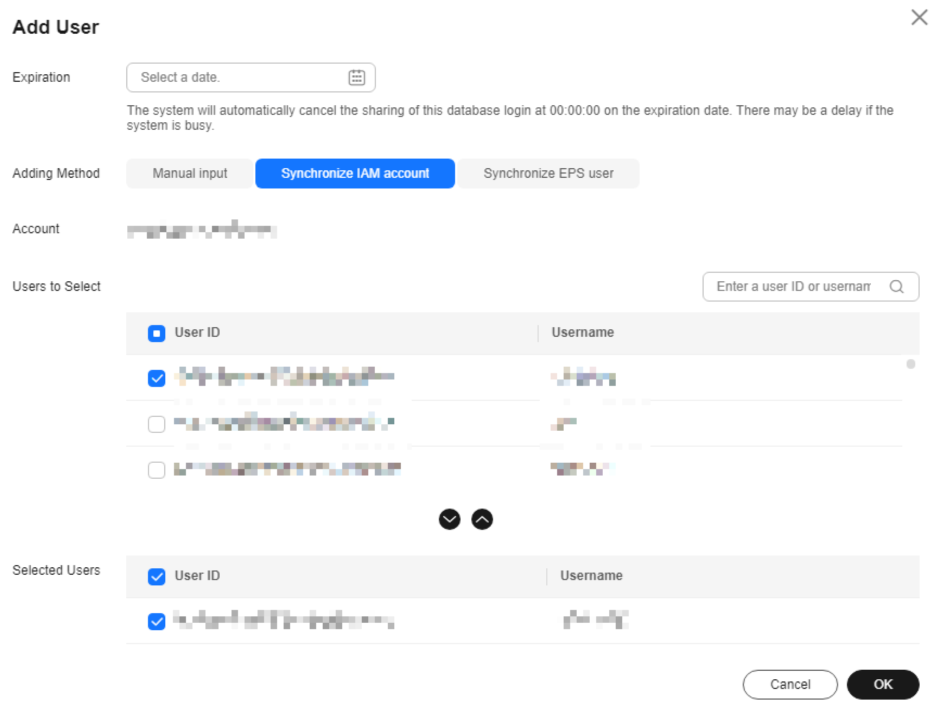
Add Login | Batch Delete | Multifactor Authentication for Critical Operations

Select a property or enter a keyword.

| □ | DB Instance | DB Engine Version | Source Database | Login Username | Remember... | Description | Created | Additional Us... | Operation |
|---|---|---|---|---|---|---|---|---|---|
| □ | rds [redacted]<br>ID: [redacted] | MySQL 5.7.43 | RDS | lbw | Yes | -- | Aug 19, 2024 15:13:00 GMT+... | View (0) | Log In Modify Delete Intelligent O&M |

Click **Add User**. On the displayed page, specify the expiration time, select **Synchronize IAM account** for **Adding Method**, then select the IAM account created in **Step 4** for **Users to Select**, and click **OK**.

**Figure 1-9** Adding a user



Add User

Expiration: Select a date.

The system will automatically cancel the sharing of this database login at 00:00:00 on the expiration date. There may be a delay if the system is busy.

Adding Method: Manual input | **Synchronize IAM account** | Synchronize EPS user

Account: [redacted]

Users to Select: Enter a user ID or usernam

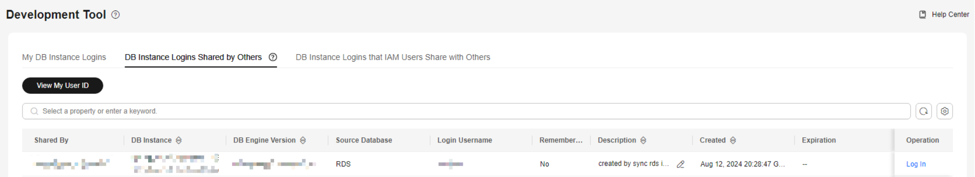
| ■ User ID | Username |
|---|---|
| ☑ [redacted] | [redacted] |
| □ [redacted] | [redacted] |
| □ [redacted] | [redacted] |

Selected Users

| ☑ User ID | Username |
|---|---|
| ☑ [redacted] | [redacted] |

**Step 7** Log in to DAS using the IAM account created in **Step 4**, and verify that it has the read-only permission.

**Figure 1-10** Verifying the read-only permission



Development Tool

Help Center
My DB Instance Logins
DB Instance Logins Shared by Others
DB Instance Logins that IAM Users Share with Others
View My User ID
Select a property or enter a keyword.

| Shared By | DB Instance | DB Engine Version | Source Database | Login Username | Remember... | Description | Created | Expiration | Operation |
|---|---|---|---|---|---|---|---|---|---|
| [redacted] | [redacted] | [redacted] | RDS | [redacted] | No | created by sync rds i... | Aug 12, 2024 20:28:47 G... | -- | Log In |

After logging in to DAS as the IAM user, choose **Development Tool** > **DB Instance Logins Shared by Others** to view the logins shared by the Huawei account. Only **Log In** is displayed in the **Operation** column.

**----End**

# 2 How Do I Check and Optimize Tables by Checking Top SQL?

## Example Problem

A user found in the exported logs that it took more than 2s for a SELECT statement to query information of table **test** and the lock wait duration was long.

## Suggestion

- Add indexes.
- Optimize tables

## Procedure

**Step 1** Log in to the Huawei Cloud console.

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

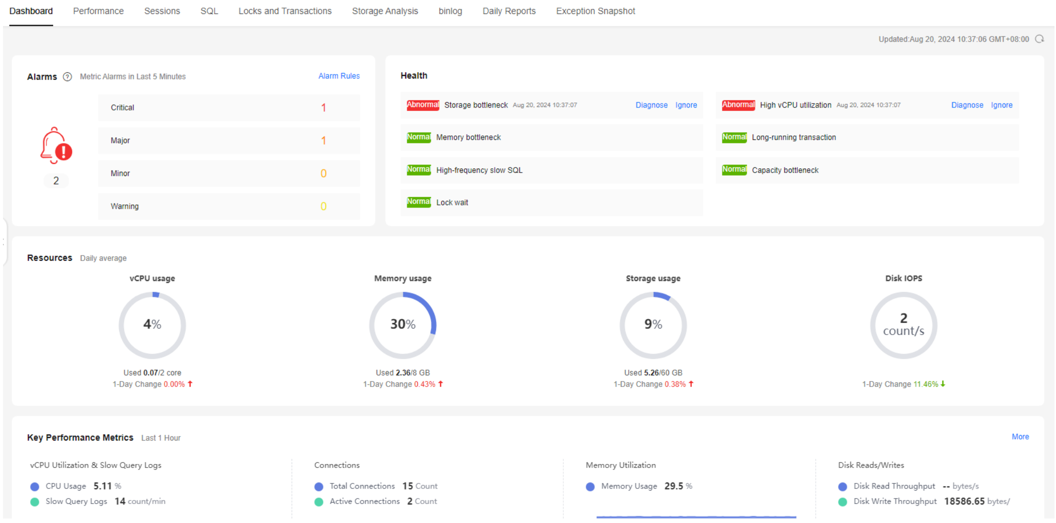
**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.

**Step 5** In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details**.

**Figure 2-1** Dashboard



Updated:Aug 20, 2024 10:37:06 GMT+08:00

Alarms Metric Alarms in Last 5 Minutes Alarm Rules

2

| | |
|---|---|
| Critical | 1 |
| Major | 1 |
| Minor | 0 |
| Warning | 0 |

Health

| | |
|---|---|
| Abnormal Storage bottleneck Aug 20, 2024 10:37:07 Diagnose Ignore | Abnormal High vCPU utilization Aug 20, 2024 10:37:07 Diagnose Ignore |
| Normal Memory bottleneck | Normal Long-running transaction |
| Normal High-frequency slow SQL | Normal Capacity bottleneck |
| Normal Lock wait | |

Resources Daily average

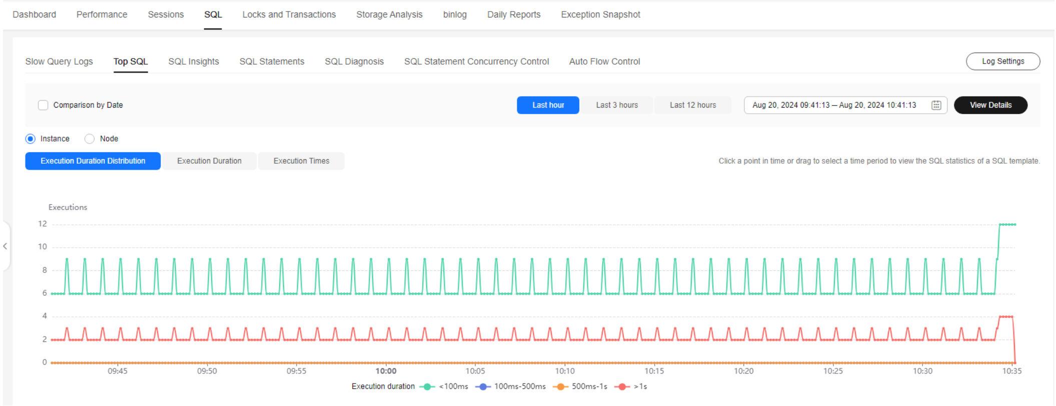
| vCPU usage | Memory usage | Storage usage | Disk IOPS |
|---|---|---|---|
| 4% | 30% | 9% | 2 count/s |
| Used 0.07/2 core | Used 2.36/8 GB | Used 5.26/60 GB | |
| 1-Day Change 0.00% ↑ | 1-Day Change 0.43% ↑ | 1-Day Change 0.38% ↑ | 1-Day Change 11.46% ↓ |

Key Performance Metrics Last 1 Hour More

| vCPU Utilization & Slow Query Logs | Connections | Memory Utilization | Disk Reads/Writes |
|---|---|---|---|
| CPU Usage 5.11 % | Total Connections 15 Count | Memory Usage 29.5 % | Disk Read Throughput -- bytes/s |
| Slow Query Logs 14 count/min | Active Connections 2 Count | | Disk Write Throughput 18586.65 bytes/ |

**Step 6** On the displayed page, click the **SQL** tab and then **TOP SQL**.

**Figure 2-2** TOP SQL



□ Comparison by Date

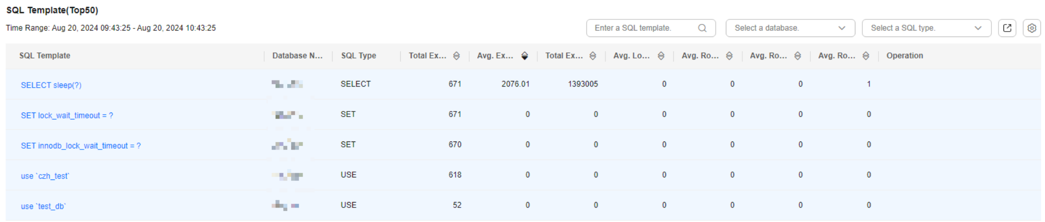
Aug 20, 2024 09:41:13 — Aug 20, 2024 10:41:13

● Instance ○ Node

Click a point in time or drag to select a time period to view the SQL statistics of a SQL template.

**Step 7** On the **TOP SQL** tab page, click **View Details** to view template information of SELECT and find the database in which SELECT was executed for longer than 2s.

**Figure 2-3** SQL statements


SQL Template(Top50)

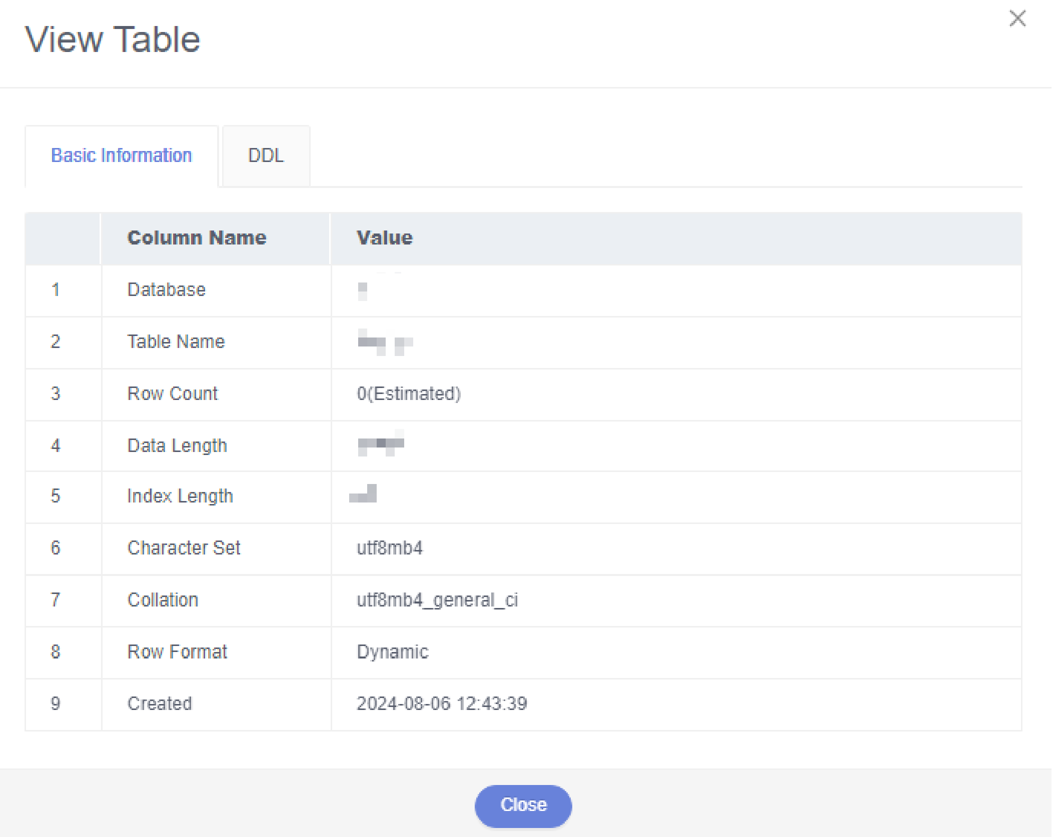
Time Range: Aug 20, 2024 09:43:25 - Aug 20, 2024 10:43:25

Enter a SQL template. Select a database. Select a SQL type.

| SQL Template | Database N... | SQL Type | Total Ex... | Avg. Ex... | Total Ex... | Avg. Lo... | Avg. Ro... | Avg. Ro... | Avg. Ro... | Operation |
|---|---|---|---|---|---|---|---|---|---|---|
| SELECT sleep(?) | [redacted] | SELECT | 671 | 2076.01 | 1393005 | 0 | 0 | 0 | 1 | |
| SET lock_wait_timeout = ? | [redacted] | SET | 671 | 0 | 0 | 0 | 0 | 0 | 0 | |
| SET innodb_lock_wait_timeout = ? | [redacted] | SET | 670 | 0 | 0 | 0 | 0 | 0 | 0 | |
| use `czh_test` | [redacted] | USE | 618 | 0 | 0 | 0 | 0 | 0 | 0 | |
| use `test_db` | [redacted] | USE | 52 | 0 | 0 | 0 | 0 | 0 | 0 | |

**Step 8** Log in to the target instance on the **Development Tool** page and choose **Database Management**. Select the database found in **Step 7**. Choose **Tables** in the navigation pane on the left, locate the table that you want to view, and click **View** in the **Operation** column. View the index length and row count in the table.

