

Distributed Database Middleware

FAQs

Issue 01
Date 2021-01-15



Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Contents

1 General Questions.....	1
1.1 What High-Reliability Mechanisms Does DDM Provide?.....	1
1.2 Does DDM Store Service-Related Data?.....	1
1.3 How Do I Select and Configure a Security Group?.....	1
2 DDM Usage.....	4
2.1 How Does DDM Perform Sharding?.....	4
2.2 How Can I Rectify a Fault in Accessing DDM by Using the JDBC Driver?.....	5
2.3 What Version and Parameters Should I Select?.....	6
2.4 Why It Takes So Long Time to Export Data from MySQL Using mysqldump?.....	8
2.5 How Should I Handle the Duplicate Primary Key Error Occurring After Data Is Imported into DDM?	8
2.6 What Should I Do If an Error Message Is Returned After I Specify an Auto-Increment Primary Key?	8
2.7 How Do I Handle the Error Reported When Parameter Configuration Does Not Time Out?.....	8
2.8 Which Should I Delete First, Schema or Associated RDS DB Instances?.....	8
2.9 Should I Manually Delete Databases and Accounts Remained in the Associated RDS DB Instances After a Schema Is Deleted?.....	9
3 SQL Syntax.....	10
3.1 Does DDM Support Cross-Database Access of SQL?.....	10
3.2 Does DDM Support Distributed JOIN?.....	10
3.3 How Do I Optimize SQL Statements?.....	10
3.4 Does DDM Support Forced Data Type Conversion?.....	11
3.5 What Should I Do If an Error Is Reported When Multiple Data Records Are Inserted into Batches Using the INSERT Statement?.....	11
4 RDS-related Questions.....	12
4.1 Is the Name of a Database Table Case-Sensitive?.....	12
4.2 What Risky Operations on RDS for MySQL Will Affect DDM?.....	12
4.3 How Do I Handle Data with Duplicate Primary Keys in a Table?.....	14
4.4 How Can I Query RDS Information by Running Command show full innodb status ?.....	16
5 Connection Management.....	17
5.1 How Do I Connect to a DDM Instance?.....	17
5.2 Can I Connect to DDM Instances from My Local Environment?.....	17
5.3 What Should I Do If Garbled Characters Are Displayed When I Connect to DDM Using MySQL?.....	18

A Change History..... 19

1 General Questions

1.1 What High-Reliability Mechanisms Does DDM Provide?

Protection of Data Integrity

DDM instance faults do not affect data integrity.

- Service data is stored in shards of RDS MySQL DB instances, but not on DDM.
- Configuration information of schemas and logical tables is stored in DDM databases. Active and standby DDM databases are highly available.

Fault Tolerance

DDM uses the RDS MySQL fault tolerance mechanism. RDS for MySQL ensures that data is written to a database. DDM considers that data is written to the database if the RDS MySQL DB instance returns a message indicating that the SQL statements is executed successfully.

1.2 Does DDM Store Service-Related Data?

DDM instances do not store service-related data, which is stored in shards of the RDS MySQL DB instances.

The hard disks on DDM nodes are used to store log files and temporary files, which are periodically cleared to ensure sufficient space.

1.3 How Do I Select and Configure a Security Group?

DDM uses VPCs and security groups to ensure security of your instances. The following provides guidance for you on how to correctly configure a security group.

Intra-VPC Access to DDM Instances

Access to a DDM instance includes access to the DDM instance from the ECS where a client is located and access to its associated RDS MySQL DB instance.

The ECS, DDM instance, and RDS MySQL DB instance must be in the same VPC. In addition, correct rules should be configured for their security groups to allow network access.

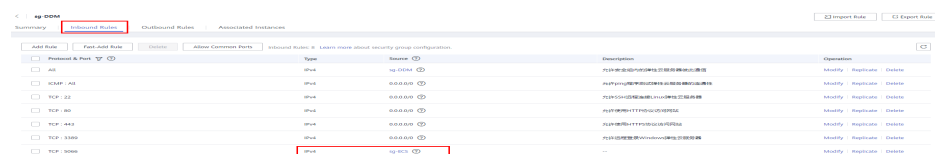
1. You are advised to configure the same security group for the ECS, DDM instance, and RDS MySQL DB instance. After a security group is created, network access in the group is not restricted by default.
2. If different security groups are configured, you may need to refer to the following configurations:

NOTE

- Assume that the ECS, DDM instance, and RDS MySQL DB instance are configured with security groups **sg-ECS**, **sg-DDM**, and **sg-RDS**, respectively.
- Assume that the service port of the DDM instance is **5066** and that of the RDS MySQL DB instance is **3306**.
- The remote end should be a security group or an IP address.

