

Data Admin Service

Best Practices

Issue 01
Date 2023-12-30



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

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?.....	1
2 How Do I Check and Optimize Tables by Checking Top SQL?.....	7
3 Fixing Slow SQL Queries.....	10
4 Lock Analysis.....	14
5 Fixing High CPU Usage on DAS.....	22
6 Fixing Insufficient Storage on DAS.....	28

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

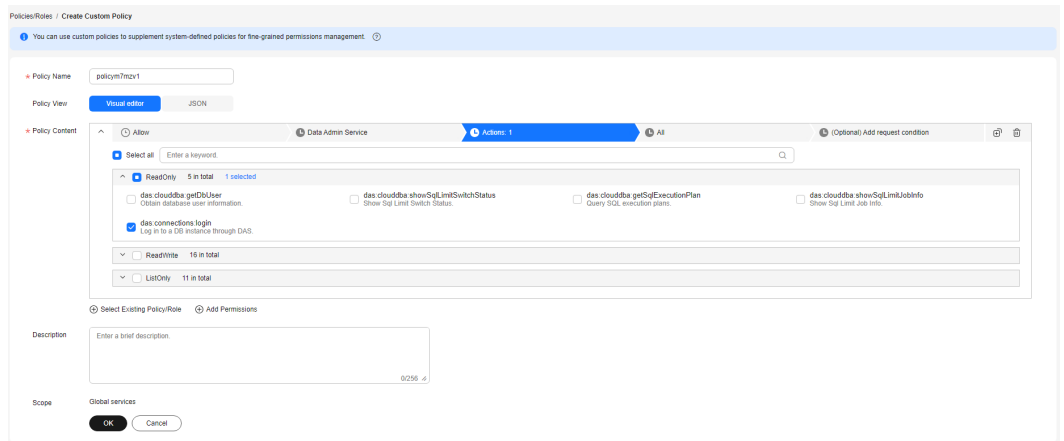


Table 1-1 Parameter description

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. 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

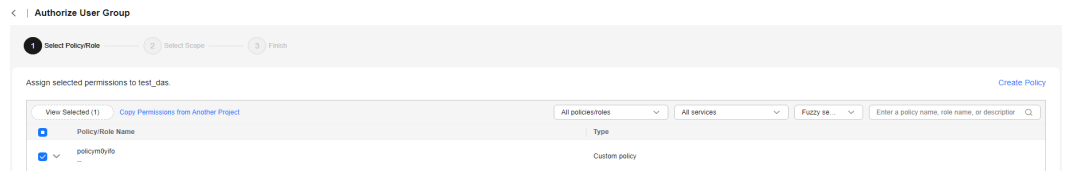


Step 3 Create a user group and assign the custom permission policy created in **Step 2** to the user group.

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

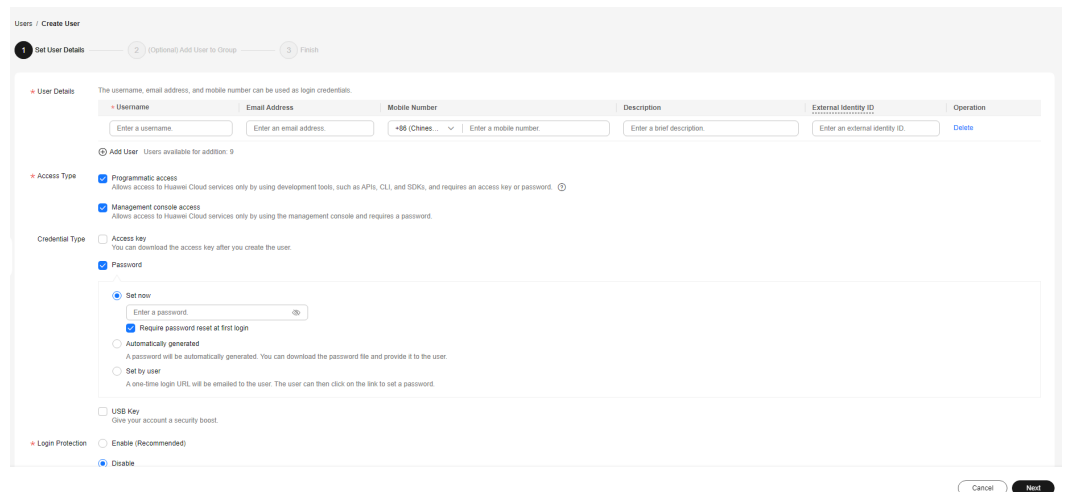


3. Click **Next**, select **All resources**, click **OK**, and complete subsequent operations.

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

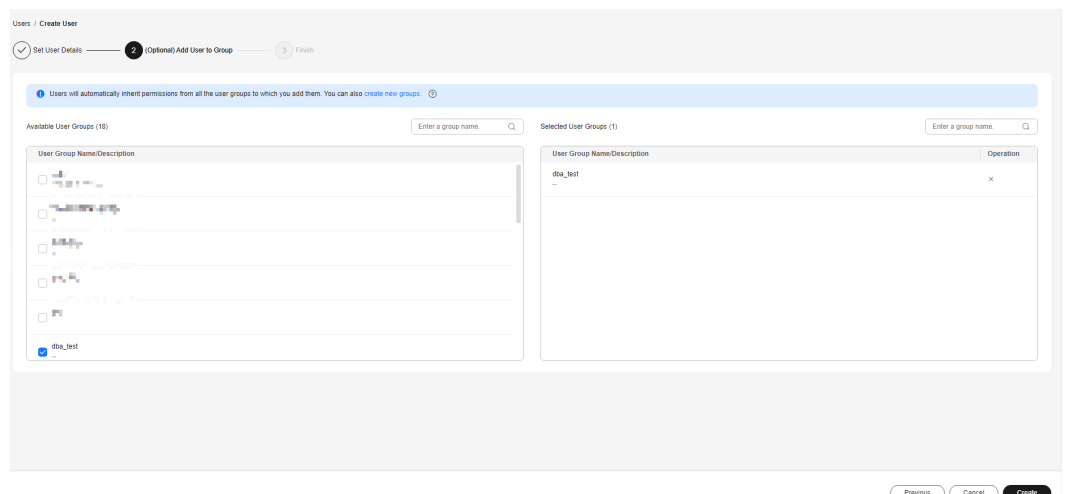
1. In the navigation pane, choose **Users** and click **Create**. On the displayed page, set user basic information.

Figure 1-4 Setting user basic information



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



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

The screenshot shows the 'Create Account' dialog box. The 'Username' field contains 'rds_test'. The 'Host IP Address' field is empty. The 'Database' section has two panels: 'Database Not Authorized' and 'Database Authorized'. Both panels show a search bar 'Enter a keyword.' and a table with columns 'Name' and 'Permission'. The 'Database Not Authorized' panel shows 'No data available'. The 'Database Authorized' panel also shows 'No data available'. Below the database panels are 'Password' and 'Confirm Password' fields. At the bottom right are 'OK' and 'Cancel' buttons.

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

DB Instance	DB Engine Version	DB Instance Type	Status
<input checked="" type="radio"/> [Instance ID]	MySQL 5.6	Single	Available
<input type="radio"/> [Instance ID]	MySQL 5.6	Single	Abnormal
<input type="radio"/> [Instance ID]	MySQL 5.7	Single	Abnormal
<input type="radio"/> [Instance ID]	MySQL 8.0	Ha	Available
<input type="radio"/> [Instance ID]	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](#).

