Data Replication Service

Troubleshooting

Issue	01
Date	2022-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Solutions to Failed Check Items

1.1 Disk Space

1.1.1 Checking Whether the Destination Database Has Sufficient Storage Space

MySQL Migration

Table 1-1 Checking whether the destination database has sufficient storage space

Check Item	Whether the destination database has sufficient storage space
Description	Check whether the destination database has sufficient storage space. If storage space is insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: The destination database does not have sufficient storage space. The destination database must have at least * GB of free space. Handling suggestion: Scale up or clean up the destination database storage space. If you clean up the storage space, you will obtain more space within 2 to 3 minutes. **NOTE** It is recommended that the size of the destination database disk be set to the smaller value of the following two values: 1. 2.5 times the size of the data to be migrated in the source database. 2. The size of the data to be migrated in the source database plus 200 GB.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

PostgreSQL Synchronization

Table 1-2 Checking whether the destination database has sufficient storage space

Check Item	Whether the destination database has sufficient storage space
Description	Check whether the destination database has sufficient storage space. If storage space is insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The destination database does not have sufficient storage space. The destination database must have at least * GB of free space. Handling suggestion: Scale up or clean up the destination database storage space. If you clean up the storage space, you will obtain more space within 2 to 3 minutes. **NOTE** It is recommended that the size of the destination database disk be set to the smaller value of the following two values: 1. 1.5 times the size of the data to be migrated in the source database. 2. The size of the data to be migrated in the source database plus 200 GB.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

MongoDB Migration

Table 1-3 Checking whether the destination database has sufficient storage space

Check Item	Whether the destination database has sufficient storage space
Description	Check whether the destination database has sufficient storage space. If storage space is insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The destination database does not have sufficient storage space. The destination database must have at least * GB of free space. Handling suggestion: Scale up or clean up the destination database storage space. If you clean up the storage space, you will obtain more space within 2 to 3 minutes. **NOTE** It is recommended that the size of the destination database disk be set to the smaller value of the following two values: 1. 1.5 times the size of the data to be migrated in the source database. 2. The size of the data to be migrated in the source database plus 200 GB.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.1.2 Checking Whether the Destination Server Has Sufficient Storage Space

Table 1-4 Checking whether the destination server has sufficient storage space

Check Item	Whether the destination server has sufficient storage space
Description	If the destination server's storage space is insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The amount of data in the source database is greater than the remaining storage space of the destination server. Handling suggestion: Modify the synchronization object.

1.2 Database Parameters

1.2.1 Checking Whether the Source Database Binlog Is Enabled

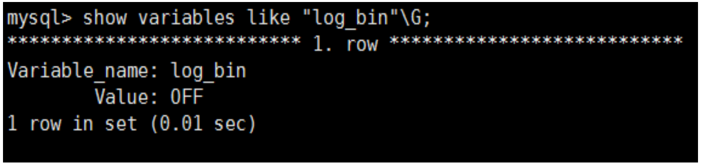
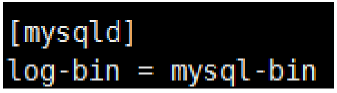
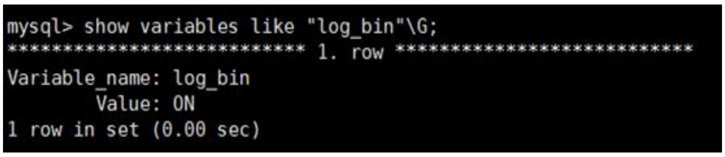
MySQL Migration

Table 1-5 Checking whether the source database binlog is enabled

Check Item	Whether the source database binlog is enabled
Description	Check whether binlog is enabled for the source database.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements. **NOTE** For details about the MySQL permissions and authorization operations required by DRS, see **Which MySQL Permissions Are Required for DRS?**
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Failure cause: The binlog function is disabled on the source database.