Add the rules in [Figure 1-1](#) to the security group of the ECS to ensure that your client can access the DDM instance.

Figure 1-1 ECS security group rules



Add the rules in [Figure 1-2](#) and [Figure 1-3](#) to the security group of the ECS where your DDM instance is located so that your DDM instance can access the associated RDS MySQL DB instance and can be accessed by your client.

Figure 1-2 Configuring security group inbound rules for your DDM instance

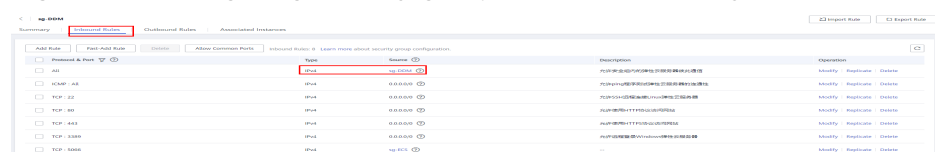
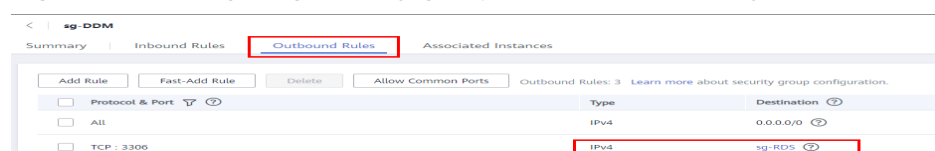
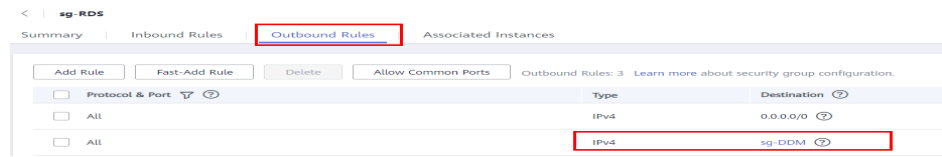


Figure 1-3 Configuring security group outbound rules for your DDM instance



Add the rules in [Figure 1-4](#) to the security group of the ECS where the RDS MySQL DB instance is located so that your DDM instance can access the RDS DB instance.

Figure 1-4 Configuring security group rules of the RDS MySQL DB instance



2 DDM Usage

2.1 How Does DDM Perform Sharding?

Distributed databases use shard-based storage, which removes capacity bottlenecks of single-node databases caused by a large amount of data in one unsharded table. Therefore, when creating a schema and logical tables, you need to consider your actual conditions and determine whether to create sharded tables and which sharding rule should be used.

NOTE

Avoid cross-shard JOIN operations on the data that is stored in different shards, to ensure optimal performance and resource availability.

- Whether logical tables are sharded

DDM supports three types of tables: broadcast tables, sharded tables, and unsharded tables. You can select the most appropriate type based on your specific needs.

- One unsharded table is created and stores data only in the first shard.
- One broadcast table is created in each shard and stores the same full data.
- Sharded tables are created in each shard, and data is distributed to these sharded tables based on a certain sharding rule.

- Which sharding rules are used

The selection of a sharding key is important for each logical table. Selecting the sharding key based on your needs is recommended. If an entity relationship exists between different logical tables, select the same field as the sharding key to avoid cross-shard JOIN.

Pay attention to the following suggestions before determining whether to use sharding:

- Do not shard tables that each have less than 10 million data records.
- Shard the tables that each have more than 10 million data records. Storing data in different sharded tables removes performance bottlenecks caused by a large amount of data in one table, while also improving concurrency capability. Select an appropriate sharding key in advance.

- Avoid across-table JOIN operations during service reading and cross-shard operations for any individual transaction.
- Include the sharding key into query conditions to avoid scanning all sharded tables.

2.2 How Can I Rectify a Fault in Accessing DDM by Using the JDBC Driver?

When you access DDM using the MySQL driver (JDBC) in load balancing mode, an infinite loop may occur during connection switchover, resulting in stack overflow.