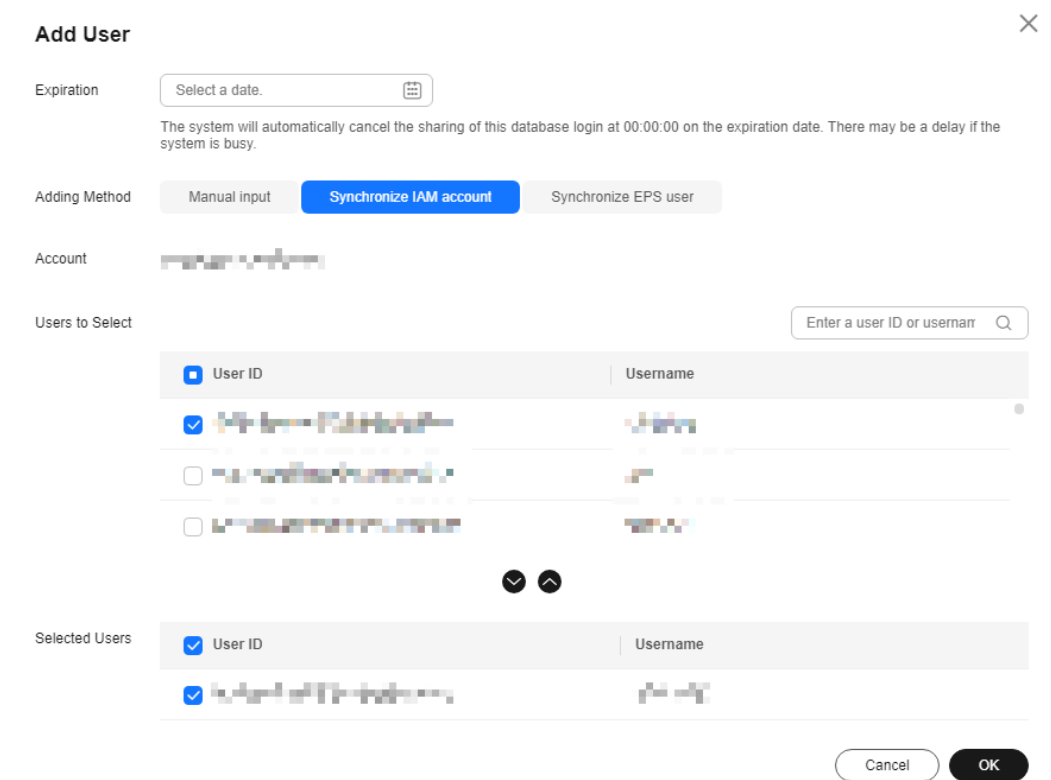
- Share the login information of the read-only account with the IAM user. Locate the target instance and click the number in the **Additional Users** column.

Figure 1-8 Sharing a login with an IAM user

DB Instance	DB Engine Version	Source Database	Login Username	Remember...	Description	Created	Additional Us...	Operation
[Instance ID]	MySQL 5.7.43	RDS	lbr	Yes	-	Aug 19, 2024 15:13:00 GMT+...	View (0)	Log In Modify Delete Intelligent CSM

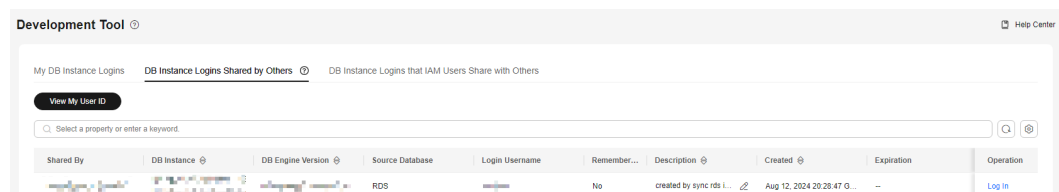
Click **Add User**. On the displayed page, specify the expiration time, select **Synchronize IAM account** for **Adding Method**, then select the IAM account created in [Step 4](#) for **Users to Select**, and click **OK**.

Figure 1-9 Adding a user



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



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 DAS console.](#)

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

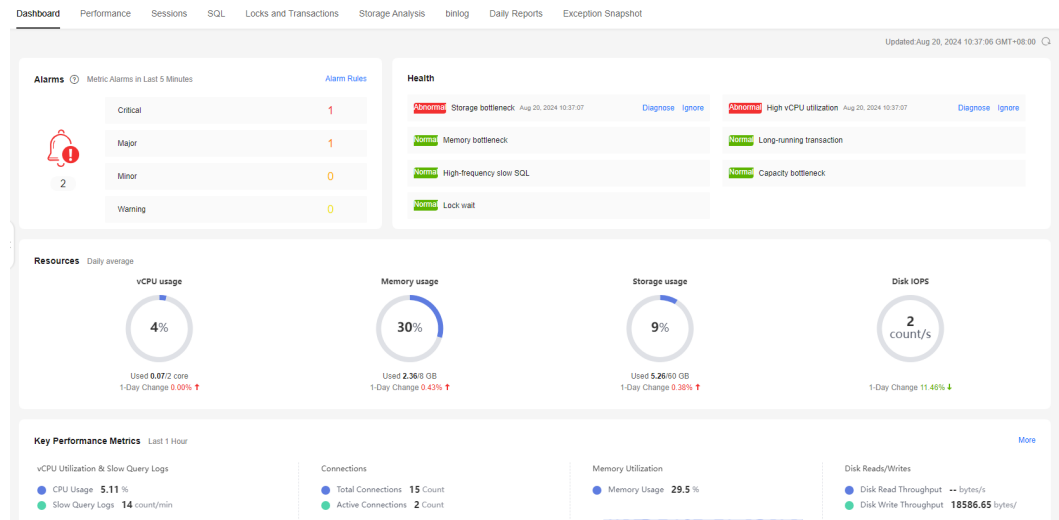
Step 3 Click  in the upper left corner, and under **Databases**, click **Data Admin Service**.

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

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

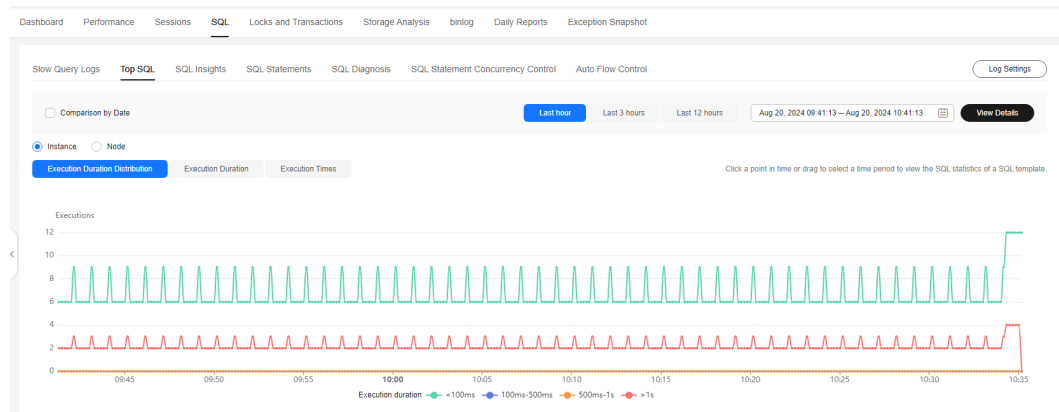
Step 5 In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details**.

Figure 2-1 Intelligent O&M instance overview page



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

Figure 2-2 TOP SQL



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.