Handling suggestion:

- If the source is an on-premises database, perform the following operations to enable binlog.
  1. Run the following command to check whether binlog is enabled:
     show variables like "log_bin"\G;
     
     ```
     mysql> show variables like "log_bin"\G;
     *************************** 1. row ***************************
     Variable_name: log_bin
             Value: OFF
     1 row in set (0.01 sec)
     ```
  2. If binlog is disabled, add **log-bin = mysql-bin** followed by **[mysqld]** in the MySQL configuration file **my.cnf** or **my.ini**.
     
     ```
     [mysqld]
     log-bin = mysql-bin
     ```
  3. Restart the database.
     
     ```
     mysql> show variables like "log_bin"\G;
     *************************** 1. row ***************************
     Variable_name: log_bin
             Value: ON
     1 row in set (0.00 sec)
     ```
- If the source is an RDS instance, enable binlog by referring to the RDS official documentation.

1.2.2 Checking Whether the Source Database Binlog Is Row-Based

MySQL Migration

Table 1-6 Checking whether the source database binlog is row-based

Check Item	Whether the source database binlog is row-based
Description	Check whether the source database binlog is row-based.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: The source database binlog is not row-based.
	Handling suggestion:
	● If the source database is an on-premises database, perform the following operations to change the binlog format of the source database:
	Method 1: You can modify the **my.cnf** or **my.ini** configuration file and restart the database.
	`binlog_format=row`
	Method 2: Stop all service connections.
	`set global binlog_format='ROW'`
	Modify the **my.cnf** or **my.ini** configuration file.
	`binlog_format=row`
	In the **ROW** format, the log growth rate increases, which may occupy more disk space.
	NOTE
	The MySQL **Global binlog_format** parameter does not take effect for connected sessions. For details, see **How Do I Set Global binlog_format=ROW to Take Effect Immediately?**
	● If the source database is an RDS DB instance on the cloud, change the **binlog_format** value of the source database to **ROW**. After the change, restart the database for the change to take effect.
	NOTE
	The MySQL **Global binlog_format** parameter does not take effect for connected sessions. For details, see **How Do I Set Global binlog_format=ROW to Take Effect Immediately?**

	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.2.3 Checking Whether the Binlog Retention Period Is Set on the Source Database

MySQL Migration

Table 1-7 Checking whether the binlog retention period is set on the source database

Check Item	Whether the binlog retention period is set on the source database
Description	Checking whether the binlog retention period is set on the source database. You are advised to store the source database binlog for a longer time, if the storage space is sufficient.

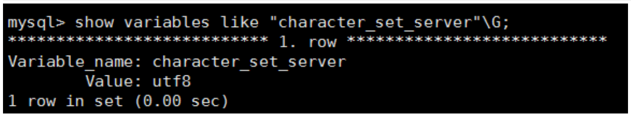
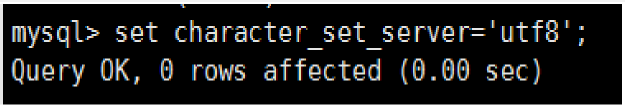
Failure Cause and Handling Suggestion	Failure cause: The binlog retention period is not set on the source database. Handling suggestion: Log in to the source database and run the following SQL statement to set the retention period of binlog: `call mysql.rds_set_configuration('binlog retention hours', n);` The value **n** indicates an integer from 1 to 168.

1.2.4 Checking Whether the Source and Destination Database Character Sets Are Consistent

MySQL Migration

Table 1-8 Checking whether the source and destination database character sets are consistent

Check Item	Whether the source and destination database character sets are consistent
Description	Check whether the character sets of the servers hosting the source and destination databases are consistent.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: The character set of the source database is inconsistent with that of the destination database. Handling suggestion: Modify the character sets. Run commands to modify the self-created source database. 1. Check whether source and destination database character sets are consistent. **show variables like "character_set_server"\G;**  mysql> show variables like "character_set_server"\G; *************************** 1. row *************************** Variable_name: character_set_server Value: utf8 1 row in set (0.00 sec) 2. Modify the character set of the source database server. **set character_set_server='utf8';**  mysql> set character_set_server='utf8'; Query OK, 0 rows affected (0.00 sec)
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

MySQL -> PostgreSQL Synchronization

Table 1-9 Checking whether the source and destination database character sets are consistent

Check Item	Whether the source and destination database character sets are consistent
Description	Check whether the character sets of the servers hosting the source and destination databases are consistent.
Failure Cause and Handling Suggestion	Failure cause: The character set of the source database is inconsistent with that of the destination database. Handling suggestion: Change the character set **character_set_server** of the source database to the same as that of the destination database, or change the character set **server_encoding** of the destination database to the same as that of the source database.