**Figure 2-4** Viewing table details



| | Column Name | Value |
|---|---|---|
| 1 | Database | [redacted] |
| 2 | Table Name | [redacted] |
| 3 | Row Count | 0(Estimated) |
| 4 | Data Length | [redacted] |
| 5 | Index Length | [redacted] |
| 6 | Character Set | utf8mb4 |
| 7 | Collation | utf8mb4_general_ci |
| 8 | Row Format | Dynamic |
| 9 | Created | 2024-08-06 12:43:39 |

**Step 9** (Example) If there are few indexes, click **Alter** and add indexes. Return to the **Tables** tab and click **Query SQL Statements**.

**----End**

# 3 Fixing Slow SQL Queries

You can use DAS to monitor databases and identify and fix slow SQL queries to improve database performance.

## Solution

You can use DAS to:

- Viewing and Optimizing Slow SQL Statements of a Single Instance
- Viewing Slow SQL Queries on All Instances

## Viewing and Optimizing Slow SQL Statements of a Single Instance

**Step 1** Log in to the Huawei Cloud console.

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

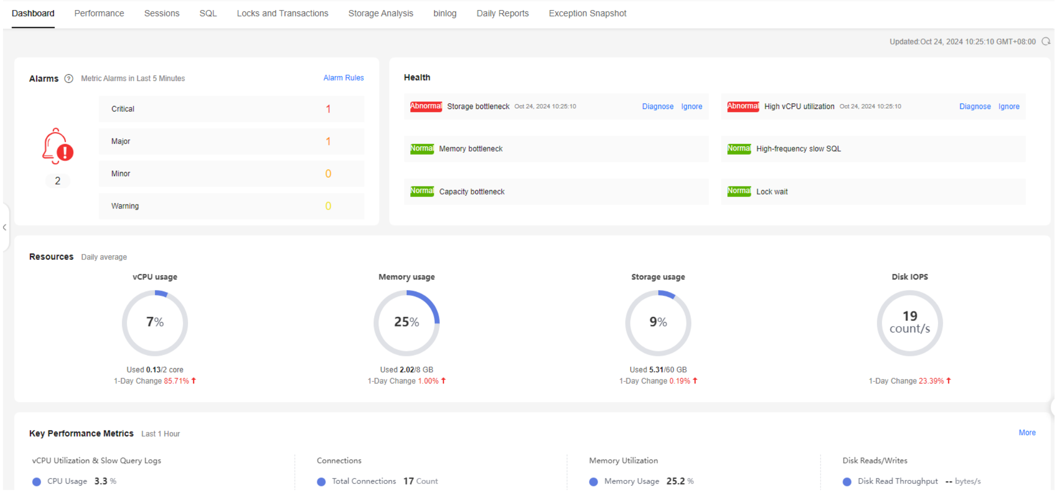
**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.

**Step 5** In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details** to go to the **Dashboard** tab page.

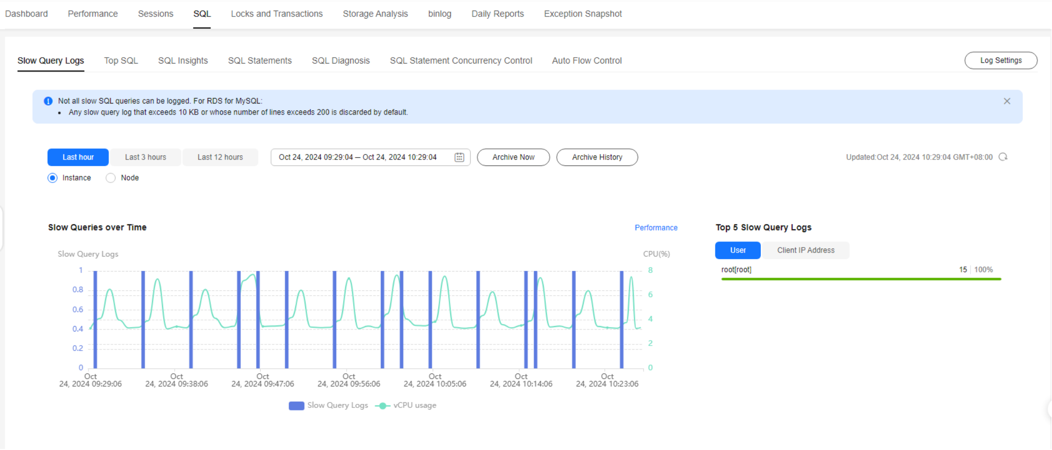
**Figure 3-1** Dashboard



**Step 6** Choose **SQL** > **Slow Query Logs**.

Select a time range and view trends, details, and statistics of the slow query logs. You can click **Export** to export a slow query log to your local PC.

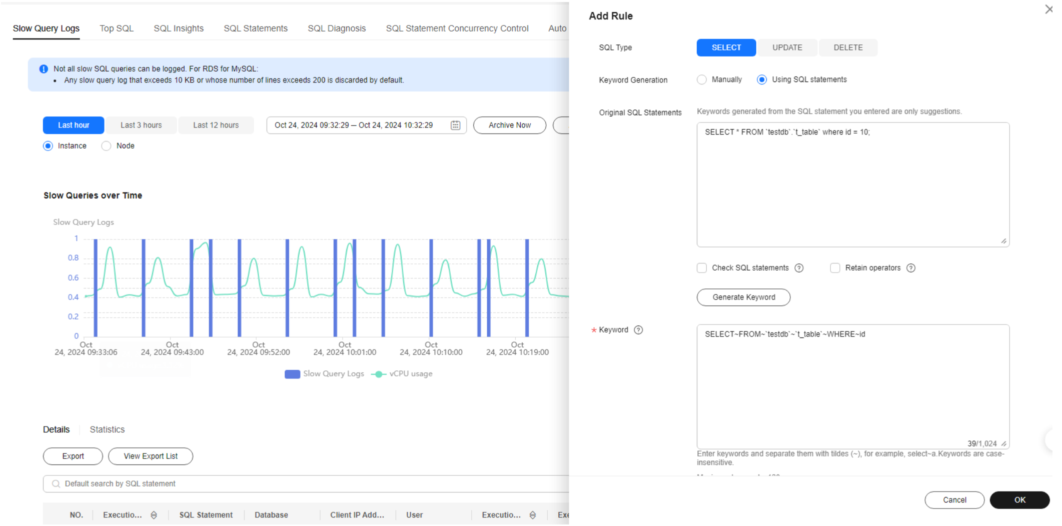
**Figure 3-2** Slow Query Logs


Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

**Step 7** In the **Details** area, click **Concurrency Control** in the **Operation** column. In the displayed dialog box, create a concurrency control rule for the current SQL statement by configuring parameters, including **Max. Concurrency** to ensure stability of core services. For details, see **SQL Statement Concurrency Control**.

**Figure 3-3** Concurrency control



Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

**Add Rule**

SQL Type: SELECT | UPDATE | DELETE (SELECT selected)

Keyword Generation: ○ Manually ● Using SQL statements

Original SQL Statements: Keywords generated from the SQL statement you entered are only suggestions.

```
SELECT * FROM `testdb`.`t_table` where id = 10;
```

□ Check SQL statements □ Retain operators

Generate Keyword

* Keyword:

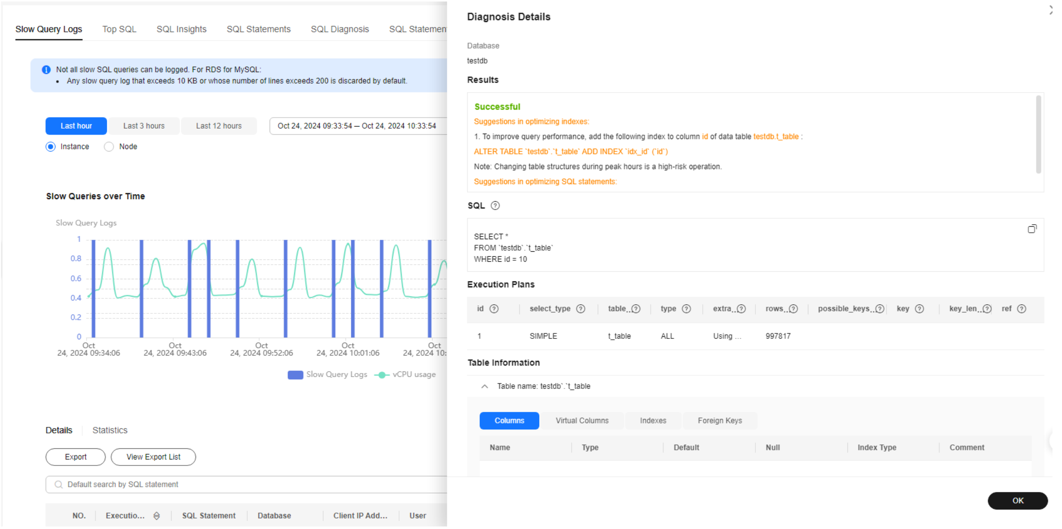
```
SELECT~FROM~`testdb`~`t_table`~WHERE~id
```

39/1,024

Enter keywords and separate them with tildes (~), for example, select~a.Keywords are case-insensitive.

**Step 8** In the **Details** area, click **Diagnose** in the **Operation** column to diagnose the current SQL statement and view index or statement optimization suggestions and an execution plan. You can paste optimized SQL statement to the database client or DAS for execution. For details, see **SQL Diagnosis**.

**Figure 3-4** Slow SQL diagnosis



Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

**Diagnosis Details**

Database: testdb

Results

Successful

Suggestions in optimizing indexes:

1. To improve query performance, add the following index to column id of data table testdb.t_table :

```
ALTER TABLE `testdb`.`t_table` ADD INDEX `idx_id` (`id`)
```

Note: Changing table structures during peak hours is a high-risk operation.

Suggestions in optimizing SQL statements:

SQL

```
SELECT *
FROM `testdb`.`t_table`
WHERE id = 10
```

Execution Plans

| id | select_type | table_.. | type | extra_.. | rows_.. | possible_keys_.. | key | key_len_.. | ref |
|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | t_table | ALL | Using ... | 997817 | | | | |

Table Information

Table name: testdb`.`t_table

| Name | Type | Default | Null | Index Type | Comment |
|---|---|---|---|---|---|

**----End**

## Viewing Slow SQL Queries on All Instances

**Step 1** Log in to the Huawei Cloud console.

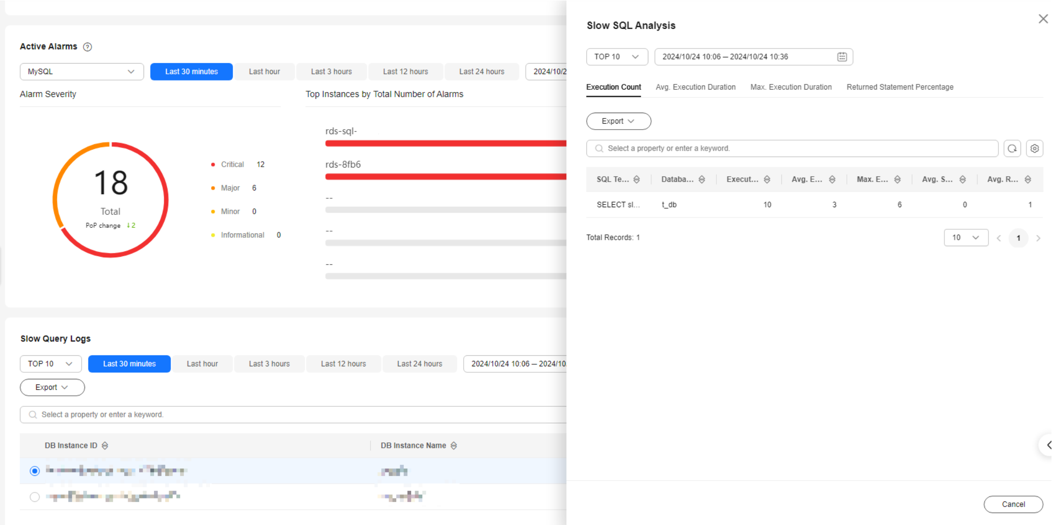
**Step 2** Click in the upper left corner and select a region and project.

**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

Select a time range and view the number of slow SQL queries of the top 10 to 30 instances in the current region. You can also view information about the slow SQL template of an instance.

**Figure 3-5** Slow query logs on the dashboard



| DB Instance ID | DB Instance Name |
|---|---|
| ● [redacted] | [redacted] |
| ○ [redacted] | [redacted] |

| SQL Te... | Databa... | Execut... | Avg. E... | Max. E... | Avg. S... | Avg. R... |
|---|---|---|---|---|---|---|
| SELECT sl... | t_db | 10 | 3 | 6 | 0 | 1 |

**----End**

# 4 Lock Analysis

Intelligent O&M provides metadata lock, InnoDB lock wait, recent deadlock, and full deadlock analysis. This section describes how to perform lock analysis on an RDS for MySQL instance.

## Prerequisites

You have created an RDS for MySQL instance.

## Procedure

**Step 1** Write test data to a table.

1. Create test database **das_test** in the destination RDS for MySQL instance. For details, see **Creating a Database**.
2. Log in to the RDS for MySQL database through DAS. For details, see **Logging In to a Huawei Cloud DB Instance**.
3. Run the following SQL statement to create the **shopping** table in the **das_test** database:

   **CREATE TABLE** *shopping* **(**

   *a* **int** *NOT NULL AUTO_INCREMENT*,

   *b* **int,**

   *c* **int,**

   **PRIMARY KEY** *(a)*,

   **UNIQUE KEY** *u_k (b, c)*

   **);**
4. Run the following command to write test data to the **shopping** table:

   **insert into** *shopping(b,c)* **values** *(1,1),(1,5),(1,9)*;

NOTE

After data is written to the table:

- Perform **Step 2** to analyze and process metadata locks.
- Perform **Step 3** to analyze InnoDB lock waits.
- Perform **Step 4** to analyze a recent deadlock.
- Perform **Step 5** to analyze full deadlocks.

**Step 2** Analyze and process metadata locks.

NOTE

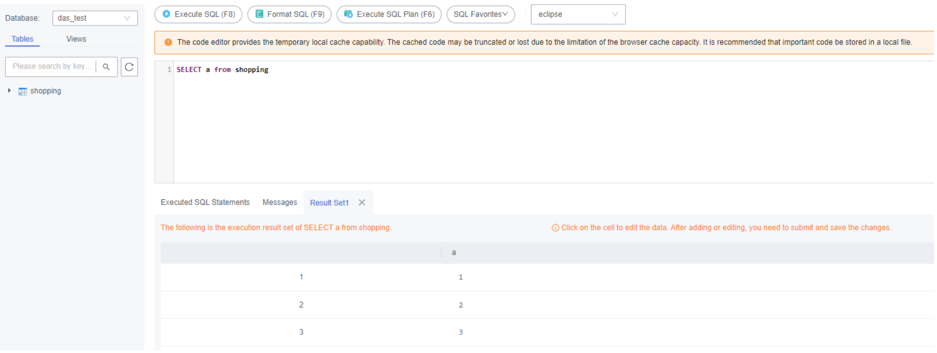
- Metadata locks are used to ensure consistency between DDL and DML operations. Executing DDL operations on a table generates metadata write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a backlog of connections.
- Metadata locks are displayed in real time, so you can quickly locate and terminate sessions holding metadata locks to restore blocked operations.
- This function is unavailable for DML locks. You can view and analyze them on the **InnoDB Locks** page.
- Up to 1,000 records can be displayed.

1. **Log in to the Huawei Cloud DB instance** and execute a SQL statement to create session 1.
   a. Query data in the **shopping** table.

      **select** *a* **from** *shopping*;

      The following result is displayed:

      **Figure 4-1** Querying data in the test table



The code editor provides the temporary local cache capability. The cached code may be truncated or lost due to the limitation of the browser cache capacity. It is recommended that important code be stored in a local file.

```
SELECT a from shopping
```

The following is the execution result set of SELECT a from shopping.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

| | a |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

   b. Start a transaction and update data in the **shopping** table. Do not submit the transaction.