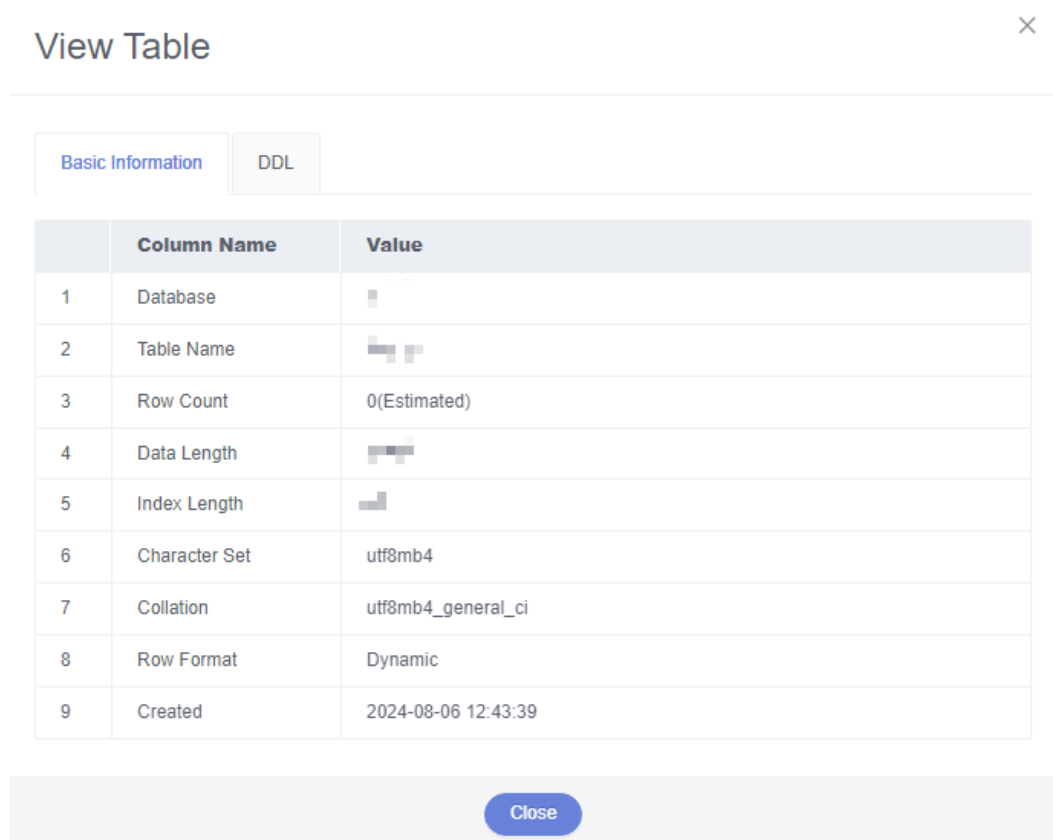
Figure 2-3 SQL statements

SQL Template (Top50)
Time Range: Aug 20, 2024 09:43:25 - Aug 20, 2024 10:43:25

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



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 DAS console](#).

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

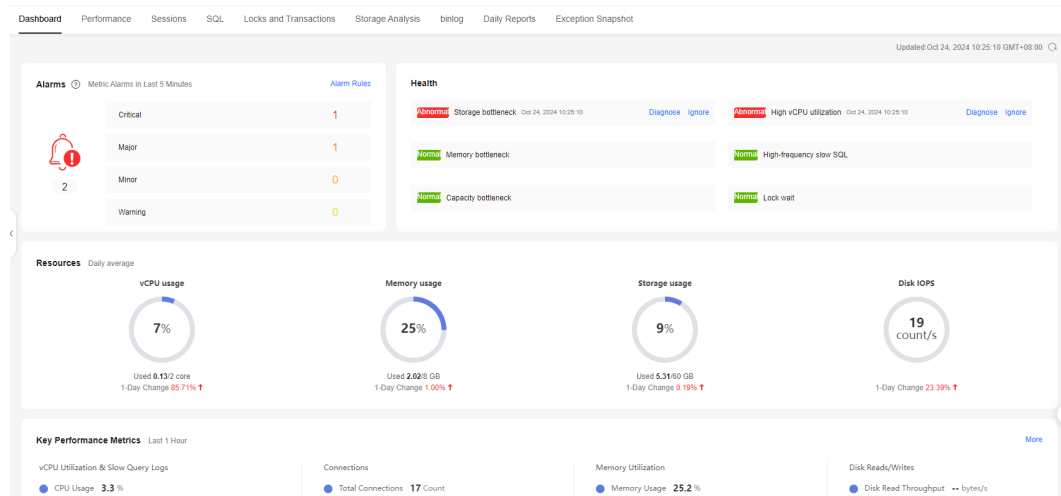
Step 3 Click  in the upper left corner, and under **Databases**, click **Data Admin Service**.

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

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.

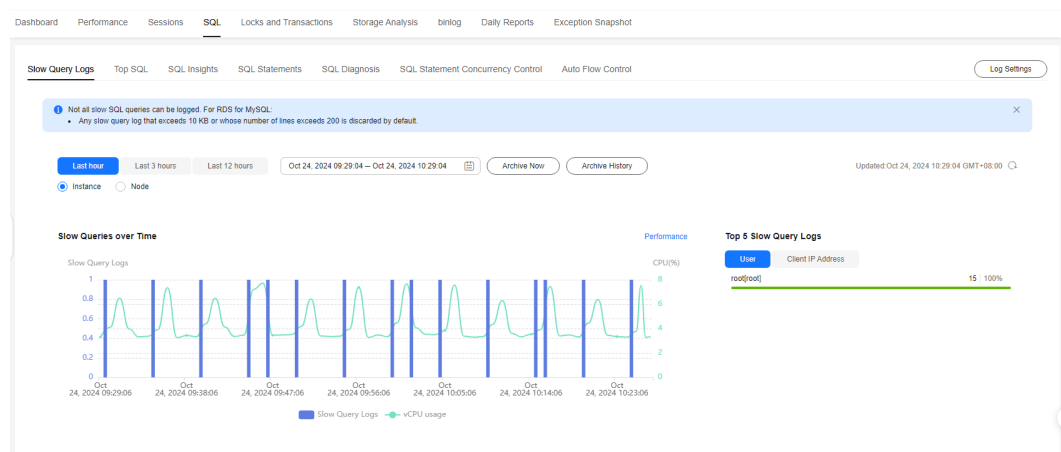
Figure 3-1 Intelligent O&M



Step 6 Choose SQL > Slow Query Logs.

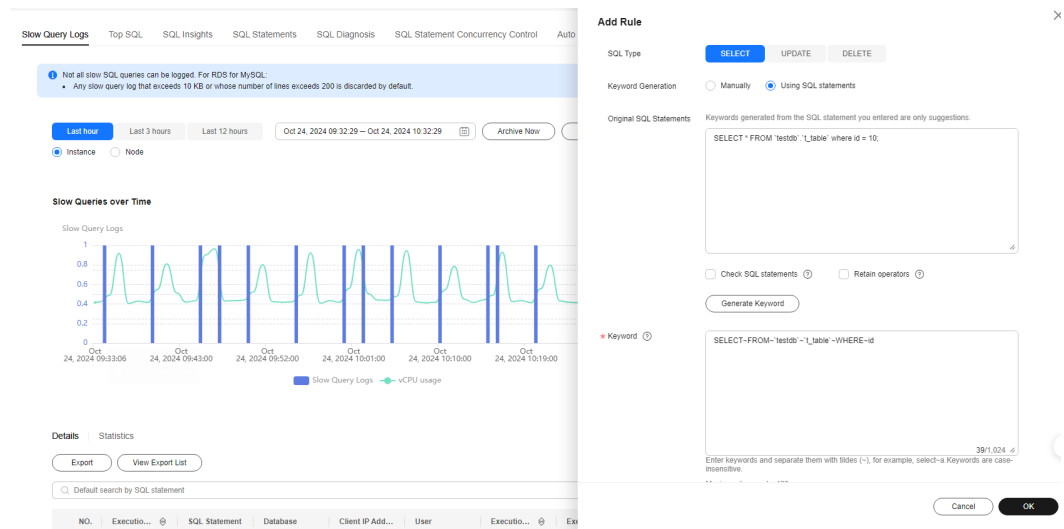
Select a time range and view trends, details, and statistics of the slow query logs. You can click **Export** to export a slow query log to your local PC.

Figure 3-2 Slow Query Logs



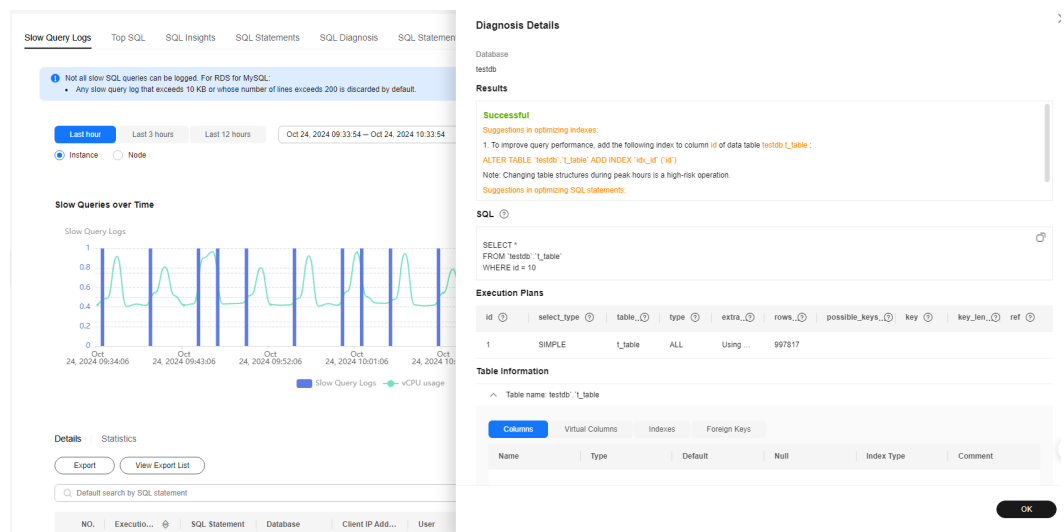
Step 7 In the **Details** area, click **Concurrency Control** in the **Operation** column. In the displayed dialog box, create a concurrency control rule for the current SQL statement by configuring parameters, including **Max. Concurrency** to ensure stability of core services. For details, see [SQL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

Figure 3-3 Concurrency control



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 Diagnosis




----End

Viewing Slow SQL Queries on All Instances

Step 1 [Log in to the DAS console](#).