1.2.5 Checking Whether the Source Database server_id Meets the Incremental Migration Requirements

MySQL

Table 1-10 Checking whether the source database server_id meets the incremental migration requirements

Check Item	Whether the source database **server_id** meets the incremental migration requirements
Description	Check whether the source database **server_id** meets the incremental migration requirements.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database failed to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The source database **server_id** does not meet the incremental migration requirements. Handling suggestion: Run the following command to modify the **server_id** value: **set global server_id=n** The value **n** indicates the source database server_id. If the source database version is MySQL 5.6, the value **n** ranges from 2 to 4294967296. Otherwise, the value **n** ranges from 1 to 4294967296.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.2.6 Checking Whether the Source and Destination Database Table Names Are Consistent in Case Sensitivity

MySQL Migration

Table 1-11 Checking whether the source and destination database table names are consistent in case sensitivity

Check Item	Whether the source and destination database table names are consistent in case sensitivity
Description	Check whether the source and destination database names and table names are consistent in case sensitivity.

Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The **lower_case_table_names** values in the source and destination databases must be the same. Handling suggestion: ● If you are migrating data out of the cloud, change the values of **lower_case_table_names** in the source and destination databases to the same. You are advised to change the parameter value in an empty database. For example, if the destination RDS DB instance is empty, run the following example command to change the **lower_case_table_names** value to the same as that in the source database: Sample command: `set global lower_case_table_names=n;` In the preceding command, **n** indicates the parameter value of the source database. After the modification, restart the database for the modification to take effect. ● If you are migrating data out of the cloud, perform the following operations: If the destination database is a self-built database, modify the **lower_case_table_names** parameter of the destination database. Add **lower_case_table_names=n** under the **[mysqld]** tag in the MySQL configuration file **my.cnf**. **n** indicates the value of parameter **same lower_case_table_names** of the source database. The database must be restarted to make the change take effect. If the destination database is a cloud database, check whether the **lower_case_table_names** parameter can be modified. If not, contact Huawei technical support.
	Failure cause: The **lower_case_table_names** parameter value of the destination database is different from that of the source database, and the source database contains uppercase database and table names. Handling suggestion: Rectify the fault by referring to **FAQs**.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

MySQL->MySQL, MySQL->Gauss(for MySQL) and Gauss(for MySQL) to MySQL Synchronization

Table 1-12 Checking whether the source and destination database table names are consistent in case sensitivity

Check Item	Whether the source and destination database table names are consistent in case sensitivity
Description	The destination database is case insensitive. The names of the mapped databases and tables contain uppercase letters. If the destination database is case-insensitive, all uppercase letters are converted to lowercase letters for storage.
Failure Cause and Handling Suggestion	Failure cause: The destination database is case insensitive. All uppercase letters are converted to lowercase letters for storage. The names of the mapped databases and tables contain uppercase letters. Handling suggestion: If the mapping relationship is correct, change the database and table names that contain uppercase letters to lowercase letters.

1.2.7 Checking Whether the Source Database GTID Is Enabled

MySQL

Table 1-13 Checking whether the source database GTID is enabled

Check Item	Whether the source database GTID is enabled
Description	The source database GTID must be enabled.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database GTID is disabled. To recover faulty migration tasks and destination databases more securely and efficiently, you are advised to enable GTID. Note that switching primary and standby DB instances may lead to the failure of the migration task.
	Handling suggestion:
	● If the source database version is MySQL 5.5, ignore this alarm.
	● If the source database version is MySQL 5.6 and later, set the following parameters to enable GTID in the database configuration file. Then, restart the database for the modifications to take effect.
	`gtid_mode = on`
	`log_slave_updates = true`
	`enforce_gtid_consistency = on`

1.2.8 Checking Whether the Source Database Contains Object Names with Non-ASCII Characters

MySQL

Table 1-14 Checking whether the source database contains object names with non-ASCII characters

Check Item	Whether the source database contains object names with non-ASCII characters
Description	If the source database contains object names with non-ASCII characters, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database cannot contain object names with non-ASCII characters. Handing suggestion: In the source database, change the object names containing non-ASCII characters.