      **BEGIN;**

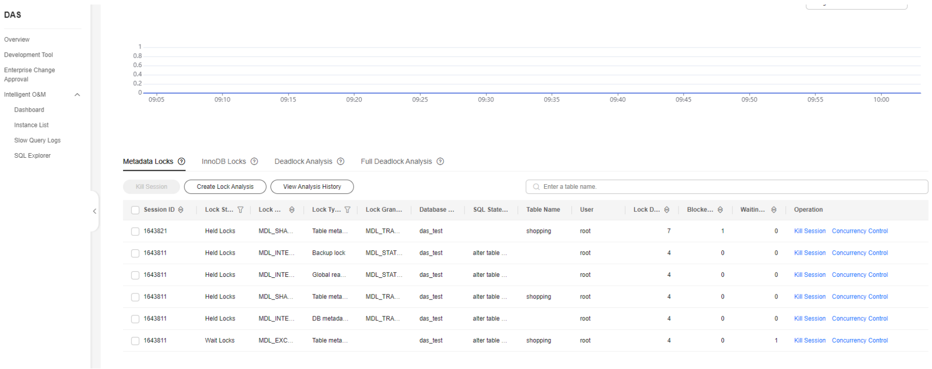
      **UPDATE** *shopping* **SET** *b = 3* **WHERE** *a =1*;

2. Create session 2 and execute the following statement to add an index to the **shopping** table:

   **ALTER TABLE** *shopping* **ADD INDEX** *idx_name(b)*;

3. Go to the DAS homepage. In the navigation pane, choose **Intelligent O&M** > **Instance List**.
4. Locate the target instance, click **Details**. Click the **Locks and Transactions**, **Locks**, and **Metadata Locks** tabs in sequence. Metadata locks of the current instance are displayed.

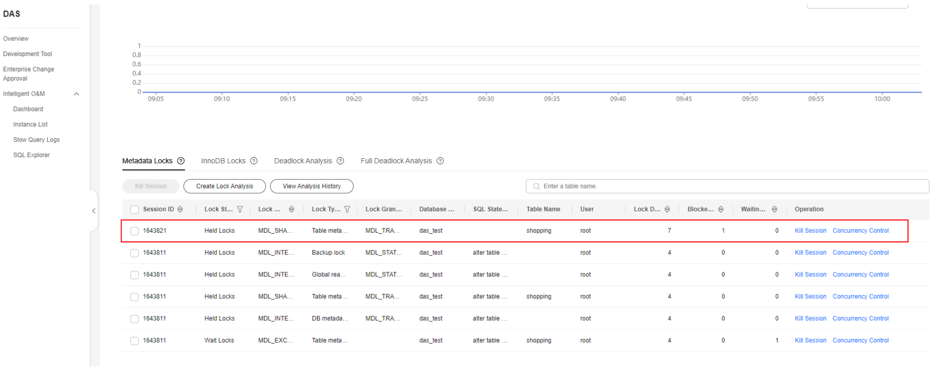
**Figure 4-2** Metadata locks



| □ | Session ID | Lock St... | Lock ... | Lock Ty... | Lock Gran... | Database ... | SQL State... | Table Name | User | Lock D... | Blocke... | Waitin... | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1643821 | Held Locks | MDL_SHA... | Table meta... | MDL_TRA... | das_test | | shopping | root | 7 | 1 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | Backup lock | MDL_STAT... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | Global rea... | MDL_STAT... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_SHA... | Table meta... | MDL_TRA... | das_test | alter table ... | shopping | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | DB metada... | MDL_TRA... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Wait Locks | MDL_EXC... | Table meta... | | das_test | alter table ... | shopping | root | 4 | 0 | 1 | Kill Session Concurrency Control |

5. Select the target session and click **Kill Session**.

**Figure 4-3** Selecting the target metadata lock



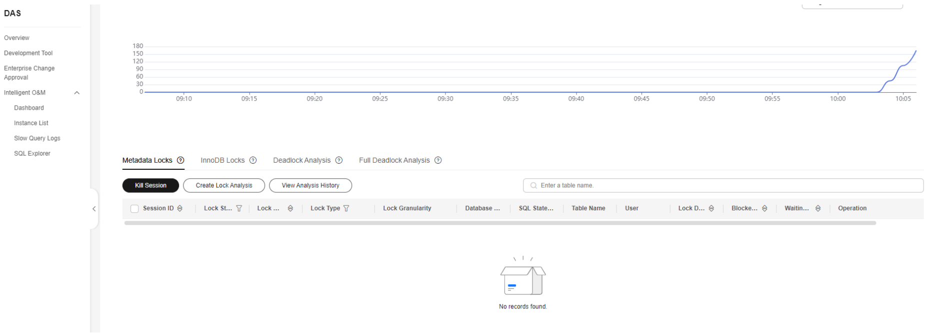
| □ | Session ID | Lock St... | Lock ... | Lock Ty... | Lock Gran... | Database ... | SQL State... | Table Name | User | Lock D... | Blocke... | Waitin... | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1643821 | Held Locks | MDL_SHA... | Table meta... | MDL_TRA... | das_test | | shopping | root | 7 | 1 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | Backup lock | MDL_STAT... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | Global rea... | MDL_STAT... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_SHA... | Table meta... | MDL_TRA... | das_test | alter table ... | shopping | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Held Locks | MDL_INTE... | DB metada... | MDL_TRA... | das_test | alter table ... | | root | 4 | 0 | 0 | Kill Session Concurrency Control |
| □ | 1643811 | Wait Locks | MDL_EXC... | Table meta... | | das_test | alter table ... | shopping | root | 4 | 0 | 1 | Kill Session Concurrency Control |

6. Refresh the metadata lock list. The query statement has been executed, and the DDL statement is being executed.

NOTE

If the **shopping** table contains a small amount of data, the DDL statement will be executed successfully immediately after the session is killed.

**Figure 4-4** Metadata lock list updated after the session is killed



| □ | Session ID | Lock St... | Lock ... | Lock Type | Lock Granularity | Database ... | SQL State... | Table Name | User | Lock D... | Blocke... | Waitin... | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

No records found

**Step 3** Analyze InnoDB lock waits.

NOTE

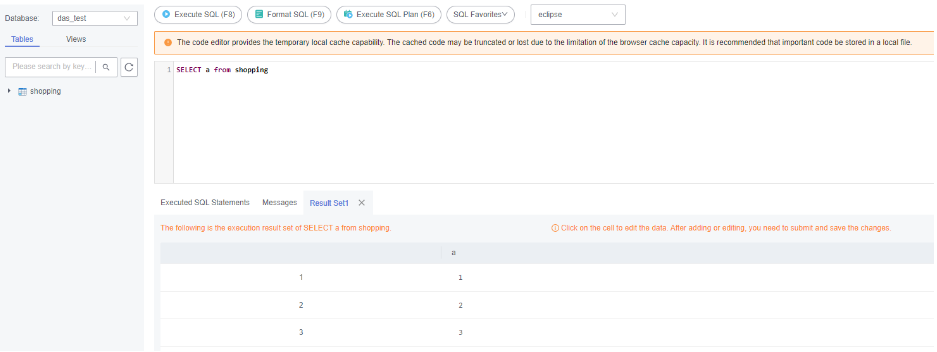
- InnoDB lock waits generated before DML operations are displayed in real time. You can quickly locate the session waits and any blocks that happened when multiple sessions update the same piece of data at the same time. You can also terminate the source session that holds locks to restore blocked operations.
- This function is unavailable for DDL locks. You can view and analyze them on the **Metadata Locks** page.

1. **Log in to the Huawei Cloud DB instance** and execute a SQL statement to create session 1.
   a. Query data in the **shopping** table.

      **select** *a* **from** *shopping*;

      The following result is displayed:

      **Figure 4-5** Querying data in the test table



The code editor provides the temporary local cache capability. The cached code may be truncated or lost due to the limitation of the browser cache capacity. It is recommended that important code be stored in a local file.

```
SELECT a from shopping
```

The following is the execution result set of SELECT a from shopping.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

| | a |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

   b. Start a transaction and update data in the **shopping** table. Do not submit the transaction.

      **BEGIN;**

      **UPDATE** *shopping* **SET** *b = 100* **WHERE** *a < 5*;

2. Create session 2 and update the statement that has been updated in session 1:

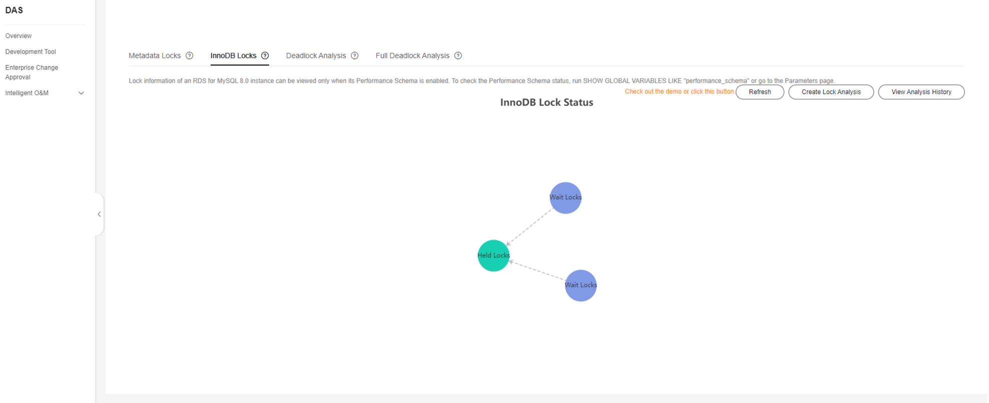
   **UPDATE** *shopping* **SET** *b = 3* **WHERE** *a = 1*;

3. Create session 3 and update the statement that has been updated in session 1:

   **UPDATE** *shopping* **SET** *b = 4* **WHERE** *a = 2*;

4. Go to the DAS homepage. In the navigation pane, choose **Intelligent O&M** > **Instance List**.
5. Locate the target instance, click **Details**. Click the **Locks and Transactions**, **Locks**, and **InnoDB Locks** tabs in sequence. InnoDB lock waits of the current instance are displayed.

**Figure 4-6** InnoDB lock waits



Lock information of an RDS for MySQL 8.0 instance can be viewed only when its Performance Schema is enabled. To check the Performance Schema status, run SHOW GLOBAL VARIABLES LIKE "performance_schema" or go to the Parameters page.

Check out the demo or click this button.

6. In session 1, run the following command to submit the transaction:

   **COMMIT;**

7. Check the InnoDB lock wait. No lock wait is displayed on the page.

**Step 4** Analyze a recent deadlock.

NOTE

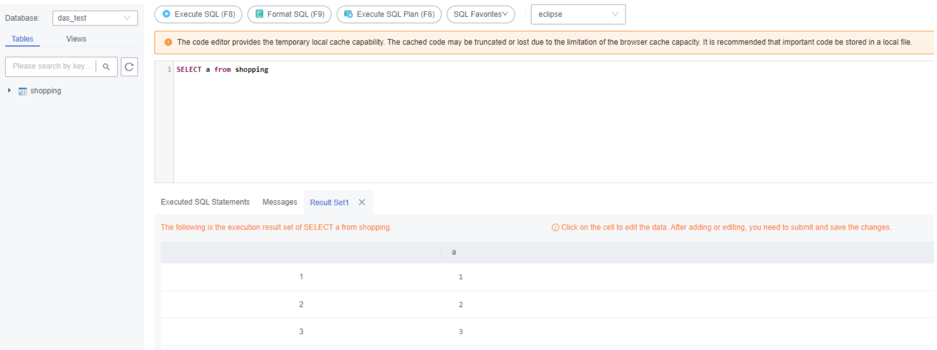
- DAS analyzes the latest deadlock log displayed in the output of **SHOW ENGINE INNODB STATUS**. If there are multiple deadlocks, only the latest one is analyzed.
- Enable **innodb_deadlock_detect** (only for RDS for MySQL 5.7).

1. **Log in to the Huawei Cloud DB instance** and execute the following SQL statement to query data in the **shopping** table.

   **select** *a* **from** *shopping***;**

   The following result is displayed:

**Figure 4-7** Querying data in the test table



The code editor provides the temporary local cache capability. The cached code may be truncated or lost due to the limitation of the browser cache capacity. It is recommended that important code be stored in a local file.

```
SELECT a from shopping
```

The following is the execution result set of SELECT a from shopping.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

| | a |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

2. Use the SQL Query module to query sessions 1 and 2. Simulate a deadlock.

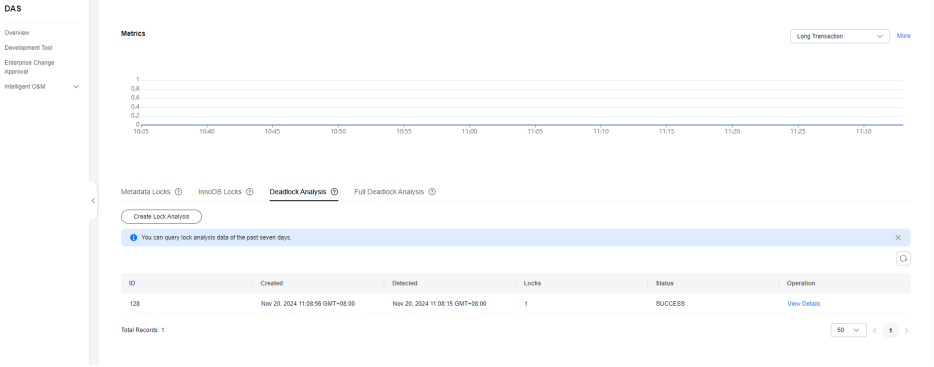
**Table 4-1** Simulating a deadlock

| Session 1 | Session 2 |
|---|---|
| begin; | begin; |
| insert into shopping(b,c) values(1,8); | - |

| Session 1 | Session 2 |
|---|---|
| - | insert into shopping(b,c) values(1,8); |
| insert into shopping(b,c) values(1,6); | - |
| - | A deadlock is generated. |

3. Go to the DAS homepage. In the navigation pane, choose **Intelligent O&M** > **Instance List**.
4. Locate the target instance, click **Details**. Click the **Locks and Transactions**, **Locks**, and **Deadlock Analysis** tabs in sequence. Click **Create Lock Analysis**, refresh the page, and check the list.