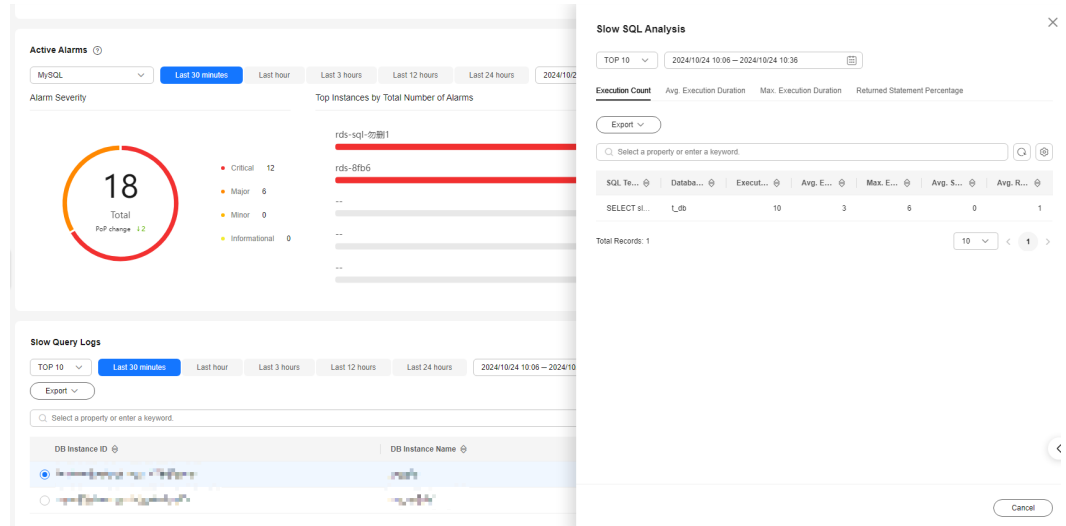
Step 2 Click  in the upper left corner and select a region and project.

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 > 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 Overview of slow SQL queries



----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);

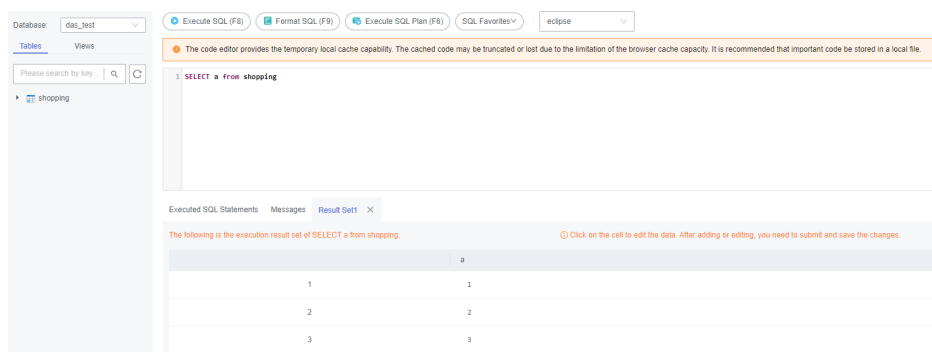
NOTE

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.

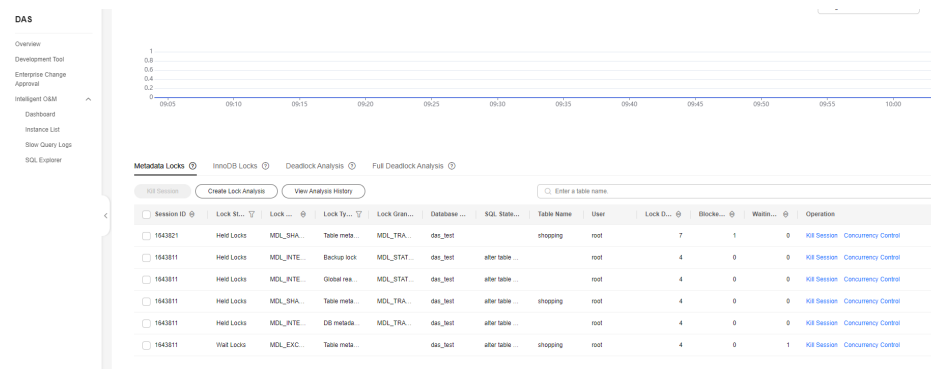
Step 2 Analyze and process metadata locks.**NOTE**

- 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

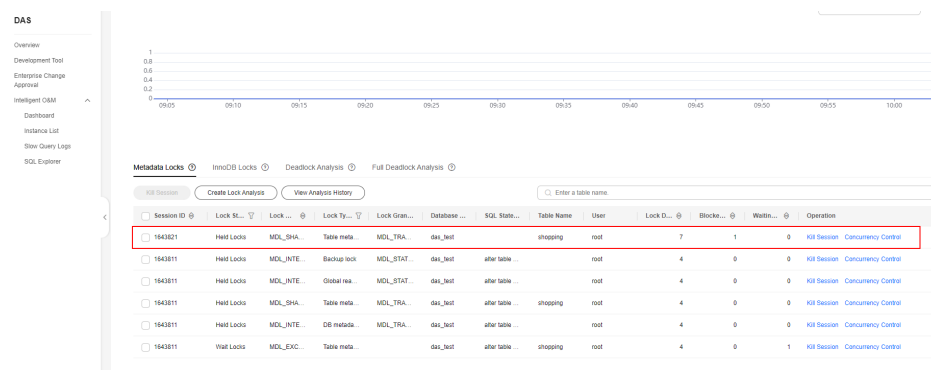
- b. Start a transaction and update data in the **shopping** table. Do not submit the transaction.
BEGIN;
UPDATE shopping SET b = 3 WHERE a = 1;
2. Create session 2 and execute the following statement to add an index to the **shopping** table:
ALTER TABLE shopping ADD INDEX idx_name(b);
 3. Go to the 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



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

Figure 4-3 Selecting the target metadata lock

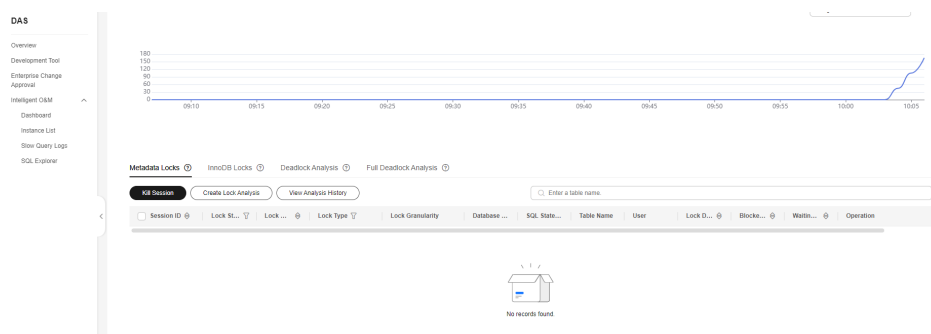


6. Refresh the metadata lock list. The query statement has been executed, and the DDL statement is being executed.

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



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.