1.2.9 Checking Whether the Source and Destination Database TIME_ZONE Values Are Consistent

MySQL

Table 1-15 Checking whether the TIME_ZONE values of the source and destination databases are the same

Check Item	Whether the **TIME_ZONE** values of the source and destination databases are the same
Description	The migration fails because the **TIME_ZONE** values of the source and destination databases are different.
Failure Cause and Handling Suggestion	Failure cause: The **TIME_ZONE** or **SYSTEM_TIME_ZONE** values of the source and destination databases must be the same. Handling suggestion: Change the **TIME_ZONE** value of the destination database to the same as that of the source database, or change the **TIME_ZONE** value of the source database to the same as that of the destination database.

1.2.10 Checking Whether the COLLATION_SERVER Values of the Source and Destination Databases Are the Same

MySQL

Table 1-16 Checking whether the COLLATION_SERVER values of the source and destination databases are the same

Check Item	Whether the **COLLATION_SERVER** values of the source and destination databases are the same
Description	The migration fails because the **COLLATION_SERVER** values of the source and destination databases are different.
Failure Cause and Handling Suggestion	Failure cause: The **COLLATION_SERVER** values of the source and destination databases must be the same. Handling suggestion: Change **COLLATION_SERVER** of the source and destination databases to the same value.

1.2.11 Checking Whether the SERVER_UUID Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 1-17 Checking whether the SERVER_UUID values of the source and destination databases are the same

Check Item	Whether the **SERVER_UUID** values of the source and destination databases are the same
Description	If the **SERVER_UUID** values of the source and destination databases are the same, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The **SERVER_UUID** values of the source and destination databases must be different. Handling suggestion: Check that the source and destination databases are not the same MySQL database.

1.2.12 Checking Whether the SERVER_ID Values of the Source and Destination Databases Are Different

MySQL

Table 1-18 Checking whether the SERVER_ID values of the source and destination databases are different

Check Item	Whether the **SERVER_ID** values of the source and destination databases are different
Description	Check whether the **SERVER_ID** values of the source and destination databases are different. If they are the same, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The **SERVER_ID** values of the source and destination databases must be different. Handling suggestion: Change **SERVER_ID** of the source and destination databases to different values.

1.2.13 Checking Whether the Source Database Contains Invalid sql_mode Values

MySQL

Table 1-19 Checking whether the source database contains invalid sql_mode values

Check Item	Whether the source database contains invalid **sql_mode** values
Description	If the source database contains invalid **sql_mode** values, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The **sql_mode** value of the source database cannot be **no_engine_substitution**. Handling suggestion: Change **sql_mode** of the source database to a proper value.

1.2.14 Checking Whether the SQL_MODE Values of the Source and Destination Databases Are the Same

MySQL

Table 1-20 Checking whether the SQL_MODE values of the source and destination databases are the same

Check Item	Whether the **SQL_MODE** values of the source and destination databases are the same
Description	Check whether the **SQL_MODE** values of source and destination databases are the same. If they are inconsistent, the migration may fail.
Failure Cause and Handling Suggestion	● If you are migrating data to the cloud, perform the following operations: Failure cause: The **SQL_MODE** values of the source and destination databases must be the same. Handling suggestion: Change the **SQL_MODE** values of the destination database to the same as those of the source database, Ensure that both the source and destination databases do not have the forbidden SQL_MODE values. For details, see **Modifying Parameters**. If MyISAM tables are to be migrated, the **SQL_MODE** values in the destination database cannot contain **NO_ENGINE_SUBSTITUTION**. ● If you are migrating data out of the cloud, perform the following operations: Item to be confirmed: The **SQL_MODE** values of the source and destination databases must be the same. Handling suggestions: Change the **SQL_MODE** values of the destination database to the same as those of the source database. Ensure that both the source and destination databases do not have the forbidden **SQL_MODE** values.