Fault Locating

1. Query the application logs and locate the fault cause.

For example, the following logs show that the fault is caused by stack overflow.

```
Caused by: java.lang.StackOverflowError
  at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
  at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
  at java.nio.charset.CharsetEncoder.encode(CharsetEncoder.java:795)
  at java.nio.charset.Charset.encode(Charset.java:843)
  at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2362)
  at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2344)
  at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:568)
  at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:626)
  at com.mysql.jdbc.Buffer.writeStringNotNull(Buffer.java:670)
  at com.mysql.jdbc.MySqlIO.sqlQueryDirect(MySqlIO.java:2636)
```

2. Analyze the overflow source.

For example, the following logs show that the overflow results from an infinite loop inside the driver.

```
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:
344)
at
com.mysql.jdbc.LoadBalancedAutoCommitInterceptor.postProcess(LoadBalancedAutoCommitIntercepto
r.java:104)
at com.mysql.jdbc.MySqlIO.invokeStatementInterceptorsPost(MySqlIO.java:2885)
at com.mysql.jdbc.MySqlIO.sqlQueryDirect(MySqlIO.java:2808)
at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2483)
at com.mysql.jdbc.ConnectionImpl.setReadOnlyInternal(ConnectionImpl.java:4961)
at com.mysql.jdbc.ConnectionImpl.setReadOnly(ConnectionImpl.java:4954)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:381)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:366)
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:
344)
```

3. Query the MySQL version, which is 5.1.44.

According to the source code of the version, when a connection is obtained, **LoadBalance** updates the connection based on the load balancing policy and copies the configurations of the old connection to the new connection. If **AutoCommit** is **true** for the new connection, parameters of the new connection are inconsistent with those of the old connection, and **loadBalanceAutoCommitStatementThreshold** is not configured, an infinite loop occurs. The connection update function calls the parameter synchronization function, and the parameter synchronization function calls the connection update function at the same time, resulting in stack overflow.

Solution

Add the **loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10** parameter to the URL for connecting to DDM.

```
//Connection example when load balancing is used
//jdbc:mysql:loadbalance://ip1:port1,ip2:port2..ipN:portN/{db_name}
String url = "jdbc:mysql:loadbalance://192.168.0.200:5066,192.168.0.201:5066/db_5133?
loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10";
```

- **loadBalanceAutoCommitStatementThreshold** indicates the number of statements executed before a reconnection.

If **loadBalanceAutoCommitStatementThreshold** is set to **5**, a reconnection is initiated after five SQL statements (queries or updates) are executed. A value of **0** indicates a sticky connection, and no reconnection is required. When automatic submission is disabled (**autocommit** is set to **false**), the system waits for the transaction to complete and then determines whether to initiate a reconnection.

2.3 What Version and Parameters Should I Select?

Currently, you cannot connect to DDM using JDBC driver 5.1.46. Versions 5.1.35 to 5.1.45 are recommended.

JDBC driver download address: <https://dev.mysql.com/doc/index-connectors.html>

Table 2-1 describes the recommended parameters for the JDBC URL.

Table 2-1 Parameters

Parameter	Description	Recommended Value
ip:port	Indicates the connection address and port number for connecting to DDM.	Query the connection address from the DDM instance details page.
db_name	Indicates the name of a schema.	Query the schema name on the Schemas page of the DDM instance details page.

Parameter	Description	Recommended Value
loadBalanceAutoCommitStatementThreshold	<p>Indicates the number of statements executed before a reconnection.</p> <ul style="list-style-type: none"> If the parameter value is set to 5, after five SQL statements (queries or updates) are executed, a reconnection is initiated. A value of 0 indicates a sticky connection, and no reconnection is required. <p>When automatic submission is disabled (autocommit is set to false), the system waits for the transaction to complete and then determines whether to initiate a reconnection.</p>	5
loadBalanceHostRemovalGracePeriod	Sets the grace period for removing a host from the load balancing connection.	15000
loadBalanceBlacklistTimeout	Sets the time for retaining a service in the global blacklist.	60000
loadBalancePingTimeout	Indicates the time (unit: ms) for waiting for the ping response of each load balancing connection.	5000
retriesAllDown	<p>Indicates the maximum number of polling retries when all connection addresses fail.</p> <p>If the threshold for retries has been reached but no valid address can be obtained, "SQLException" will be displayed.</p>	10
connectTimeout	<p>Specifies the timeout interval for establishing a socket connection with a database server.</p> <p>Unit: ms. A value of 0 indicates that connection establishment never times out. This parameter setting is used for JDK 1.4 or later versions.</p>	10000
socketTimeout	<p>Specifies the timeout interval for a socket operation (read and write).</p> <p>Unit: ms. A value of 0 indicates that a socket operation never times out.</p>	Set this parameter based on your service requirements.