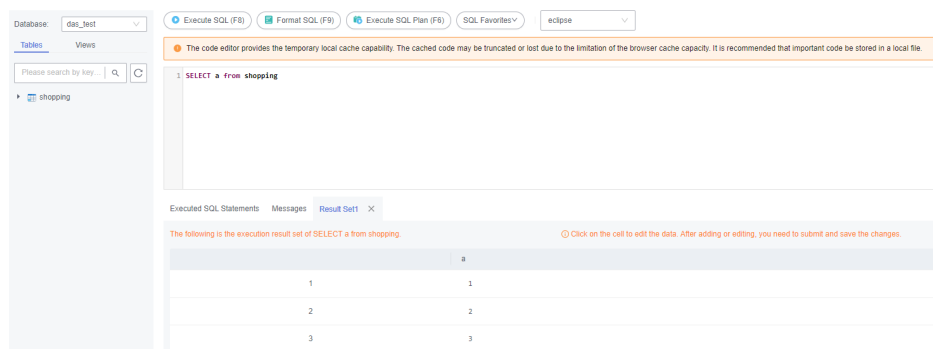
1. **Log in to the Huawei Cloud DB instance** and execute a SQL statement to create session 1.

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



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



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.

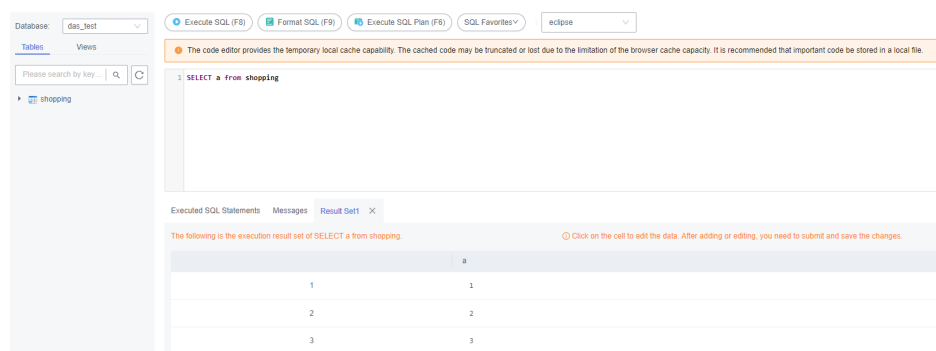
NOTE

- 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



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

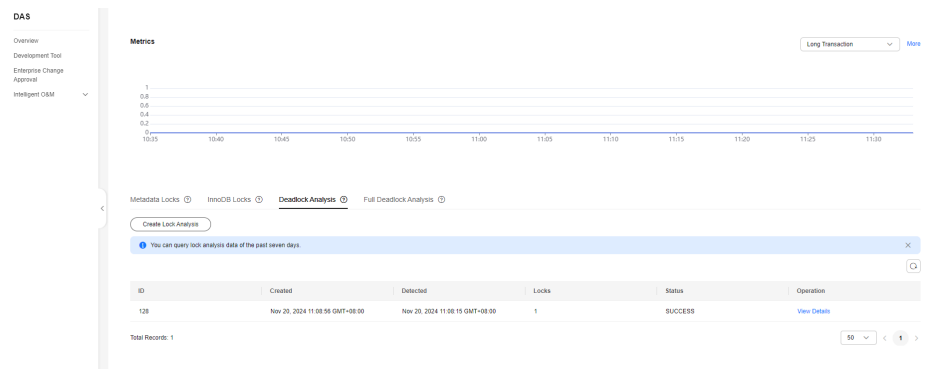
Table 4-1 Simulating a deadlock

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.

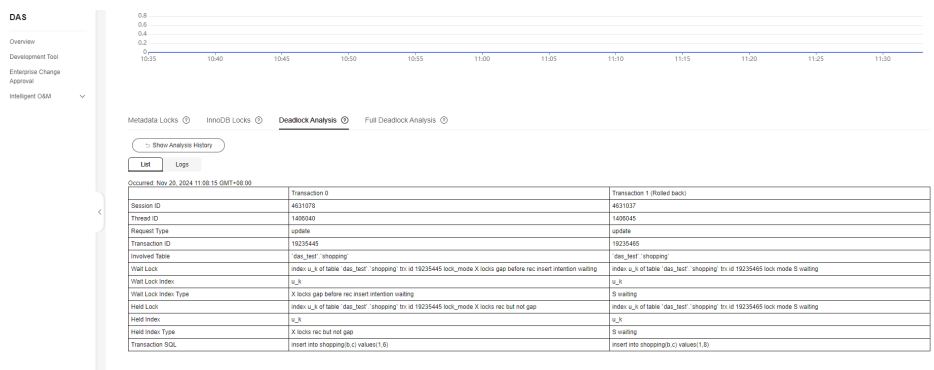
- Go to the DAS homepage. In the navigation pane, choose **Intelligent O&M > Instance List**.
- Locate the target instance, click **Details**. Click the **Locks and Transactions**, **Locks**, and **Deadlock Analysis** tabs in sequence. Click **Create Lock Analysis**, refresh the page, and check the list.

Figure 4-8 Recent deadlock



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



Figure 4-9 Details of a recent deadlock



Step 5 Analyze full deadlocks.

NOTE

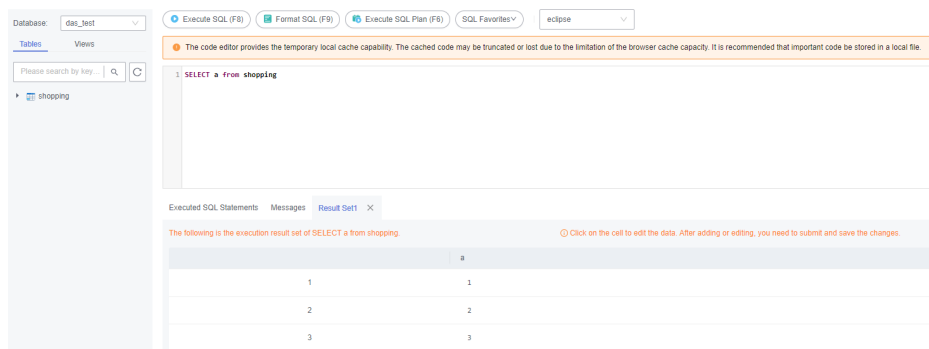
- DAS analyzes error logs at regular intervals, parses deadlock information, and performs comprehensive deadlock analysis.
- Dependency parameters:
 - Enable **innodb_deadlock_detect** (only for RDS for MySQL 5.7).
 - Enable **innodb_print_all_deadlocks** and set **log_error_verbosity** (only for RDS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

1. **Log in to the DAS 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



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

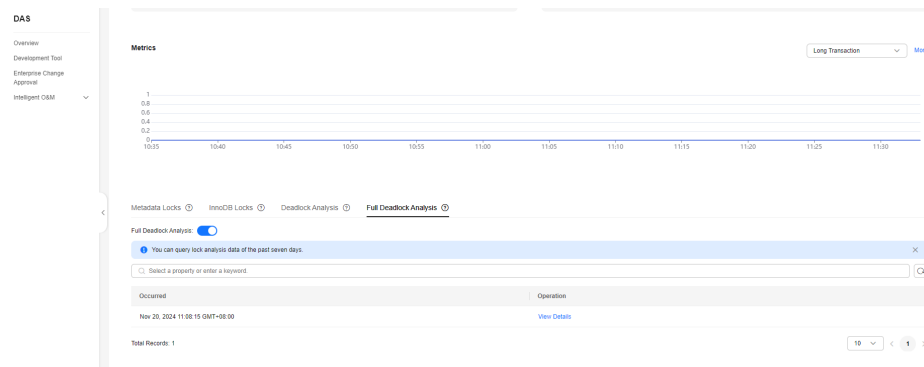
Table 4-2 Simulating a deadlock

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.