**Figure 4-8** Recent deadlock



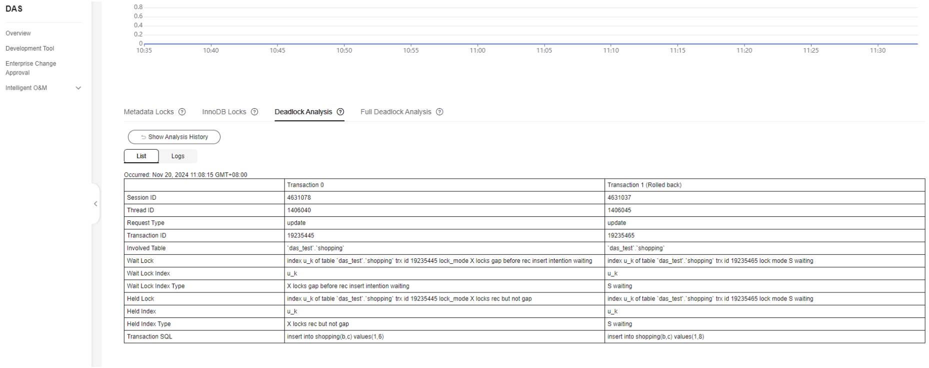
You can query lock analysis data of the past seven days.

| ID | Created | Detected | Locks | Status | Operation |
|---|---|---|---|---|---|
| 128 | Nov 20, 2024 11:08:56 GMT+08:00 | Nov 20, 2024 11:08:15 GMT+08:00 | 1 | SUCCESS | View Details |

Total Records: 1

5. Click **View Details** in the **Operation** column to view parsed deadlocks and original logs.

**Figure 4-9** Details of a recent deadlock



Occurred: Nov 20, 2024 11:08:15 GMT+08:00

| | Transaction 0 | Transaction 1 (Rolled back) |
|---|---|---|
| Session ID | 4631078 | 4631037 |
| Thread ID | 1406040 | 1406045 |
| Request Type | update | update |
| Transaction ID | 19235445 | 19235465 |
| Involved Table | `das_test`.`shopping` | `das_test`.`shopping` |
| Wait Lock | index u_k of table `das_test`.`shopping` trx id 19235445 lock_mode X locks gap before rec insert intention waiting | index u_k of table `das_test`.`shopping` trx id 19235465 lock mode S waiting |
| Wait Lock Index | u_k | u_k |
| Wait Lock Index Type | X locks gap before rec insert intention waiting | S waiting |
| Held Lock | index u_k of table `das_test`.`shopping` trx id 19235445 lock_mode X locks rec but not gap | index u_k of table `das_test`.`shopping` trx id 19235465 lock mode S waiting |
| Held Index | u_k | u_k |
| Held Index Type | X locks rec but not gap | S waiting |
| Transaction SQL | insert into shopping(b,c) values(1,6) | insert into shopping(b,c) values(1,8) |

**Step 5** Analyze full deadlocks.

NOTE

- DAS analyzes error logs at regular intervals, parses deadlock information, and performs comprehensive deadlock analysis.
- Dependency parameters:
  - Enable **innodb_deadlock_detect** (only for RDS for MySQL 5.7).
  - Enable **innodb_print_all_deadlocks** and set **log_error_verbosity** (only for RDS for MySQL of versions other than 5.7) to **3**.
- Up to 10,000 records can be displayed.

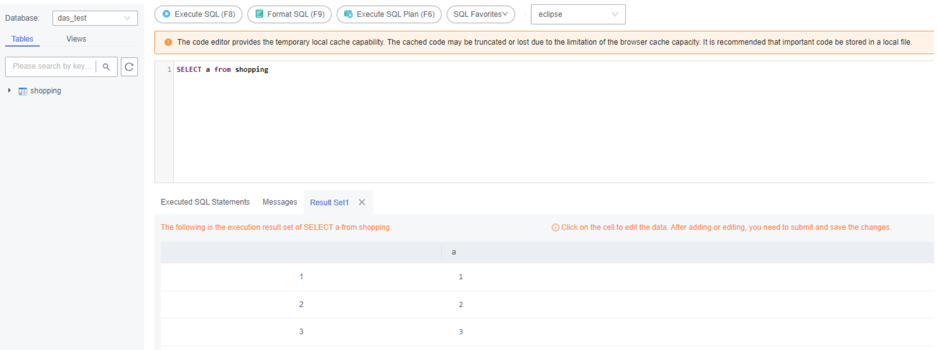
1. **Log in to the console**.
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner. Choose **Databases** > **Data Admin Service**.
4. In the navigation pane, choose **Intelligent O&M** > **Instance List**.
5. Locate the target instance, click **Details**. Click the **Locks and Transactions**, **Locks**, and **Full Deadlock Analysis** tabs in sequence. Enable **Full Deadlock Analysis**.
6. In the navigation pane, choose **Development Tool**.

   You can also click **Go to Development Tool** on the overview page.
7. Select the target database, click **Log In** in the **Operation** column, and run the following SQL statement to query data in the **shopping** table.

   **select** *a* **from** *shopping*;

   The following result is displayed:

   **Figure 4-10** Querying data in the test table



Database: das_test

```
SELECT a from shopping
```

The code editor provides the temporary local cache capability. The cached code may be truncated or lost due to the limitation of the browser cache capacity. It is recommended that important code be stored in a local file.

The following is the execution result set of SELECT a from shopping.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

| | a |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

8. Use the SQL Query module to query sessions 1 and 2. Simulate a deadlock.

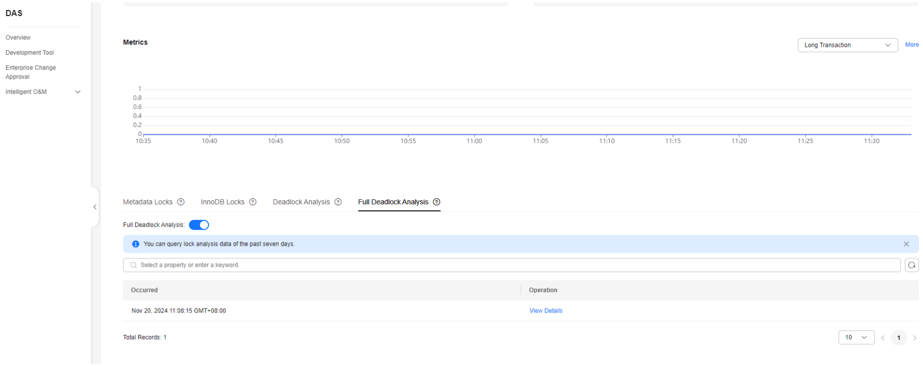
**Table 4-2** Simulating a deadlock

| Session 1 | Session 2 |
|---|---|
| begin; | begin; |
| insert into shopping(b,c) values(1,8); | - |
| - | insert into shopping(b,c) values(1,8); |
| insert into shopping(b,c) values(1,6); | - |

| Session 1 | Session 2 |
|---|---|
| - | A deadlock is generated. |

9. View the full deadlock analysis list.

**Figure 4-11** Full deadlock analysis



Full Deadlock Analysis:

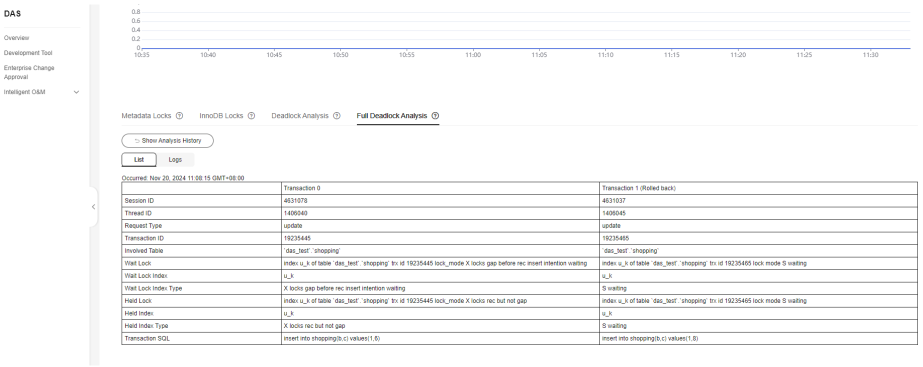
You can query lock analysis data of the past seven days.

| Occurred | Operation |
|---|---|
| Nov 20, 2024 11:08:15 GMT+08:00 | View Details |

Total Records: 1

10. Click **View Details** in the **Operation** column to view parsed deadlocks and original logs.

**Figure 4-12** Full deadlock details



Occurred: Nov 20, 2024 11:08:15 GMT+08:00

| | Transaction 0 | Transaction 1 (Rolled back) |
|---|---|---|
| Session ID | 4631078 | 4631037 |
| Thread ID | 1406040 | 1406045 |
| Request Type | update | update |
| Transaction ID | 19235445 | 19235465 |
| Involved Table | \`das_test\`.\`shopping\` | \`das_test\`.\`shopping\` |
| Wait Lock | index u_k of table \`das_test\`.\`shopping\` trx id 19235445 lock_mode X locks gap before rec insert intention waiting | index u_k of table \`das_test\`.\`shopping\` trx id 19235465 lock mode S waiting |
| Wait Lock Index | u_k | u_k |
| Wait Lock Index Type | X locks gap before rec insert intention waiting | S waiting |
| Held Lock | index u_k of table \`das_test\`.\`shopping\` trx id 19235445 lock_mode X locks rec but not gap | index u_k of table \`das_test\`.\`shopping\` trx id 19235465 lock mode S waiting |
| Held Index | u_k | u_k |
| Held Index Type | X locks rec but not gap | S waiting |
| Transaction SQL | insert into shopping(b,c) values(1,6) | insert into shopping(b,c) values(1,8) |

**----End**

# 5 Fixing High CPU Usage on DAS

High CPU usage is caused by the following reasons:

- Slow query logs

  Slow SQL queries are inefficient because a large amount of data causes high I/O usage and low QPS. You can view the CPU usage, QPS, and row read rate on the DAS performance page. To solve the issues, you can kill sessions or optimize indexes through SQL diagnosis.

- High-concurrent operations

  Increased QPS means a DB instance has to process a large number of concurrent operations, so the CPU usage increases too. On the DAS performance page, you can view metrics such as QPS, number of active connections, and CPU usage. To solve the issues, you can use functions such as SQL statement concurrency control and auto flow control or kill sessions. After core services are restored, you can evaluate whether the issues were caused by service anomaly and adjust the services if necessary. You are advised to upgrade instance specifications if they cannot meet increasing service demands.

## Solution (MySQL as an Example)

After receiving an alarm indicating that CPU usage of a Huawei Cloud DB instance increases sharply, the SRE will:

**Step 1** **Log in to the Huawei Cloud console**.

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

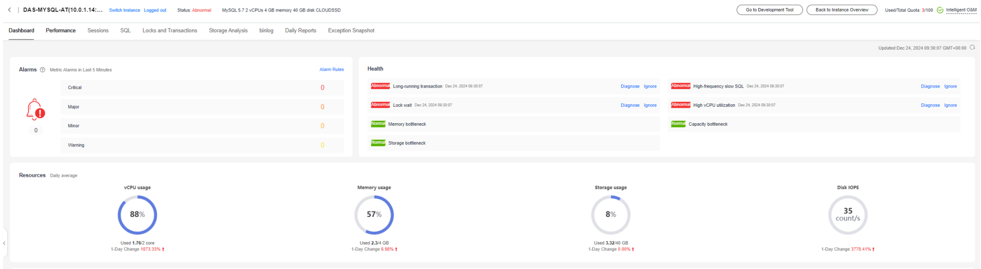
**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.

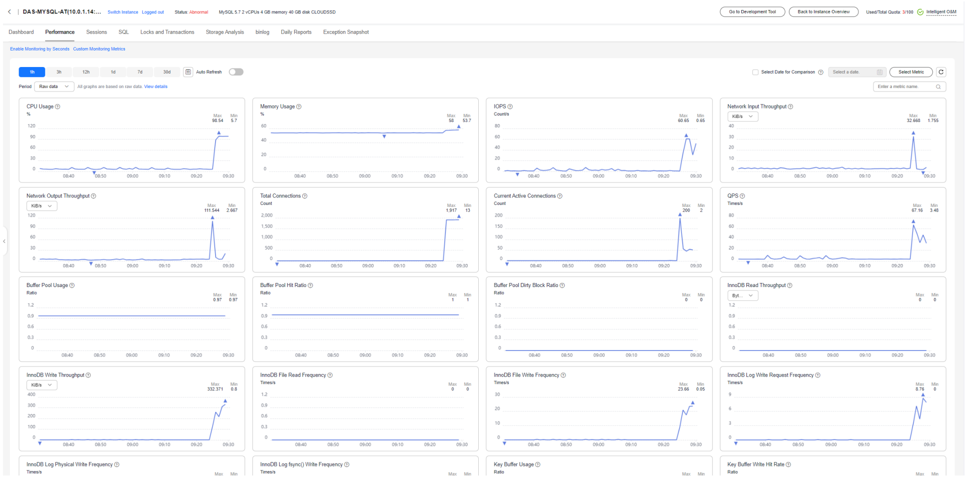
**Step 5** In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details** to go to the **Dashboard** tab page.

**Figure 5-1** Dashboard



**Step 6** Click the **Performance** tab.

**Figure 5-2** Performance monitoring



**Step 7** Check key metrics, such as current active connections, total connections, QPS, slow query logs, and CPU usage.

**Figure 5-3** Current Active Connections

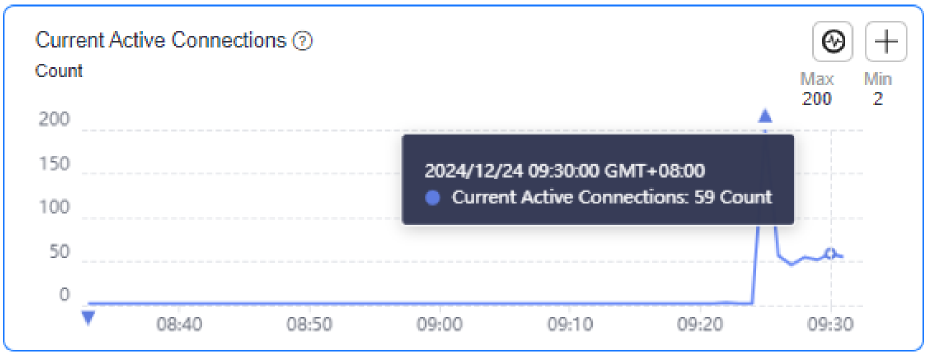
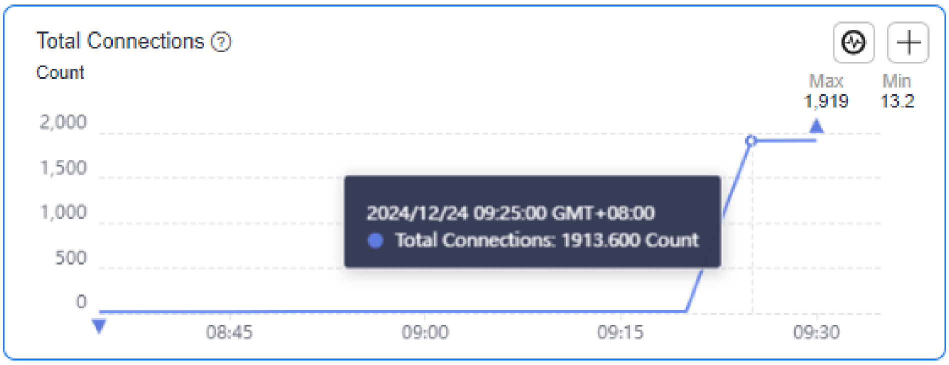
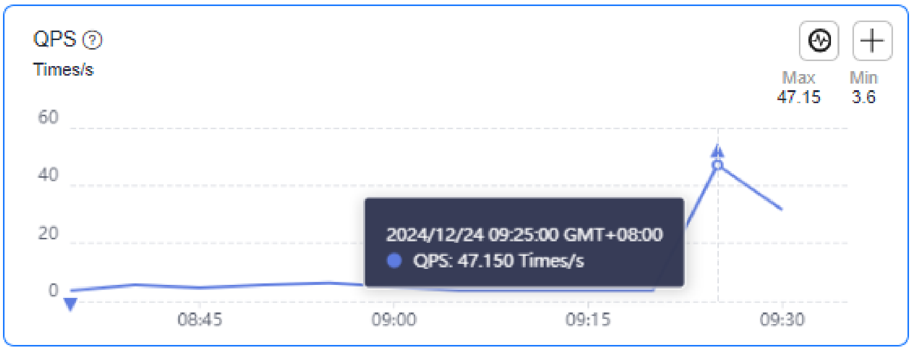