2.4 Why It Takes So Long Time to Export Data from MySQL Using mysqldump?

The version of the mysqldump client may be inconsistent with that of the supported MySQL server.

Using the same version of the mysqldump client and MySQL server is recommended.

2.5 How Should I Handle the Duplicate Primary Key Error Occurring After Data Is Imported into DDM?

When creating a table, set the start value for automatic increment and ensure that the start value is greater than the maximum auto-increment value of imported data.

2.6 What Should I Do If an Error Message Is Returned After I Specify an Auto-Increment Primary Key?

Execute the following SQL statement to modify the start value of the auto-increment primary key so that the value is greater than the maximum value of primary keys in existing tables:

```
ALTER SEQUENCE Schema name.SEQ name START WITH New start value
```

2.7 How Do I Handle the Error Reported When Parameter Configuration Does Not Time Out?

Adjust the **SocketTimeOut** value or leave this parameter blank. The default value is 0, indicating that the client is not disconnected.

2.8 Which Should I Delete First, Schema or Associated RDS DB Instances?

You should delete a schema first and then its associated RDS DB instances. If you delete the associated RDS DB instances before deleting the schema, the schema cannot be deleted. You need to delete it by deleting the associated DDM instance.

2.9 Should I Manually Delete Databases and Accounts Remained in the Associated RDS DB Instances After a Schema Is Deleted?

If you do not need to delete the databases or accounts, you can manually delete them to free up space.

3 SQL Syntax

3.1 Does DDM Support Cross-Database Access of SQL?

DDM does not support cross-schema access of SQL statements which contain database names.

DDM automatically deletes the database names in SQL statements, for example, the statement **select * from dn1.item** will be automatically changed to **select * from item**.

3.2 Does DDM Support Distributed JOIN?

DDM supports distributed JOIN.

- Redundant fields are added during table design.
- Cross-shard JOIN is implemented by using broadcast tables, ER shards, and ShareJoin.
- Currently, DDM does not allow cross-database update or deletion of multiple tables.

3.3 How Do I Optimize SQL Statements?

- You are advised to use INNER instead of LEFT JOIN or RIGHT JOIN.
- When LEFT JOIN or RIGHT JOIN is used, ON is preferentially executed, and WHERE is executed at the end. Therefore, when using LEFT JOIN or RIGHT JOIN, ensure that the conditions are judged in the ON statement to reduce the execution of WHERE.
- When possible, use JOIN instead of subqueries to avoid full scanning of large tables.

3.4 Does DDM Support Forced Data Type Conversion?

Data type conversion is an advanced function. DDM will be gradually improved to be compatible with more SQL syntax. If necessary, submit a service ticket for processing.

3.5 What Should I Do If an Error Is Reported When Multiple Data Records Are Inserted into Batches Using the INSERT Statement?

Solution

Split the INSERT statement into multiple small statements, or configure parameters on the console, modify the value of **max_allowed_packet**, and restart the instance to have the modification take effect.

4 RDS-related Questions

4.1 Is the Name of a Database Table Case-Sensitive?

By default, DDM and RDS MySQL are case-insensitive to data table names and sequence names.

4.2 What Risky Operations on RDS for MySQL Will Affect DDM?

[Table 4-1](#) lists risky operations on RDS for MySQL.

Table 4-1 Risky operations on RDS for MySQL

Operation Type	Operation	Impact of the Operation
Operations on the RDS console	Deleting an RDS MySQL DB instance	After an RDS MySQL DB instance is deleted, all schemas and logical tables of the DDM instance associated with the DB instance unavailable.
	Performing the primary/standby switchover of an RDS MySQL DB instance	<p>RDS may be intermittently interrupted during the primary/standby switchover. In addition, a small amount of data may be lost in case of long delay in primary/standby synchronization.</p> <ul style="list-style-type: none"> • Creating schemas or logical tables or scaling out schemas is not allowed on DDM during the primary/standby switchover of the RDS MySQL DB instance. • After a primary/standby switchover of an RDS MySQL instance, the RDS MySQL instance ID in DDM remains unchanged.