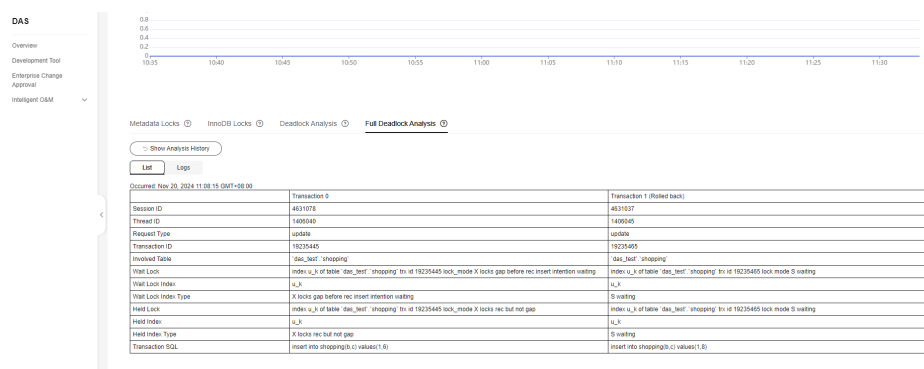
9. View the full deadlock analysis list.

Figure 4-11 Full deadlock



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

Figure 4-12 Full deadlock details



----End

5 Fixing High CPU Usage on DAS

High CPU usage is caused by the following reasons:


- **Slow SQL queries**
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 connections**
When QPS of a DB instance increases, it has to process a large number of concurrent operations at the same time, and the CPU usage increases accordingly. 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)

If a Huawei Cloud database pushes an alarm indicating that CPU usage of the DB instance increases sharply, the SRE will:

Step 1 [Log in to the DAS console.](#)

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

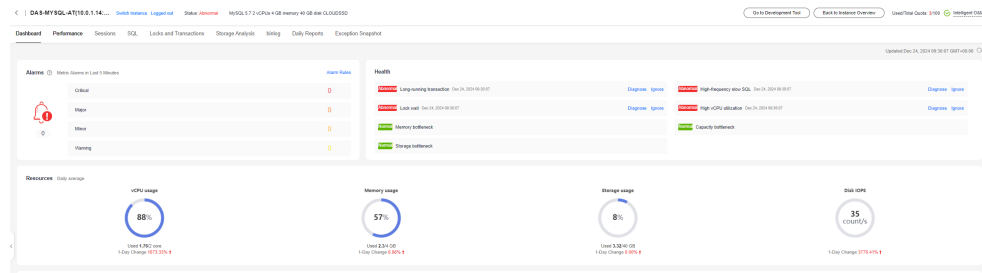
Step 3 Click  in the upper left corner. Choose **Databases > Data Admin Service**.

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

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

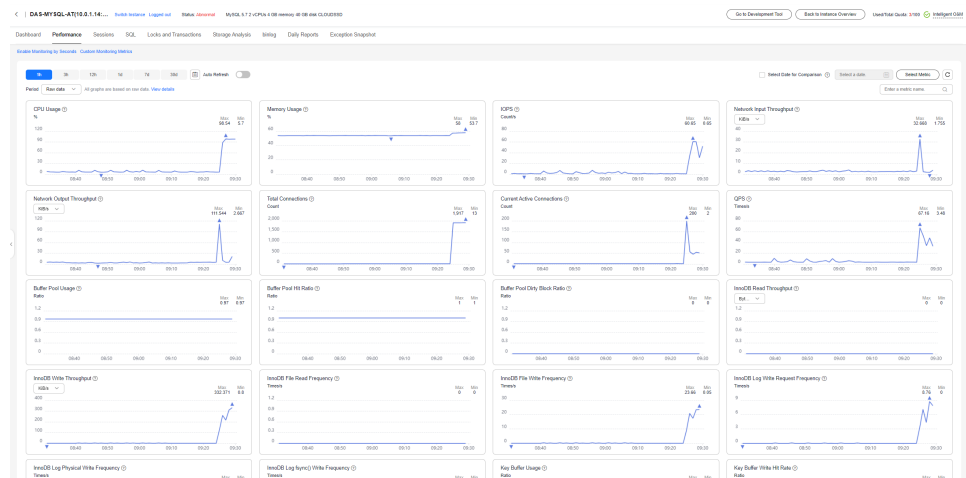
Step 5 In the upper right corner of the instance list, filter instances by engine, name, or IP address. Click **Details** to go to the **Dashboard** tab page.

Figure 5-1 Intelligent O&M



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

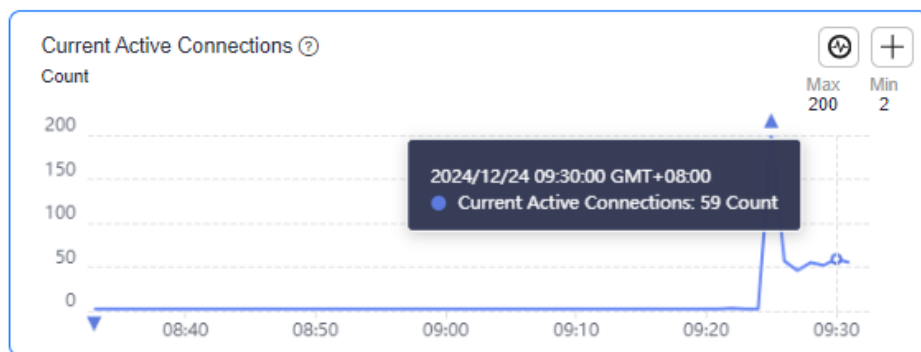


Figure 5-4 Total Connections



Figure 5-5 QPS

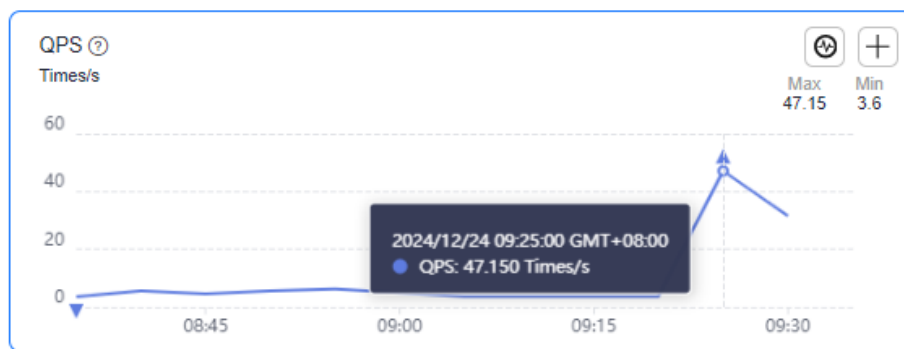


Figure 5-6 Slow Query Logs

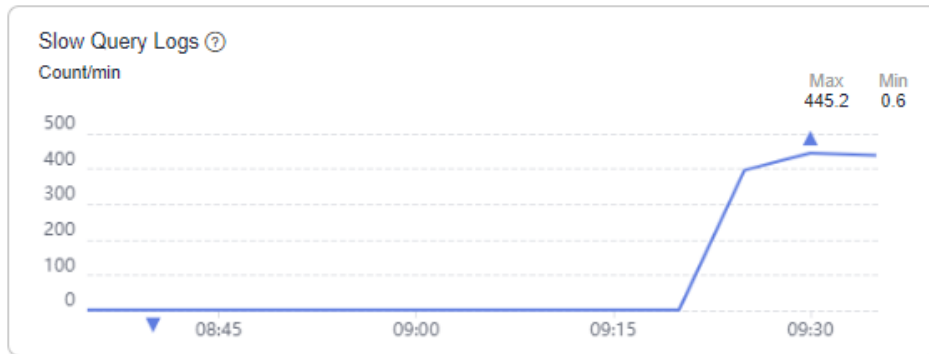
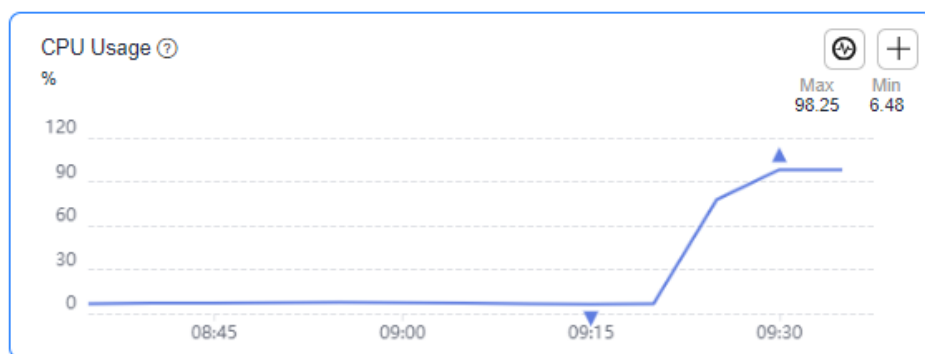
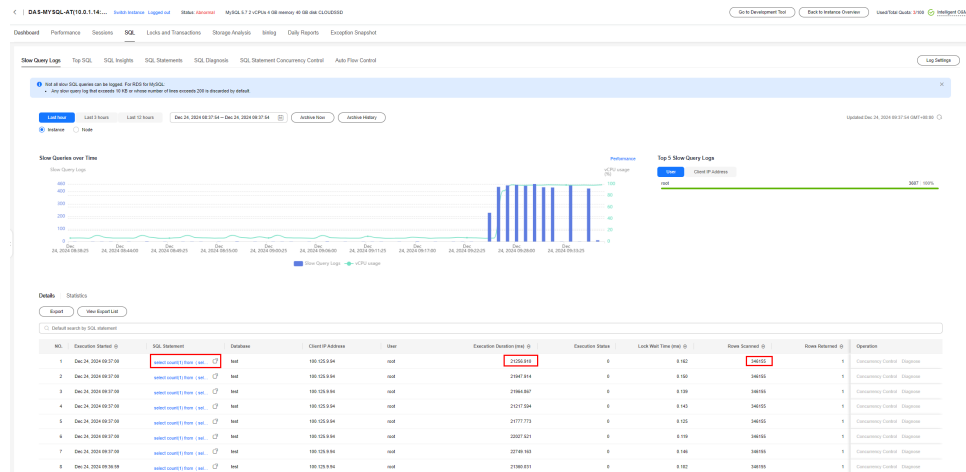