**Figure 5-4** Total Connections



2024/12/24 09:25:00 GMT+08:00
● Total Connections: 1913.600 Count

**Figure 5-5** QPS



2024/12/24 09:25:00 GMT+08:00
● QPS: 47.150 Times/s

**Figure 5-6** Slow Query Logs

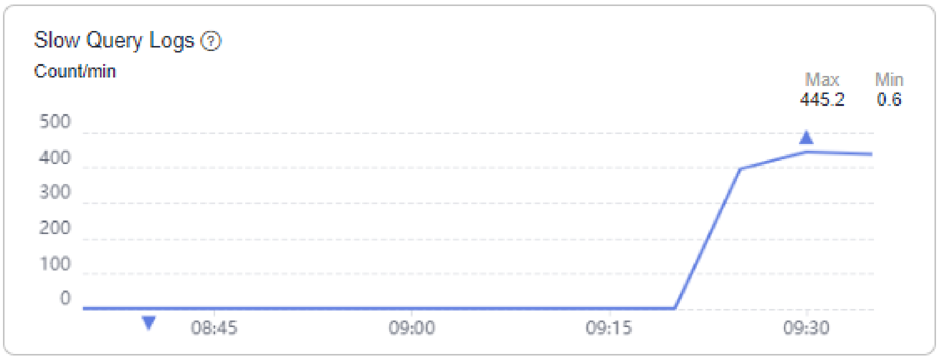
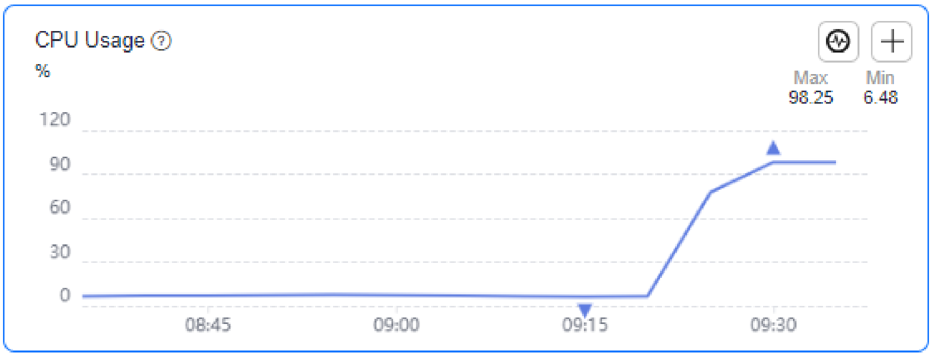


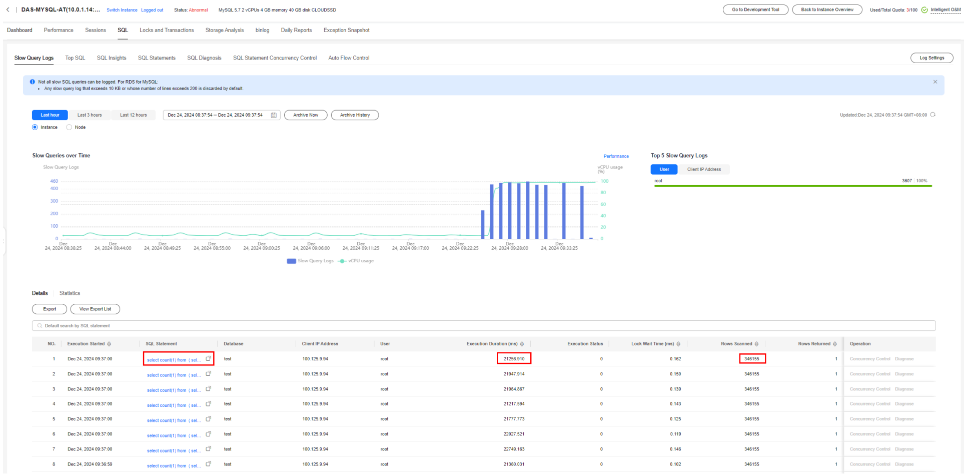
**Figure 5-7** CPU Usage



**Step 8** Verify the CPU usage increases sharply due to a sudden surge of slow query logs and high-concurrent connections.

**Step 9** Click the **SQL** tab and then **Slow Query Logs** to view details.

**Figure 5-8** Slow Query Logs



Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

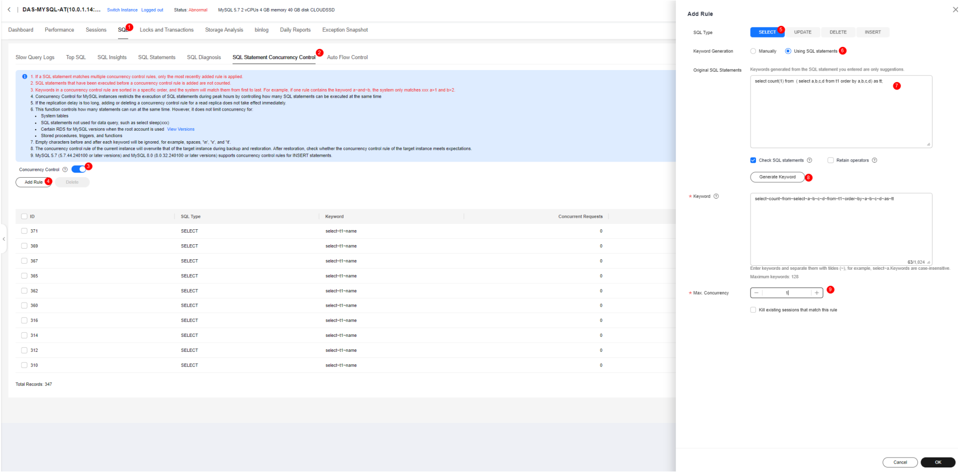
| NO. | Execution Started | SQL Statement | Database | Client IP Address | User | Execution Duration (ms) | Execution Status | Lock Wait Time (ms) | Rows Scanned | Rows Returned | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 21256.910 | 0 | 0.162 | 346155 | 1 | Concurrency Control Diagnose |
| 2 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 21947.914 | 0 | 0.150 | 346155 | 1 | Concurrency Control Diagnose |
| 3 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 21964.867 | 0 | 0.139 | 346155 | 1 | Concurrency Control Diagnose |
| 4 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 21217.594 | 0 | 0.143 | 346155 | 1 | Concurrency Control Diagnose |
| 5 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 21777.773 | 0 | 0.125 | 346155 | 1 | Concurrency Control Diagnose |
| 6 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 22027.521 | 0 | 0.119 | 346155 | 1 | Concurrency Control Diagnose |
| 7 | Dec 24, 2024 09:37:00 | select count(1) from （sel... | test | 100.125.9.94 | root | 22749.163 | 0 | 0.146 | 346155 | 1 | Concurrency Control Diagnose |
| 8 | Dec 24, 2024 09:36:59 | select count(1) from （sel... | test | 100.125.9.94 | root | 21360.031 | 0 | 0.102 | 346155 | 1 | Concurrency Control Diagnose |

A large number of same slow query logs are sent in a short period of time. Executing a single SQL statement takes about 1 minute. The SRE and service department decide to restrict this type of SQL statement and restore other core services of the database.

**Step 10** Click the **SQL Statement Concurrency Control** tab and enable **Concurrency Control**.

**Step 11** Create a SQL concurrency control rule, select **SELECT** for **SQL Type** and **Using SQL statements** for **Keyword Generation**, enter the original SQL statement, click **Generate Keyword**, set **Max. Concurrency** to **1**, and kill the existing sessions that meet the rule.

**Figure 5-9** SQL type



1. If a SQL statement matches multiple concurrency control rules, only the most recently added rule is applied.
2. SQL statements that have been executed before a concurrency control rule is added are not counted.
3. 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.
4. Concurrency Control for MySQL instances restricts the execution of SQL statements during peak hours by controlling how many SQL statements can be executed at the same time
5. If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.
6. This function controls how many statements can run at the same time. However, it does not limit concurrency for:
   - System tables
   - SQL statements not used for data query, such as select sleep(xxx)
   - Certain RDS for MySQL versions when the root account is used View Versions
   - Stored procedures, triggers, and functions
7. Empty characters before and after each keyword will be ignored, for example, spaces, '\n', '\r', and '\t'.
8. The concurrency control rule of the current instance will overwrite that of the target instance during backup and restoration. After restoration, check whether the concurrency control rule of the target instance meets expectations.
9. MySQL 5.7 (5.7.44.240100 or later versions) and MySQL 8.0 (8.0.32.240100 or later versions) supports concurrency control rules for INSERT statements.

| ID | SQL Type | Keyword | Concurrent Requests |
|---|---|---|---|
| 371 | SELECT | select~t1~name | 0 |
| 369 | SELECT | select~t1~name | 0 |
| 367 | SELECT | select~t1~name | 0 |
| 365 | SELECT | select~t1~name | 0 |
| 362 | SELECT | select~t1~name | 0 |
| 360 | SELECT | select~t1~name | 0 |
| 316 | SELECT | select~t1~name | 0 |
| 314 | SELECT | select~t1~name | 0 |
| 312 | SELECT | select~t1~name | 0 |
| 310 | SELECT | select~t1~name | 0 |

Total Records: 347

SQL Type: SELECT

Keyword Generation: ○ Manually ● Using SQL statements

Original SQL Statements: Keywords generated from the SQL statement you entered are only suggestions.

```
select count(1) from （select a,b,c,d from t1 order by a,b,c,d) as tt;
```

☑ Check SQL statements ☐ Retain operators

* Keyword:

```
select~count~from~select~a~b~c~d~from~t1~order~by~a~b~c~d~as~tt
```

63/1,024

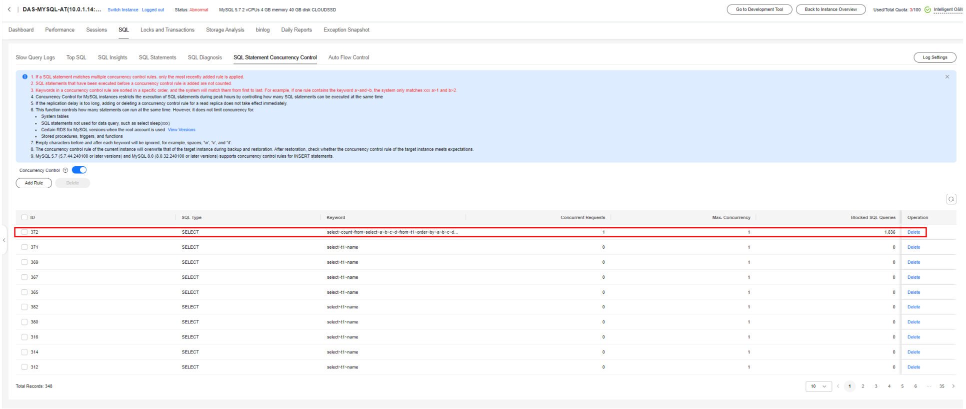
Enter keywords and separate them with tildes (~), for example, select~a.Keywords are case-insensitive.

Maximum keywords: 128

* Max. Concurrency: 1

☐ Kill existing sessions that match this rule

**Figure 5-10** Selecting a SQL statement concurrency control rule



DAS-MYSQL-AT(10.0.1.14:... Switch Instance Logged out Status: Abnormal MySQL 5.7 2 vCPUs 4 GB memory 40 GB disk CLOUDSSD

1. If a SQL statement matches multiple concurrency control rules, only the most recently added rule is applied.
2. SQL statements that have been executed before a concurrency control rule is added are not counted.
3. 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.
4. Concurrency Control for MySQL instances restricts the execution of SQL statements during peak hours by controlling how many SQL statements can be executed at the same time
5. If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.
6. This function controls how many statements can run at the same time. However, it does not limit concurrency for:
   - System tables
   - SQL statements not used for data query, such as select sleep(xxx)
   - Certain RDS for MySQL versions when the root account is used View Versions
   - Stored procedures, triggers, and functions
7. Empty characters before and after each keyword will be ignored, for example, spaces, '\n', '\r', and '\t'.
8. The concurrency control rule of the current instance will overwrite that of the target instance during backup and restoration. After restoration, check whether the concurrency control rule of the target instance meets expectations.
9. MySQL 5.7 (5.7.44.240100 or later versions) and MySQL 8.0 (8.0.32.240100 or later versions) supports concurrency control rules for INSERT statements.

Concurrency Control

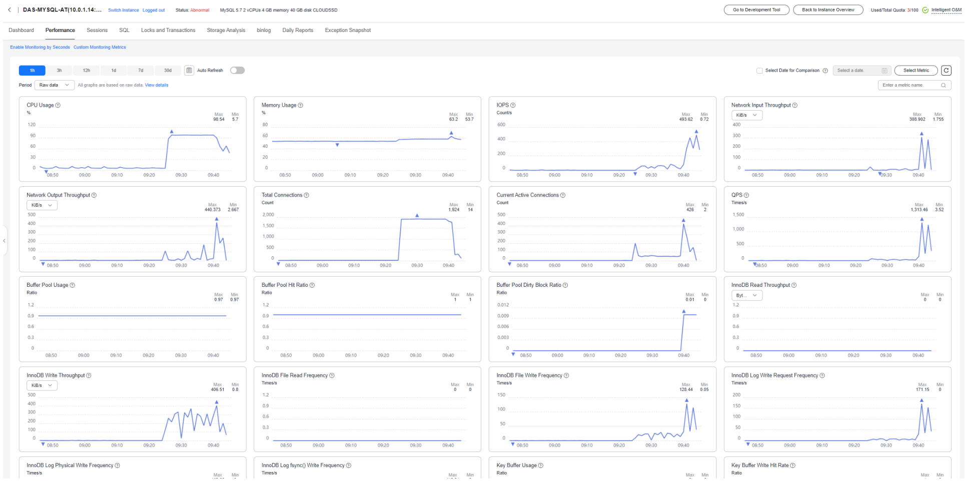
| ID | SQL Type | Keyword | Concurrent Requests | Max. Concurrency | Blocked SQL Queries | Operation |
|---|---|---|---|---|---|---|
| 372 | SELECT | select~count~from~select~a~b~c~d~from~t1~order~by~a~b~c~d... | 1 | 1 | 1,836 | Delete |
| 371 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 369 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 367 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 365 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 362 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 360 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 316 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 314 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |
| 312 | SELECT | select~t1~name | 0 | 1 | 0 | Delete |

Total Records: 348

The SQL statement concurrency control rule has taken effect and the SQL statements that match the rule are intercepted.

**Step 12** Click the **Performance** tab and check the database performance metrics. The CPU usage starts to decrease and the service department reports that core services have recovered.