Operation Type	Operation	Impact of the Operation
	Restarting an RDS MySQL DB instance	An RDS MySQL DB instance being restarted is unavailable, and the DDM service will be affected.
	Resetting a password	After the RDS DB instance password is reset, enter the changed password on DDM when creating a schema.
	Modifying a parameter template	The following parameters are set to fixed values. If their values are changed, DDM will not function properly. <ul style="list-style-type: none"> ● lower_case_table_names: Set this parameter to 1, indicating that data table names and sequence names are case-insensitive. ● local_infile: Set this parameter to ON in scale-out scenarios.
	Modifying a security group	The DDM instance cannot connect to the RDS MySQL DB instance.
	Modifying a VPC	The DDM instance and RDS MySQL DB instance cannot communicate with each other if they are in different VPCs.
	Restoring data	Restoring data may damage data integrity.
Operations through an RDS MySQL client	Deleting a physical database created on DDM	After a physical database is deleted, the original data will be lost and new data cannot be written into the database.
	Deleting a physical account created on DDM	After a physical account is deleted, logical tables cannot be created on DDM.
	Deleting a physical table created on DDM	After a logical table is deleted, data will be lost on DDM, and the table cannot be used on DDM.

Operation Type	Operation	Impact of the Operation
	Modifying the name of a logical table created on DDM	After the name of a logical table is modified, DDM cannot obtain the table data, and the table cannot be used on DDM.
	Changing a record	Changing a record in a broadcast table will affect the data consistency of shards.
	Modifying a whitelist	The DDM instance cannot access the RDS MySQL DB instance if the DDM instance is not in the DB instance whitelist.

4.3 How Do I Handle Data with Duplicate Primary Keys in a Table?

Scenario 1

The logical table of a DDM instance contains a data record whose type value of the primary key is a boundary value. When you enter a data record with the value exceeding the required range, data with duplicate primary keys is displayed.

Procedure

- Step 1** Log in to the RDS console.
- Step 2** On the **Instance Management** page, locate the target RDS MySQL DB instance and click its name.
- Step 3** On the **Basic Information** page, choose **Parameters** in the left pane.
- Step 4** Click the **Parameters** tab and enter **sql_mode** in the text box. Then click the expanding button in the **Value** column, select **STRICT_ALL_TABLES** or **STRICT_TRANS_TABLES**, and click **Save**.

NOTE

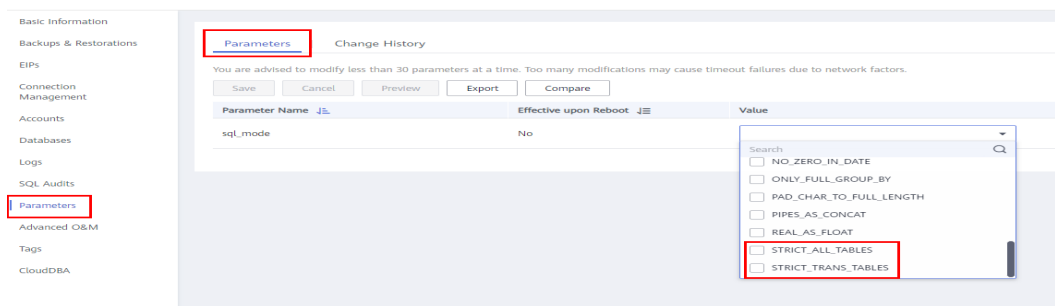
STRICT_ALL_TABLES and **STRICT_TRANS_TABLES** are both strict modes. The strict mode controls how MySQL handles invalid or missing values.

- An invalid value might have the wrong data type for the column, or might be out of range.
- A value is missing when a new row to be inserted does not contain a value for a non-NULL column that has no explicit DEFAULT clause in its definition.
- If the DDM instance version is earlier than 2.4.1.3, do not set **sql_mode** to **ANSI_QUOTES**. If you set it to **ANSI_QUOTES**, double quotation marks used for each string will be translated into an identifier during SQL execution, making the string invalid.