Figure 5-7 CPU Usage



- 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



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 an 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

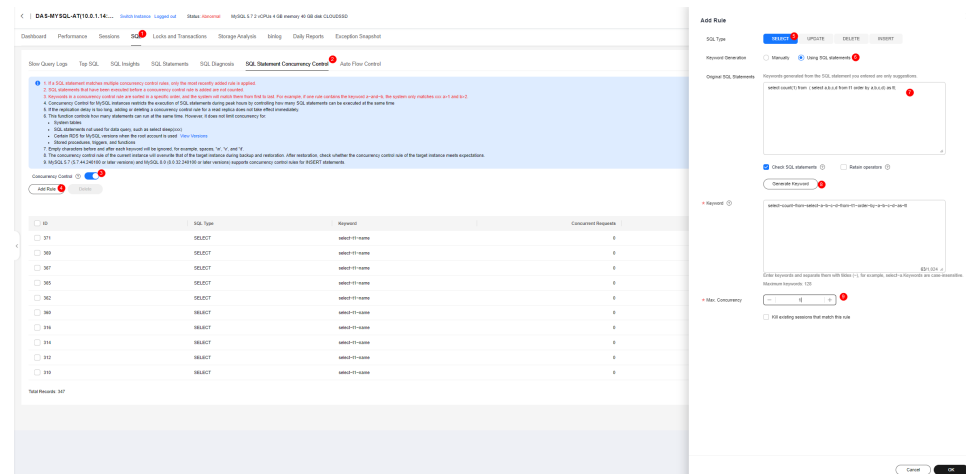
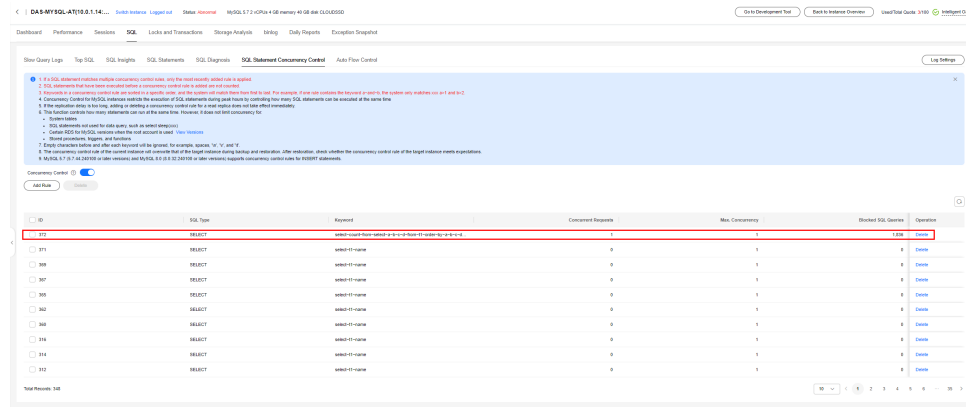


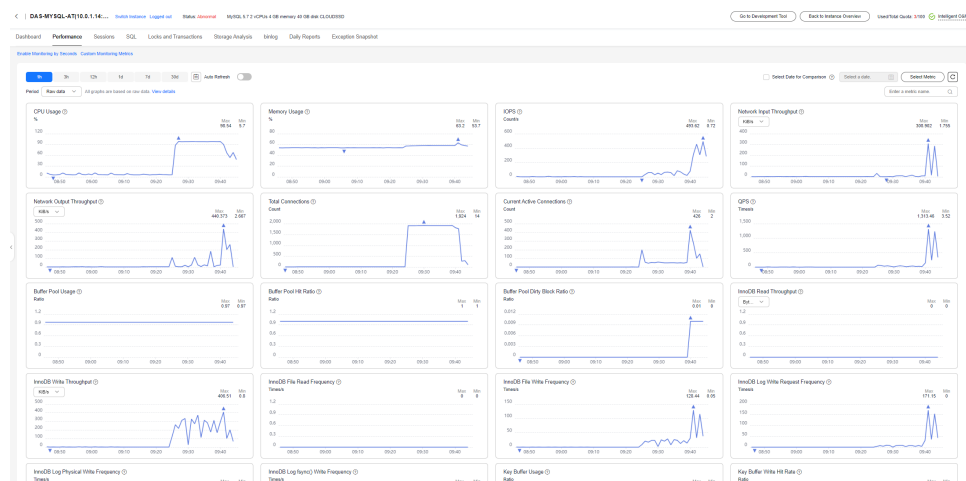
Figure 5-10 Selecting a SQL statement concurrency control rule



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



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

Figure 5-12 Slow queries over time

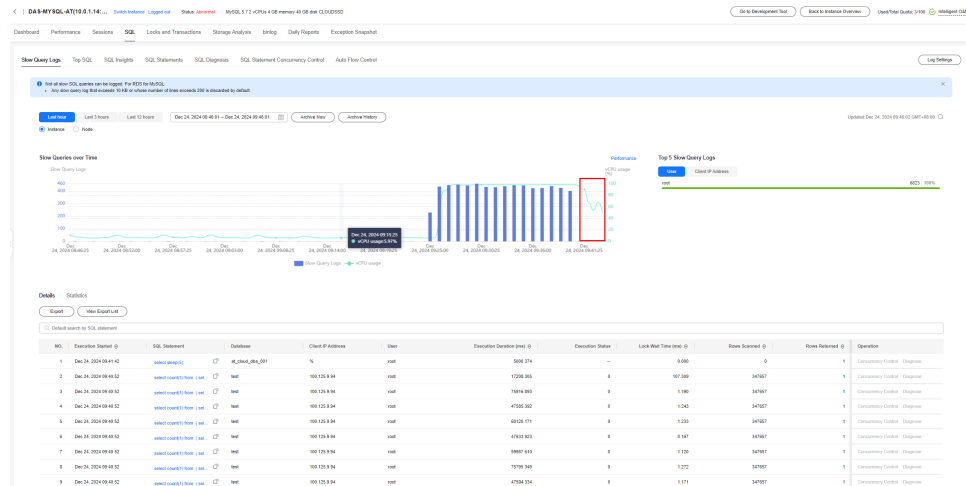
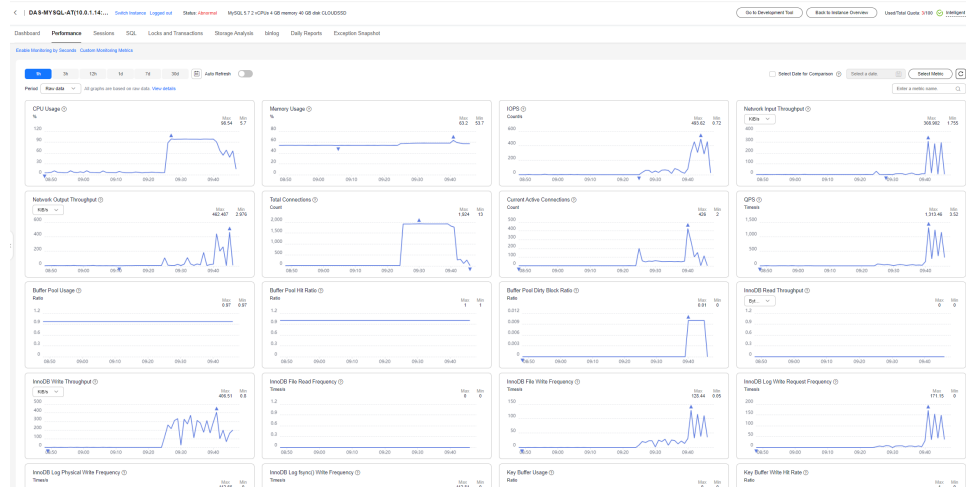


Figure 5-13 Performance metrics



----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.