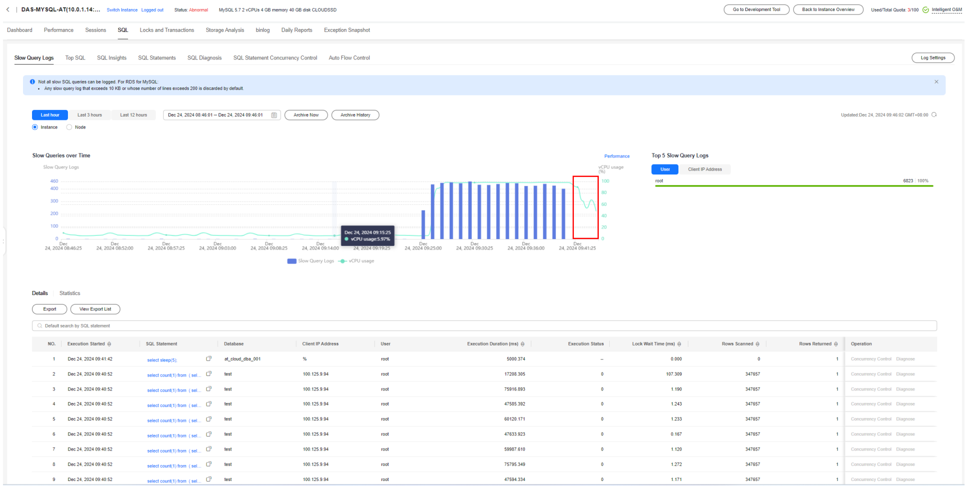
**Figure 5-11** Performance metrics



DAS-MYSQL-AT(10.0.1.14:... Switch Instance Logged out Status: Abnormal MySQL 5.7 2 vCPUs 4 GB memory 40 GB disk CLOUDSSD
Go to Development Tool
Back to Instance Overview
Used/Total Quota: 3/100
Intelligent O&M
Dashboard
Performance
Sessions
SQL
Locks and Transactions
Storage Analysis
binlog
Daily Reports
Exception Snapshot
Enable Monitoring by Seconds
Custom Monitoring Metrics
1h 3h 12h 1d 7d 30d
Auto Refresh
Period Raw data
All graphs are based on raw data. View details
Select Date for Comparison
Select a date.
Select Metric
Enter a metric name.
CPU Usage % Max 98.54 Min 5.7
Memory Usage % Max 63.2 Min 53.7
IOPS Count/s Max 493.62 Min 0.72
Network Input Throughput KiB/s Max 308.902 Min 1.755
Network Output Throughput KiB/s Max 440.373 Min 2.667
Total Connections Count Max 1,924 Min 14
Current Active Connections Count Max 426 Min 2
QPS Times/s Max 1,313.46 Min 3.52
Buffer Pool Usage Ratio Max 0.97 Min 0.97
Buffer Pool Hit Ratio Ratio Max 1 Min 1
Buffer Pool Dirty Block Ratio Ratio Max 0.01 Min 0
InnoDB Read Throughput Byt... Max 0 Min 0
InnoDB Write Throughput KiB/s Max 406.51 Min 0.8
InnoDB File Read Frequency Times/s Max 0 Min 0
InnoDB File Write Frequency Times/s Max 128.44 Min 0.05
InnoDB Log Write Request Frequency Times/s Max 171.15 Min 0
InnoDB Log Physical Write Frequency Times/s
InnoDB Log fsync() Write Frequency Times/s
Key Buffer Usage Ratio
Key Buffer Write Hit Rate Ratio
08:50 09:00 09:10 09:20 09:30 09:40

After the SQL statement concurrency control takes effect, no same slow query is logged.

**Figure 5-12** Slow queries over time



DAS-MYSQL-AT(10.0.1.14:... Switch Instance Logged out Status: Abnormal MySQL 5.7 2 vCPUs 4 GB memory 40 GB disk CLOUDSSD

Go to Development Tool | Back to Instance Overview | Used/Total Quota: 3/100 | Intelligent O&M

Dashboard | Performance | Sessions | SQL | Locks and Transactions | Storage Analysis | binlog | Daily Reports | Exception Snapshot

Slow Query Logs | Top SQL | SQL Insights | SQL Statements | SQL Diagnosis | SQL Statement Concurrency Control | Auto Flow Control | Log Settings

Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

Last hour | Last 3 hours | Last 12 hours | Dec 24, 2024 08:46:01 – Dec 24, 2024 09:46:01 | Archive Now | Archive History

● Instance ○ Node

Updated:Dec 24, 2024 09:46:02 GMT+08:00

Top 5 Slow Query Logs

User | Client IP Address

root 6823 | 100%

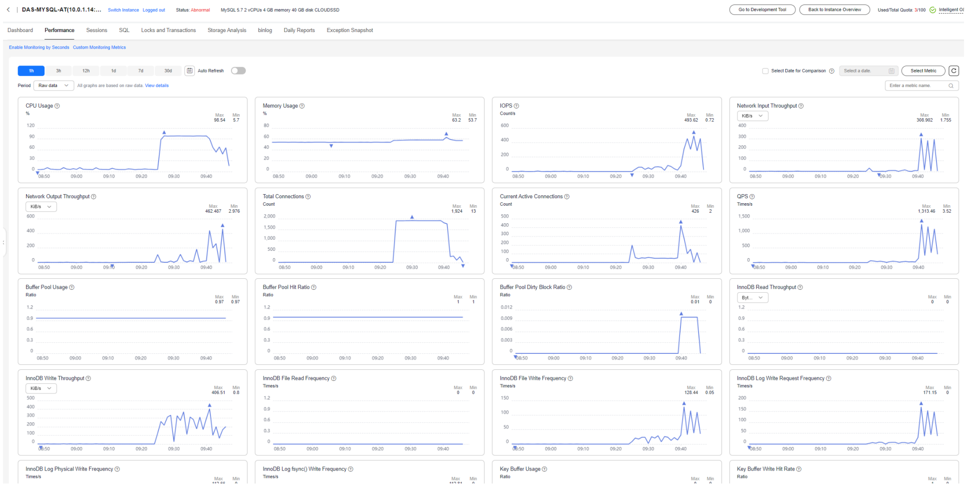
Details | Statistics

Export | View Export List

Default search by SQL statement

| NO. | Execution Started | SQL Statement | Database | Client IP Address | User | Execution Duration (ms) | Execution Status | Lock Wait Time (ms) | Rows Scanned | Rows Returned | Operation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Dec 24, 2024 09:41:42 | select sleep(5); | at_cloud_dba_001 | % | root | 5000.374 | -- | 0.000 | 0 | 1 | Concurrency Control Diagnose |
| 2 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 17208.305 | 0 | 107.309 | 347657 | 1 | Concurrency Control Diagnose |
| 3 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 75916.893 | 0 | 1.190 | 347657 | 1 | Concurrency Control Diagnose |
| 4 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 47585.392 | 0 | 1.243 | 347657 | 1 | Concurrency Control Diagnose |
| 5 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 60120.171 | 0 | 1.233 | 347657 | 1 | Concurrency Control Diagnose |
| 6 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 47633.923 | 0 | 0.167 | 347657 | 1 | Concurrency Control Diagnose |
| 7 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 59987.610 | 0 | 1.120 | 347657 | 1 | Concurrency Control Diagnose |
| 8 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 75795.349 | 0 | 1.272 | 347657 | 1 | Concurrency Control Diagnose |
| 9 | Dec 24, 2024 09:40:52 | select count(1) from ( sel... | test | 100.125.9.94 | root | 47594.334 | 0 | 1.171 | 347657 | 1 | Concurrency Control Diagnose |

**Figure 5-13** Performance metrics



DAS-MYSQL-AT(10.0.1.14:... Switch Instance Logged out Status: Abnormal MySQL 5.7 2 vCPUs 4 GB memory 40 GB disk CLOUDSSD

Go to Development Tool | Back to Instance Overview | Used/Total Quota: 3/100 | Intelligent O&M

Dashboard | Performance | Sessions | SQL | Locks and Transactions | Storage Analysis | binlog | Daily Reports | Exception Snapshot

Enable Monitoring by Seconds | Custom Monitoring Metrics

1h | 3h | 12h | 1d | 7d | 30d | Auto Refresh

□ Select Date for Comparison | Select a date. | Select Metric

Period: Raw data | All graphs are based on raw data. View details

Enter a metric name.

**----End**

# 6 Fixing Insufficient Storage on DAS

DAS intelligent O&M gives insights into space overview, disk space distribution, intelligent table diagnosis, disk space change trend, and top database tables.

This section describes how to fix insufficient storage of an RDS for MySQL instance.

## Prerequisites

You have created an RDS for MySQL instance.

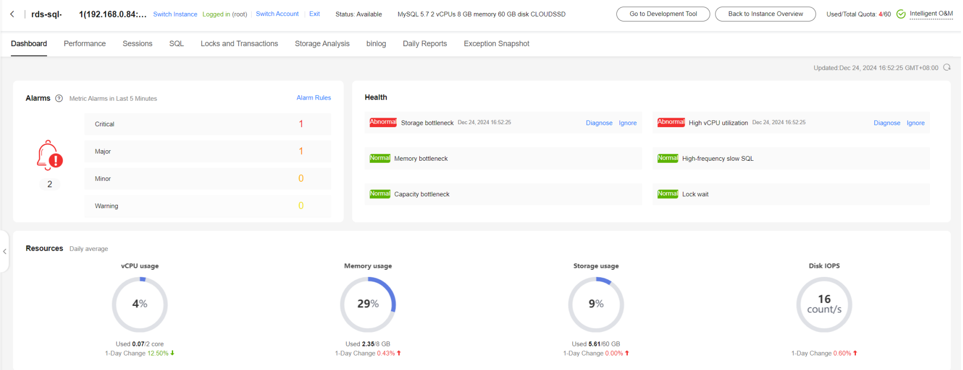
## Procedure

**Step 1** Go to the capacity analysis page.

1. Log in to the Huawei Cloud console.
2. Click in the upper left corner and select a region and project.
3. Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.
4. In the navigation pane, choose **Intelligent O&M** > **Instance List**.

   Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.
5. In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details** to go to the **Dashboard** tab page.

**Figure 6-1** Dashboard

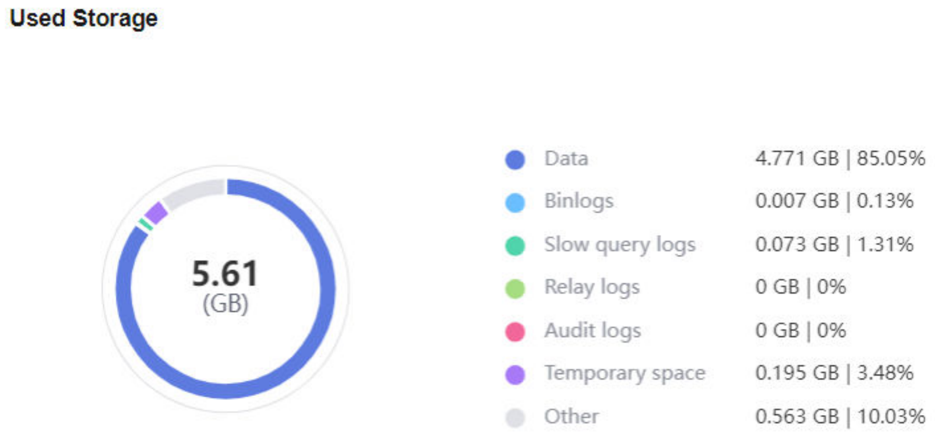


6. Click the **Storage Analysis** tab.

**Step 2** Analyze the overall disk space.

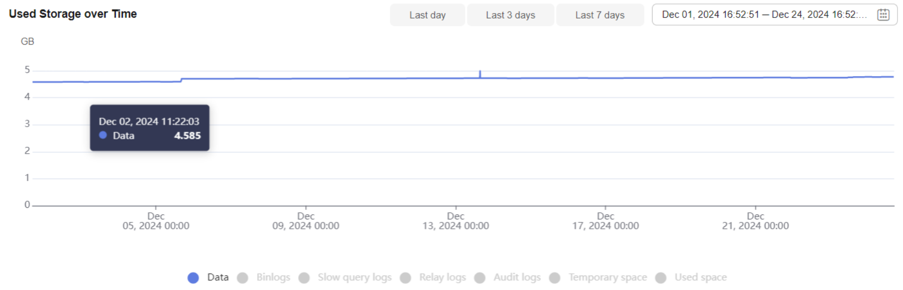
1. In the **Used Storage** area, view disk space distribution and usage.

**Figure 6-2** Disk space distribution



2. Analyze which module is taking up your disk space and check the used space changes of that module.

**Figure 6-3** Used disk space

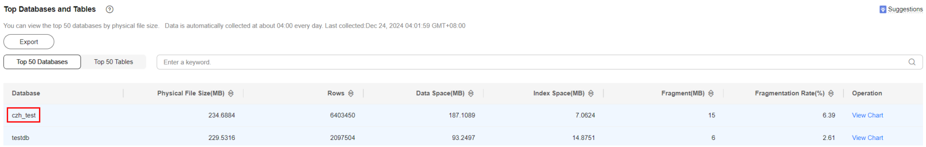


3. After locating the anomaly on a day, check operations on the instance on that day and determine whether to release the space.

**Step 3** Analyze data usage on the disk.

1. The top databases and tables help customers locate abnormal increase in data size.
2. Check the top databases or tables to find an unexpected data size.

**Figure 6-4** Top databases



Top Databases and Tables

You can view the top 50 databases by physical file size. Data is automatically collected at about 04:00 every day. Last collected:Dec 24, 2024 04:01:59 GMT+08:00

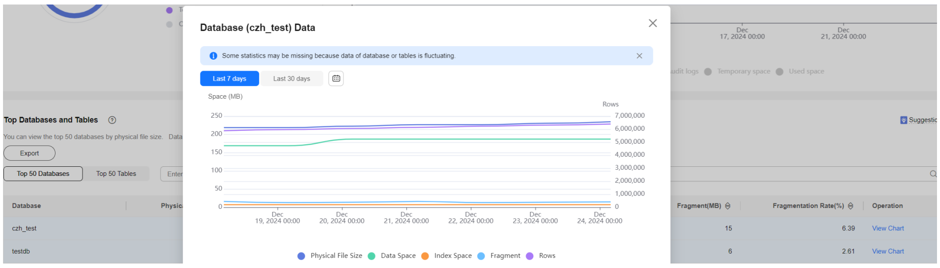
| Database | Physical File Size(MB) | Rows | Data Space(MB) | Index Space(MB) | Fragment(MB) | Fragmentation Rate(%) | Operation |
|---|---|---|---|---|---|---|---|
| czh_test | 234.6884 | 6403450 | 187.1089 | 7.0624 | 15 | 6.39 | View Chart |
| testdb | 229.5316 | 2097504 | 93.2497 | 14.8751 | 6 | 2.61 | View Chart |

3. The space usage is abnormal.

a. Delete invalid data from database tables.

b. If invalid data cannot be identified, click **View Chart** to view the space usage changes and locate the time when the exception occurred.

**Figure 6-5** Data trend



Database (czh_test) Data

Some statistics may be missing because data of database or tables is fluctuating.

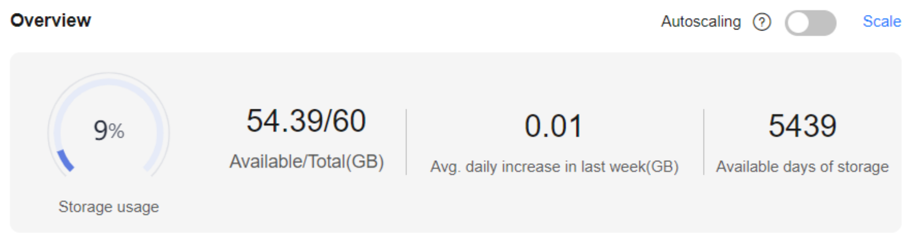
c. Obtain audit logs and SQL Explorer information of the instance at the time when the exception occurred. Delete the data which grew sharply due to abnormal changes.

4. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

**Step 4** Scale up the storage.

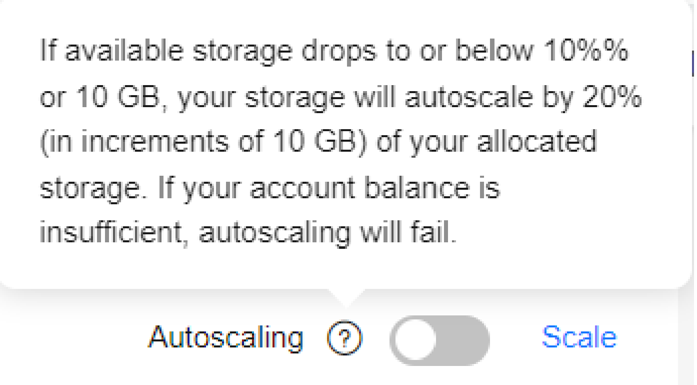
1. Check the **Overview** area. If the remaining usage is less than 10% or the available space is less than 10 GB, you are advised to click **Scale**.