For example, **logic** in **select * from test where tb = "logic"** cannot be parsed correctly.

For more information about SQL modes, see [Server SQL Modes](#).

Figure 4-1 Modifying instance parameters



Step 5 On the **Instances** page, restart the DDM instance.

----End

Scenario 2

For sharded tables (**hash\range\mod**) with composite primary keys, the primary key is repeated when you insert **0** and **1** to data with the sharding key.

Procedure

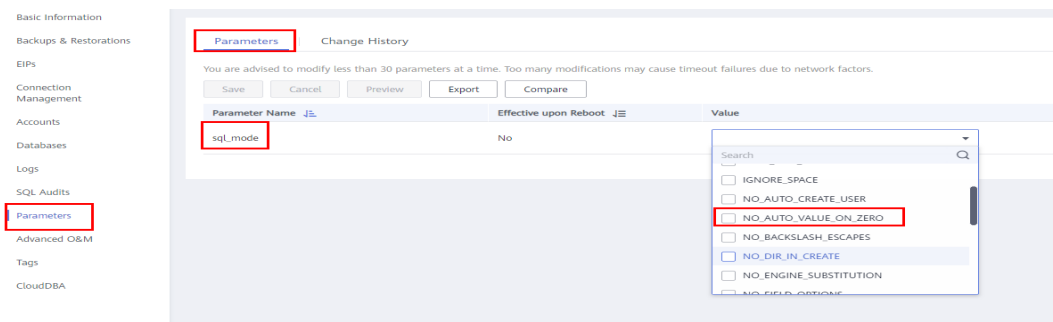
Step 1 Log in to the RDS console.

Step 2 On the **Instance Management** page, locate the target RDS MySQL DB instance and click its name.

Step 3 On the **Basic Information** page, choose **Parameters** in the left pane.

Step 4 Click the **Parameters** tab and enter **sql_mode** in the text box. Then click the expanding button in the **Value** column, select **NO_AUTO_VALUE_ON_ZERO** in the drop-down list, and click **Save**.

Figure 4-2 Modifying instance parameters



Step 5 On the **Instances** page, restart the DDM instance.

----End

4.4 How Can I Query RDS Information by Running Command `show full innodb status`?

After connecting to a DDM instance through the MySQL client, you can run command **show full innodb status** to query information about the associated RDS MySQL DB instances. The following information can be queried:

- Current time and duration since the last output.
- Status of the master thread.
- SEMAPHORES including event counts and available waiting threads when there is high-concurrency workload. You can use the information to locate performance bottlenecks if any.

5 Connection Management

5.1 How Do I Connect to a DDM Instance?

DDM is compatible with the MySQL protocol and allows you to access DDM instances through the MySQL client.

Prerequisites

The ECS where the client is installed should be in the same VPC and subnet as those of the target DDM instance.

Procedure

- Step 1** Log in to the target ECS.
- Step 2** Download and install the MySQL client.
- Step 3** View and obtain the connection information about the target DDM instance.
- Step 4** Connect to the DDM instance.
- Step 5** Perform operations on the DDM instance by executing MySQL commands.
- Step 6** View the details.

----End

5.2 Can I Connect to DDM Instances from My Local Environment?

You cannot connect to a DDM instance in your local environment except in the following scenarios:

1. DDM instances use Virtual Private Clouds (VPCs) to manage security of services. The DDM instances that you create can only be accessed by ECSs in the same VPC as your instances.

2. You can use an ECS that can communicate with your instance to forward your requests, to connect to a DDM instance from your local environment.
3. In addition, you can also connect your local environment to a DDM instance through a direct connection. To connect to the DDM instance with this method, you need to enable port number **5066** in inbound and outbound directions of the required security group.

5.3 What Should I Do If Garbled Characters Are Displayed When I Connect to DDM Using MySQL?

If the MySQL connection code is inconsistent with the actual one, garbled characters may be displayed during parsing on DDM.

In this case, configure **default-character-set=utf8** to specify the encoding system.

Example:

```
mysql -h127.0.0.1 -P5066 -Dbase --default-character-set=utf8 -uddmuser -p
```

A Change History

Released On	Description
2020-10-20	This issue is the third official release. Added FAQs about RDS.
2020-08-07	This issue is the second official release. Added FAQs about DDM usage.
2020-04-30	This issue is the first official release.