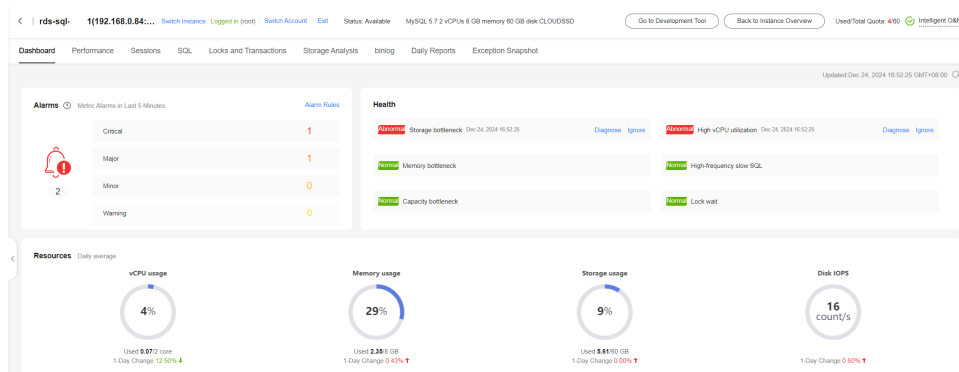
1. [Log in to the DAS console](#).
2. Click  in the upper left corner and select a region and project.
3. Click  in the upper left corner. Choose **Databases > Data Admin Service**.
4. In the navigation pane, choose **Intelligent O&M > Instance List**. Alternatively, on the **Overview** page, click **Go 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 Intelligent O&M

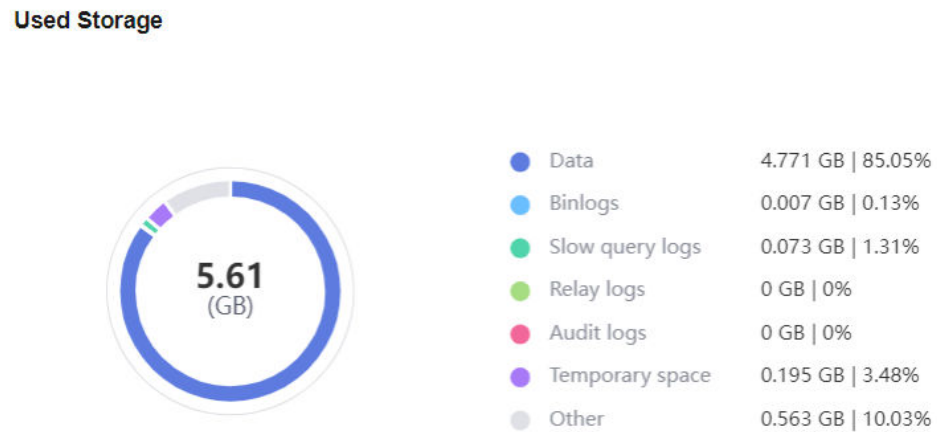


6. Click the **Storage Analysis** tab.

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

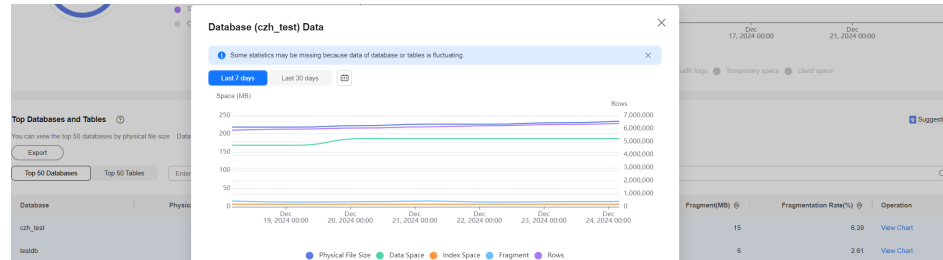
You can view the top 50 databases by physical file size. Data is automatically collected at about 04:00 every day. Last collected Dec 24, 2024 04:01:59 GMT+08:00.

Database	Physical File Size(MB)	Rows	Data Space(MB)	Index Space(MB)	Fragment(MB)	Fragmentation Rate(%)	Operation
testb	229.5316	2097504	93.2497	14.8751	6	2.61	View Chart
testa	234.6884	9403460	157.1089	7.0654	15	6.39	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

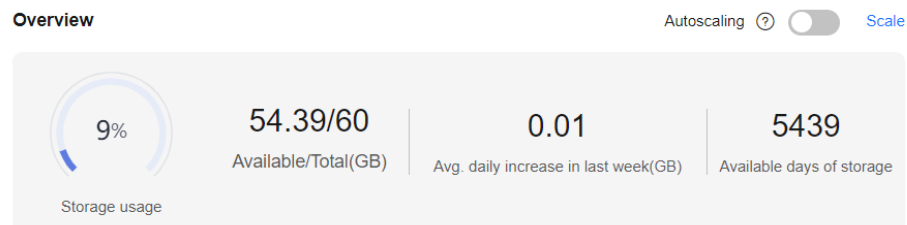


- c. Obtain audit logs and SQL Explorer information of the instance at the time when the exception occurred. Delete the data which grew sharply due to abnormal changes.
4. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

Step 4 Scale up the storage.

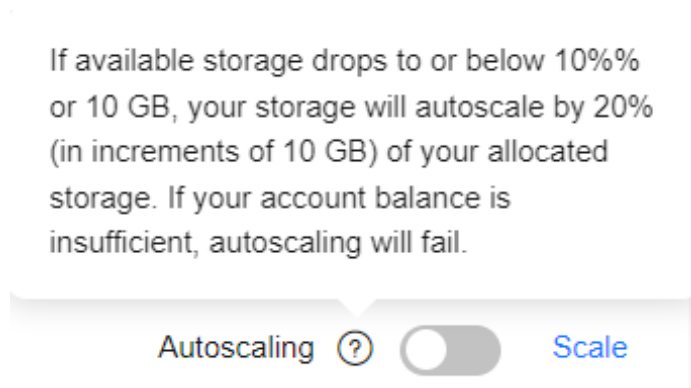
1. Check the **Overview** area. If the remaining usage is less than 10% or the available space is less than 10 GB, you are advised to click **Scale**.

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.

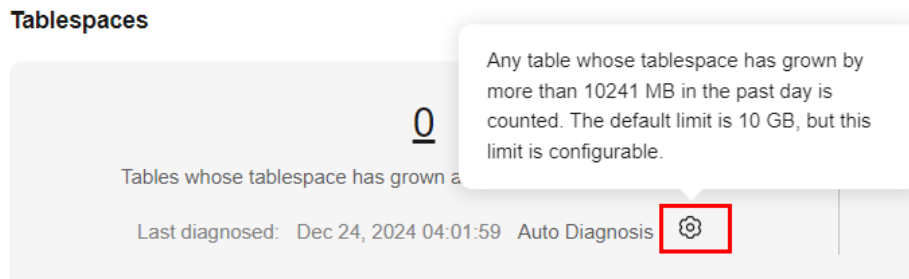
Figure 6-7 Autoscaling



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



----End