**Figure 6-6** Overview



2. You can also toggle on **Autoscaling**. The instance storage will be automatically scaled up under certain conditions to ensure availability.

**Figure 6-7** Autoscaling

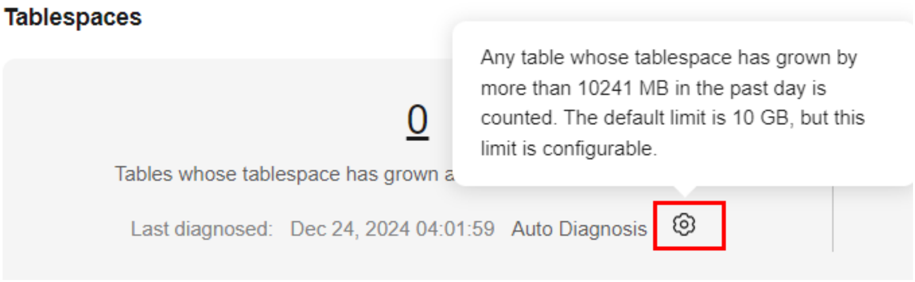


If available storage drops to or below 10%% or 10 GB, your storage will autoscale by 20% (in increments of 10 GB) of your allocated storage. If your account balance is insufficient, autoscaling will fail.

**Step 5** Configure an anomaly detection alarm.

After configuring a threshold, you can identify unexpected space usage increase on the **Tablespaces** page. When a sudden increase in data size exceeds the threshold (10 GB by default), an alarm is reported.

**Figure 6-8** Tablespaces



Tablespaces

0

Tables whose tablespace has grown a

Any table whose tablespace has grown by more than 10241 MB in the past day is counted. The default limit is 10 GB, but this limit is configurable.

Last diagnosed: Dec 24, 2024 04:01:59 Auto Diagnosis

**----End**

# 7 DAS Security Best Practices

Security is a shared responsibility between Huawei Cloud and you. Huawei Cloud is responsible for the security of cloud services to provide a secure cloud. As a tenant, you should properly use the security capabilities provided by cloud services to protect data, and securely use the cloud. For details, see **Shared Responsibilities**.

Consider the following aspects for your security configurations:

- **Properly Managing Database Accounts and Passwords to Reduce Data Leakage Risks**
- **Enabling CTS to Review Cloud Users' Operations**
- **Collecting All Query Logs**
- **Using Fine-Grained Authorization to Control the Usage Scope of DAS Resources**
- **Isolating Networks for Data Synchronization**
- **Properly Using Authentication Credentials to Prevent Data Leakage**

## Properly Managing Database Accounts and Passwords to Reduce Data Leakage Risks

1. **Change the administrator password periodically.**

   The default database administrator account **root** has elevated privileges. For enhanced security, you are advised to periodically change its password by following **Editing a Database User**.

2. **Configure password complexity.**

   As a collector of information, a database system is easy to be the target of attacks. You need to keep your database account and password secure. In addition, configure the complexity of your password to avoid using weak passwords.

3. **Configure a password expiration policy.**

   Using the same password too long makes it easier for hackers to crack or guess your password. You are advised to set a password expiration policy to limit the time for using the same password.

## Enabling CTS to Review Cloud Users' Operations

- Cloud Trace Service (CTS) records operations on cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.
- After you enable CTS and create and configure a tracker, CTS can record management events of DAS. For details, see **Enabling CTS**.
- CTS allows you to query resources from multiple dimensions, which makes it easier to review operations and locate faults.

## Collecting All Query Logs

You can enable **Collect All Query Logs** to improve security. For details, see **Enabling SQL Insights**.

1. All SQL operations are recorded to quickly locate the source of abnormal behaviors such as data leakage and mis-deletion.
2. Attacks such as SQL injection and brute-force attacks can be blocked in real time.
3. Unauthorized operations (such as high-risk commands) can be monitored to reduce internal risks.
4. Attack paths can be accurately restored to evaluate the extent of their impact.
5. The probability of malicious operations is reduced.

## Using Fine-Grained Authorization to Control the Usage Scope of DAS Resources

1. Grant least privileges to IAM users with different roles to prevent data leakage or misoperations. To better isolate and manage permissions, you are advised to configure independent IAM administrators and grant them privileges to manage IAM policies. The IAM administrator can create different user groups based on your service requirements. User groups correspond to different data access scenarios. By adding users to user groups and binding IAM policies to user groups, the IAM administrator can grant different data access privileges to employees in different departments based on the principle of least privilege.
2. You are advised to use fine-grained authorization to control user privileges. Fine-grained policies are split by API. You can **create custom privilege policies** based on privileges required for DAS operations.

## Isolating Networks for Data Synchronization

For security purposes, firewalls, ACL rules, and security groups can control database access on DAS.

## Properly Using Authentication Credentials to Prevent Data Leakage

When you use code or API Explorer to call APIs, you need to obtain a token using an account password or AK/SK. You need to comply with the secure encoding rules, properly manage authentication credentials, and do not hardcode authentication information in plaintext.

# 8 Restoring Data by Parsing Binlogs

## Application Scenarios

If you have mistakenly executed a DML statement in a database, you can parse binlogs on DAS to quickly restore data. Events related to the DML statement are recorded in binlogs in sequence. You can execute the generated event rollback statements.

## Prerequisites

- You have created an RDS for MySQL instance.
- You have enabled binlogs.

## Procedure

**Step 1** Preset data.

1. Create test database **test** in the destination RDS for MySQL instance. For details, see **Creating a Database**.
2. Log in to the RDS for MySQL database through DAS. For details, see **Logging In to a Huawei Cloud DB Instance**.
3. Run the following SQL statement to create the **shopping** table in the **test** database:

```
CREATE TABLE shopping (
a int NOT NULL AUTO_INCREMENT,
b int,
c int,
PRIMARY KEY (a),
UNIQUE KEY u_k (b, c)
);
```

4. Run the following command to write test data to the **shopping** table:

```
insert into shopping(b,c) values(1,1),(2,5),(3,9),(4,6);
```

5. Run the following command to set **b** in the **shopping** table to **10**:

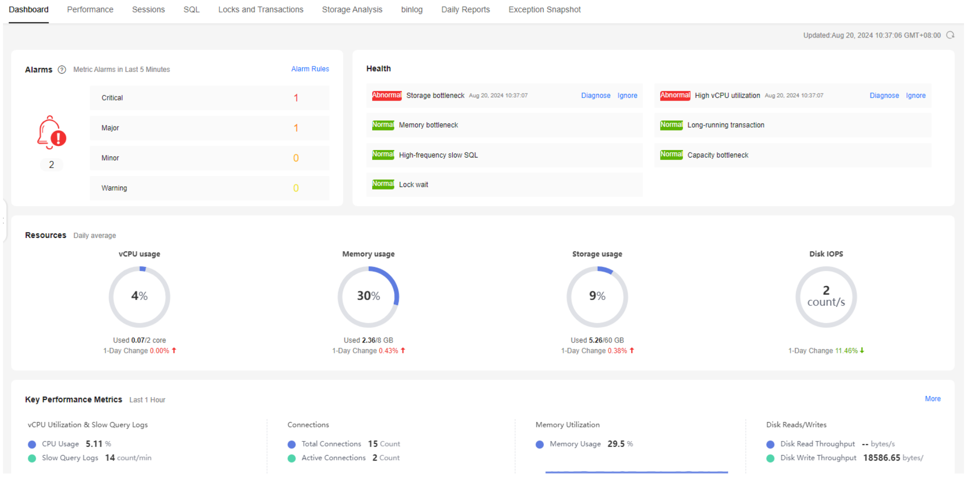
```
update shopping set b = 10;
```

**Step 2** Obtain all operation records in a specified period from parsed binlogs.

1. Log in to the console.
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner. Choose **Databases** > **Data Admin Service**.
4. In the navigation pane, choose **Intelligent O&M** > **Instance List**.

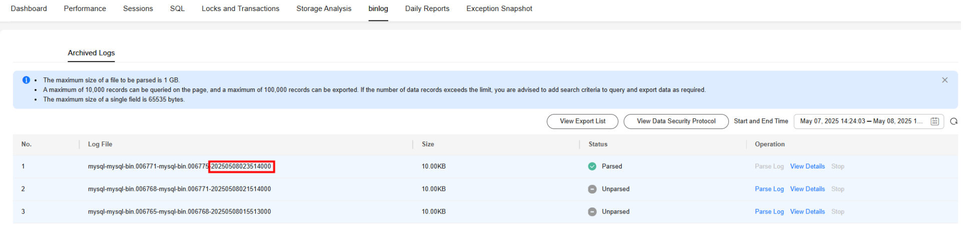
   Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.
5. In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details** to go to the **Dashboard** tab page.

**Figure 8-1** Dashboard


Dashboard
Performance
Sessions
SQL
Locks and Transactions
Storage Analysis
binlog
Daily Reports
Exception Snapshot
Updated:Aug 20, 2024 10:37:06 GMT+08:00
Alarms
Metric Alarms in Last 5 Minutes
Alarm Rules
2
Critical 1
Major 1
Minor 0
Warning 0
Health
Abnormal Storage bottleneck Aug 20, 2024 10:37:07 Diagnose Ignore
Abnormal High vCPU utilization Aug 20, 2024 10:37:07 Diagnose Ignore
Normal Memory bottleneck
Normal Long-running transaction
Normal High-frequency slow SQL
Normal Capacity bottleneck
Normal Lock wait
Resources Daily average
vCPU usage
4%
Used 0.07/2 core
1-Day Change 0.00%
Memory usage
30%
Used 2.36/8 GB
1-Day Change 0.43%
Storage usage
9%
Used 5.26/60 GB
1-Day Change 0.38%
Disk IOPS
2 count/s
1-Day Change 11.46%
Key Performance Metrics Last 1 Hour
More
vCPU Utilization & Slow Query Logs
CPU Usage 5.11 %
Slow Query Logs 14 count/min
Connections
Total Connections 15 Count
Active Connections 2 Count
Memory Utilization
Memory Usage 29.5 %
Disk Reads/Writes
Disk Read Throughput -- bytes/s
Disk Write Throughput 18586.65 bytes/

6. Click **binlog**.

**Figure 8-2** Archived logs



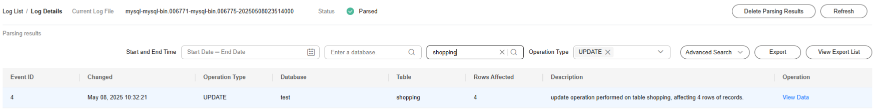
- The maximum size of a file to be parsed is 1 GB.
- A maximum of 10,000 records can be queried on the page, and a maximum of 100,000 records can be exported. If the number of data records exceeds the limit, you are advised to add search criteria to query and export data as required.
- The maximum size of a single field is 65535 bytes.

Start and End Time May 07, 2025 14:24:03 – May 08, 2025 1...

| No. | Log File | Size | Status | Operation |
|---|---|---|---|---|
| 1 | mysql-mysql-bin.006771-mysql-bin.006775-20250508023514000 | 10.00KB | Parsed | Parse Log View Details Stop |
| 2 | mysql-mysql-bin.006768-mysql-bin.006771-20250508021514000 | 10.00KB | Unparsed | Parse Log View Details Stop |
| 3 | mysql-mysql-bin.006765-mysql-bin.006768-20250508015513000 | 10.00KB | Unparsed | Parse Log View Details Stop |

7. **20250508023514000** in the file name indicates that the file was executed before 02:35 on May 8, 2025 (UTC time), which is the same as the execution time of the UPDATE statement.
8. Click **Parse Log** and wait until the parsing is complete.
9. Click **View Details**. The SQL records are displayed. You can filter tables, databases, and operation types to restore data. The **shopping** table and UPDATE are used as an example.

**Figure 8-3** Viewing log details



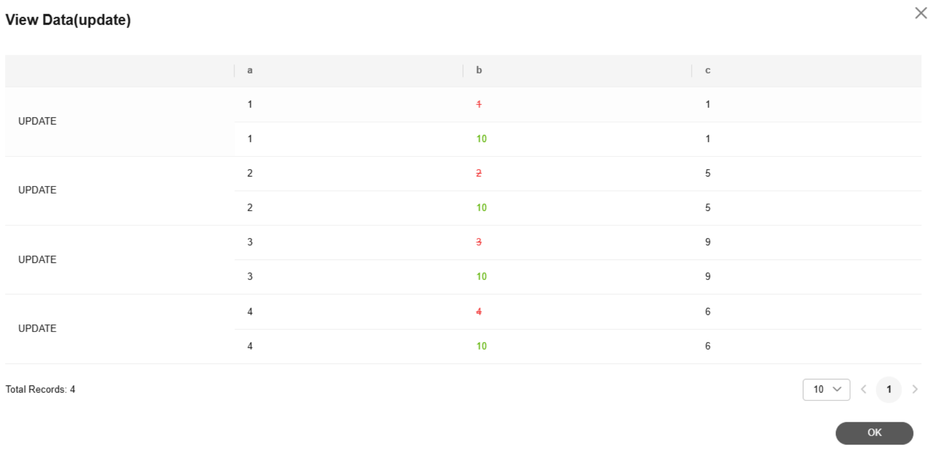
Current Log File mysql-mysql-bin.006771-mysql-bin.006775-20250508023514000 Status Parsed

Start and End Time Start Date – End Date
Enter a database.
shopping
Operation Type UPDATE

| Event ID | Changed | Operation Type | Database | Table | Rows Affected | Description | Operation |
|---|---|---|---|---|---|---|---|
| 4 | May 08, 2025 10:32:21 | UPDATE | test | shopping | 4 | update operation performed on table shopping, affecting 4 rows of records. | View Data |

10. Click **View Data** to check whether UPDATE statements are displayed.

**Figure 8-4** Viewing data



View Data(update)

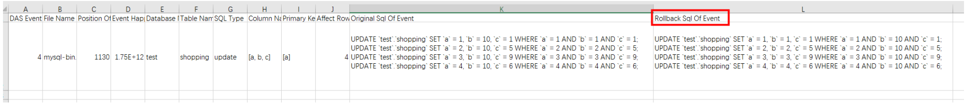
| | a | b | c |
|---|---|---|---|
| UPDATE | 1 | ~~1~~ | 1 |
| | 1 | 10 | 1 |
| UPDATE | 2 | ~~2~~ | 5 |
| | 2 | 10 | 5 |
| UPDATE | 3 | ~~3~~ | 9 |
| | 3 | 10 | 9 |
| UPDATE | 4 | ~~4~~ | 6 |
| | 4 | 10 | 6 |

Total Records: 4

**Step 3** Obtain rollback statements based on the parsed binlogs.

1. Obtain all operation records in a specified period. Filter desired results, click **Export**, and select an OBS bucket to export the results.
2. Click **View Export List** to obtain the latest exported records.
3. Select the records that match both the file name and the corresponding number from the search results. Click **Download**.
4. Obtain rollback statements from the **Rollback Sql Of Event** column in the downloaded file.

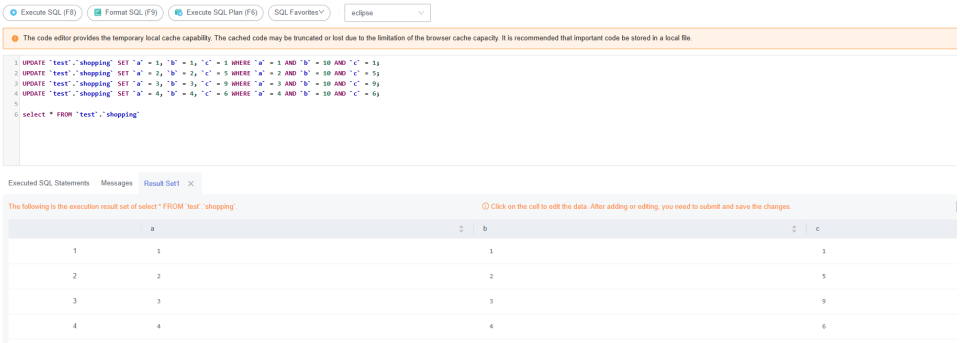
**Figure 8-5** Event rollback statements



| A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DAS Event | File Name | Position Of | Event Hap | Database I | Table Nam | SQL Type | Column Na | Primary Ke | Affect Row | Original Sql Of Event | Rollback Sql Of Event |
| 4 | mysql-bin. | 1130 | 1.75E+12 | test | shopping | update | [a, b, c] | [a] | 4 | UPDATE `test`.`shopping` SET `a` = 1, `b` = 10, `c` = 1 WHERE `a` = 1 AND `b` = 1 AND `c` = 1;<br>UPDATE `test`.`shopping` SET `a` = 2, `b` = 10, `c` = 5 WHERE `a` = 2 AND `b` = 2 AND `c` = 5;<br>UPDATE `test`.`shopping` SET `a` = 3, `b` = 10, `c` = 9 WHERE `a` = 3 AND `b` = 3 AND `c` = 9;<br>UPDATE `test`.`shopping` SET `a` = 4, `b` = 10, `c` = 6 WHERE `a` = 4 AND `b` = 4 AND `c` = 6; | UPDATE `test`.`shopping` SET `a` = 1, `b` = 1, `c` = 1 WHERE `a` = 1 AND `b` = 10 AND `c` = 1;<br>UPDATE `test`.`shopping` SET `a` = 2, `b` = 2, `c` = 5 WHERE `a` = 2 AND `b` = 10 AND `c` = 5;<br>UPDATE `test`.`shopping` SET `a` = 3, `b` = 3, `c` = 9 WHERE `a` = 3 AND `b` = 10 AND `c` = 9;<br>UPDATE `test`.`shopping` SET `a` = 4, `b` = 4, `c` = 6 WHERE `a` = 4 AND `b` = 10 AND `c` = 6; |

5. Access Development Tool or use other tools for establishing database connections. Execute the rollback statements to roll back the UPDATE operation.

**Figure 8-6** Rollback



The code editor provides the temporary local cache capability. The cached code may be truncated or lost due to the limitation of the browser cache capacity. It is recommended that important code be stored in a local file.

```
UPDATE `test`.`shopping` SET `a` = 1, `b` = 1, `c` = 1 WHERE `a` = 1 AND `b` = 10 AND `c` = 1;
UPDATE `test`.`shopping` SET `a` = 2, `b` = 2, `c` = 5 WHERE `a` = 2 AND `b` = 10 AND `c` = 5;
UPDATE `test`.`shopping` SET `a` = 3, `b` = 3, `c` = 9 WHERE `a` = 3 AND `b` = 10 AND `c` = 9;
UPDATE `test`.`shopping` SET `a` = 4, `b` = 4, `c` = 6 WHERE `a` = 4 AND `b` = 10 AND `c` = 6;

select * FROM `test`.`shopping`
```

The following is the execution result set of select * FROM `test`.`shopping`.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

| | a | b | c |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 5 |
| 3 | 3 | 3 | 9 |
| 4 | 4 | 4 | 6 |

**----End**

# 9 Intelligent Database Diagnosis

DAS provides intelligent diagnosis to monitor database performance metrics and pinpoint SQL queries closely related to metric changes.

## Procedure

**Step 1** Log in to the console.

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

**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.

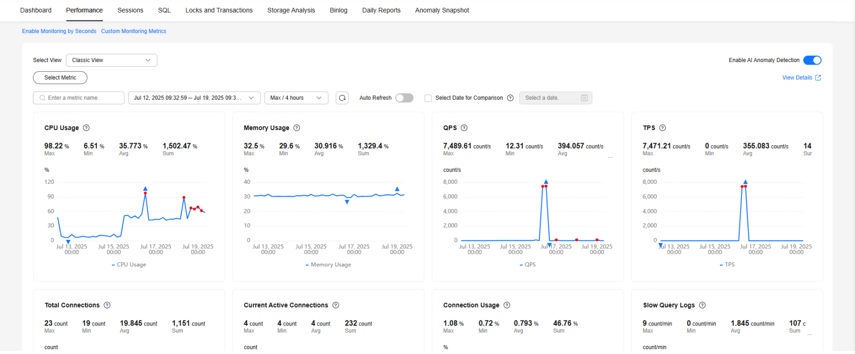
**Step 5** In the upper right corner of the **Instance List** page, search for instances by engine type, instance name, or instance IP address.

**Step 6** Click the **Performance** tab.

- AI anomaly detection

  DAS detects abnormal performance metrics and marks them in red. You can focus on the anomalies during issue analysis. AI anomaly detection is enabled by default. You can toggle on or off **Enable AI Anomaly Detection** in the upper right corner.

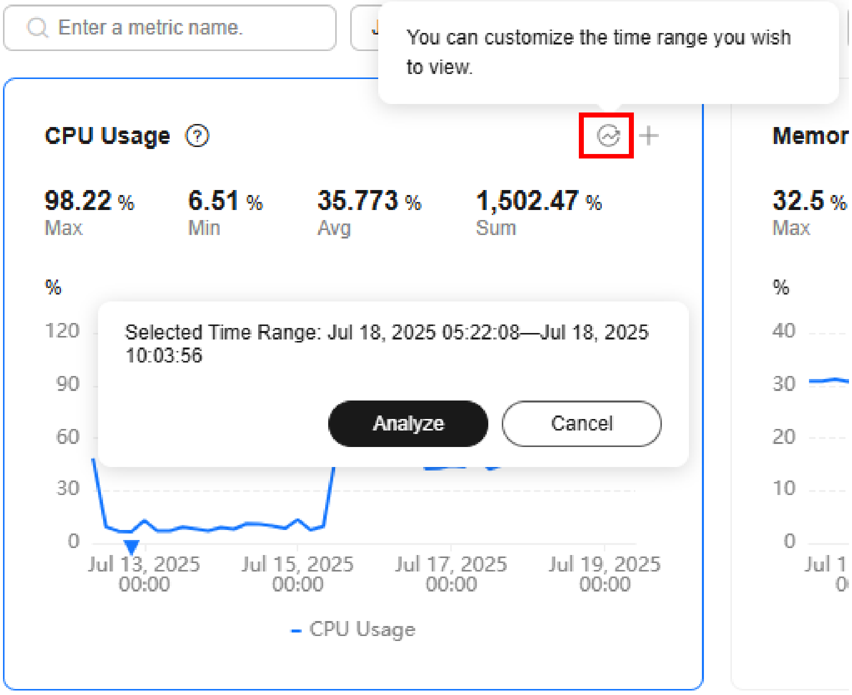
**Figure 9-1** AI anomaly detection



- Select a period close to the time when the anomaly occurred for fault locating and analysis.

  After clicking ⊛, you can drag the cursor in the chart to select a period close to the time when the anomaly occurred. In the displayed dialog box, click **Analyze**. The analysis details page is displayed.

**Figure 9-2** Fault locating and analysis

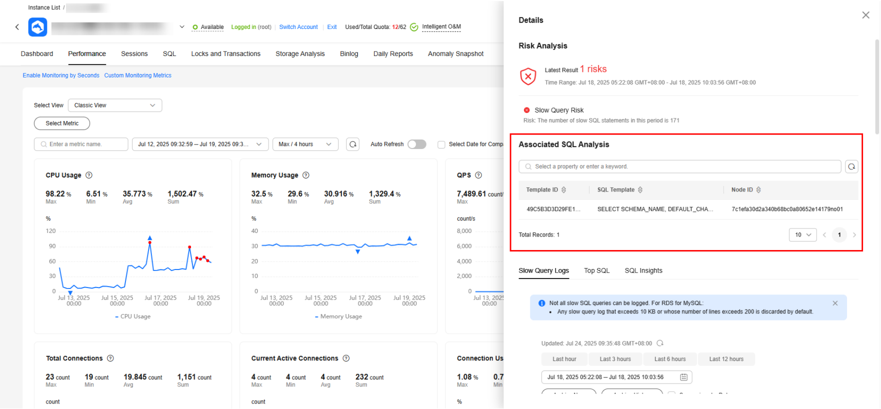


You can customize the time range you wish to view.

Selected Time Range: Jul 18, 2025 05:22:08—Jul 18, 2025 10:03:56

The analysis details include analysis results of risks, associated SQL queries, slow query logs, and top SQL queries. In the **Associated SQL Analysis** area, SQL templates closely related to metric changes are displayed.

**Figure 9-3** Analysis details



Details

Risk Analysis

Latest Result 1 risks

Time Range: Jul 18, 2025 05:22:08 GMT+08:00 - Jul 18, 2025 10:03:56 GMT+08:00

Slow Query Risk

Risk: The number of slow SQL statements in this period is 171

Associated SQL Analysis

| Template ID | SQL Template | Node ID |
|---|---|---|
| 49C5B3D3D29FE1... | SELECT SCHEMA_NAME, DEFAULT_CHA... | 7c1efa30d2a340b68bc0a80652e14179no01 |

Total Records: 1

Not all slow SQL queries can be logged. For RDS for MySQL:

- Any slow query log that exceeds 10 KB or whose number of lines exceeds 200 is discarded by default.

Updated: Jul 24, 2025 09:35:48 GMT+08:00

Jul 18, 2025 05:22:08 — Jul 18, 2025 10:03:56

**----End**

# 10 SQL Diagnosis

DAS can diagnose SQL statements executed on databases.

## Prerequisites

You have created an RDS for MySQL instance.

## Procedure

**Step 1** Log in to the console.

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

**Step 3** Click ☰ in the upper left corner. Choose **Databases** > **Data Admin Service**.

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

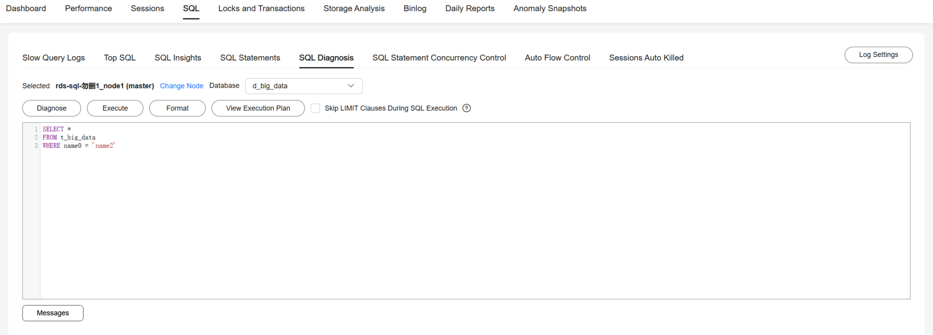
Alternatively, on the **Overview** page, click **Go to Intelligent O&M**.

**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** Click the **SQL** tab and then **SQL Diagnosis**.

**Step 7** Select a node and database. Enter the SQL statement to be diagnosed.

**Figure 10-1** SQL diagnosis



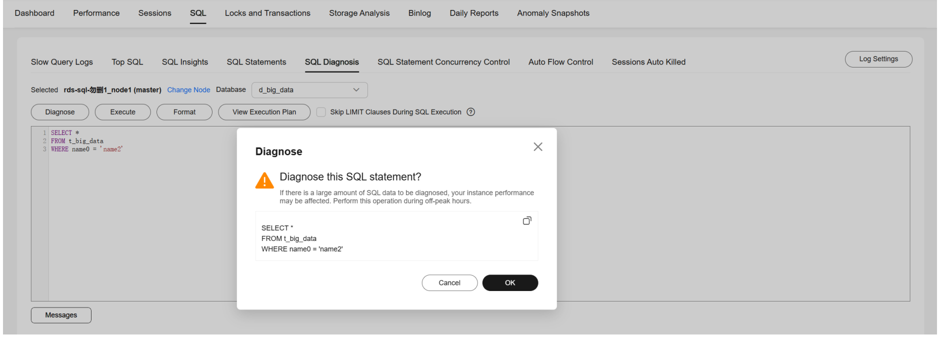
Selected **rds-sql-勿删1_node1 (master)** Database d_big_data

□ Skip LIMIT Clauses During SQL Execution

```
SELECT *
FROM t_big_data
WHERE name0 = 'name2'
```

**Step 8** Click **Diagnose**. In the displayed dialog box, click **OK**.

**Figure 10-2** Confirming SQL diagnosis



Selected **rds-sql-勿删1_node1 (master)** Change Node Database d_big_data

☐ Skip LIMIT Clauses During SQL Execution

```
SELECT *
FROM t_big_data
WHERE name0 = 'name2'
```

**Diagnose**

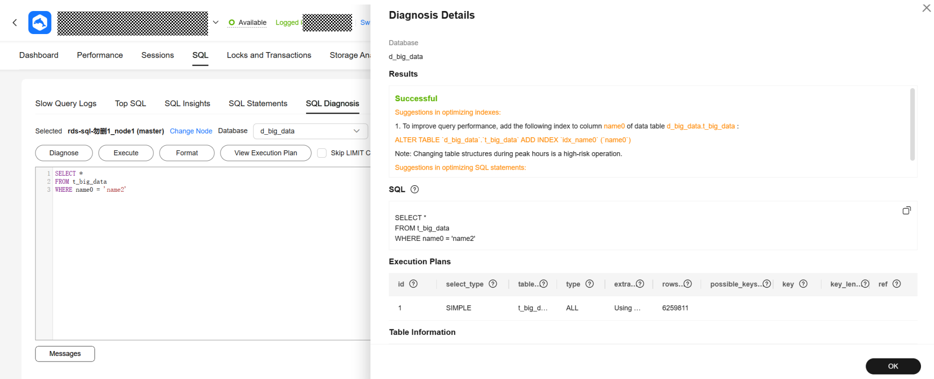
Diagnose this SQL statement?

If there is a large amount of SQL data to be diagnosed, your instance performance may be affected. Perform this operation during off-peak hours.

```
SELECT *
FROM t_big_data
WHERE name0 = 'name2'
```

**Step 9** On the **Diagnosis Details** page, check the database, diagnosis results, diagnosed SQL statement, execution plan, and table.

**Figure 10-3** SQL diagnosis details



Selected **rds-sql-勿删1_node1 (master)** Change Node Database d_big_data

```
SELECT *
FROM t_big_data
WHERE name0 = 'name2'
```

**Diagnosis Details**

Database
d_big_data

**Results**

Successful

Suggestions in optimizing indexes:

1. To improve query performance, add the following index to column name0 of data table d_big_data.t_big_data :

ALTER TABLE `d_big_data`.`t_big_data` ADD INDEX `idx_name0` (`name0`)

Note: Changing table structures during peak hours is a high-risk operation.

Suggestions in optimizing SQL statements:

**SQL**

```
SELECT *
FROM t_big_data
WHERE name0 = 'name2'
```

**Execution Plans**

| id | select_type | table.. | type | extra.. | rows.. | possible_keys.. | key | key_len.. | ref |
|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | t_big_d... | ALL | Using ... | 6259811 | | | | |

**Table Information**

**----End**