1.2.15 Checking Whether the sql_mode Value in the Destination Database Is Not NO_ENGINE_SUBSTITUTION

MySQL Migration and Synchronization

Table 1-21 Checking whether the sql_mode value in the destination database is not NO_ENGINE_SUBSTITUTION

Check Item	Whether the sql_mode value in the destination database is not NO_ENGINE_SUBSTITUTION

Description	MyISAM tables are included in the migration objects. The sql_mode value in the destination database cannot be NO_ENGINE_SUBSTITUTION.
Failure Cause and Handling Suggestion	Failure cause: The **sql_mode** value in the destination database is **NO_ENGINE_SUBSTITUTION**. Handling suggestion: In the destination database, set **SQL_MODE** to a value other than NO_ENGINE_SUBSTITUTION. For details, see **Modifying Parameters** in the *Relational Database Service User Guide*.

1.2.16 Checking Whether the innodb_strict_mode Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 1-22 Checking whether the innodb_strict_mode values of the source and destination databases are the same

Check Item	Whether the **innodb_strict_mode** values of the source and destination databases are the same
Description	Check whether the **innodb_strict_mode** values of source and destination databases are the same. If they are inconsistent, the migration may fail.
Failure Cause and Handling Suggestion	● If you are migrating data to the cloud, perform the following operations: Failure cause: The **innodb_strict_mode** values of the source and destination databases must be the same. Handling suggestion: Create a parameter group for the destination database and change **innodb_strict_mode** to the same value as that of the source database. For details, see **Creating a Parameter Group** in *Relational Database Service User Guide*. ● If you are migrating data out of the cloud, perform the following operations: Failure cause: The **innodb_strict_mode** values of the source and destination databases must be the same. Handling suggestion: Change **innodb_strict_mode** of the destination database to the same value as that of the source database.

1.2.17 Checking Whether the max_wal_senders Value of the Source Database Is Correctly Configured

PostgreSQL Synchronization

Table 1-23 Checking whether the max_wal_senders value of the source database is correctly configured

Check Item	Whether the **max_wal_senders** value of the source database is correctly configured
Description	The **max_wal_senders** value of the source database must be greater than the number of used replication slots. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **max_wal_senders** value of the source database is less than or equal to the number of used replication slots. Handling suggestion: Set **max_wal_senders** to a value greater than the number of used replication slots and restart the database to apply the changes. Run the following command to query the number of used replication slots in the current database: `select count(1) from pg_replication_slots;`

1.2.18 Checking Whether the WAL_LEVEL Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 1-24 Checking whether the WAL_LEVEL value in the source database is correct

Check Item	Whether the WAL_LEVEL value in the source database is correct
Description	Check whether **wal_level** of the source database is set to **logical**. If the value is not **logical**, the incremental logs of the source database cannot be logically decoded. As a result, incremental synchronization cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The **wal_level** value in the source database is incorrect. Handling suggestion: Change the **wal_level** value of the source database to **logical**. For details about how to modify the parameter for self-built databases, see: ● Run **alter system set wal_level = logical** in the source database as a super user and restart the database to apply the changes. ● Alternatively, modify the **postgresql.conf** configuration file, set **wal_level** to **logical**, and restart the database to apply the changes.

	Failure cause: The source database version is not supported. Handling suggestion: Ensure that the source database version is supported by DRS. Supported source database versions include PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12, 13, and RDS for PostgreSQL Enhanced Edition.
	Failure cause: The destination database version is not supported. Handling suggestion: Ensure that the destination database version is supported by DRS. The destination database supports the following major versions: RDS for PostgreSQL 9.5, 9.6, 10, 11, 12, 13, and Enhanced Edition. If the source database is RDS for PostgreSQL Enhanced Edition, the destination database supports only RDS for PostgreSQL Enhanced Edition.

1.2.19 Checking Whether the MAX_REPLICATION_SLOTS Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 1-25 Checking whether the MAX_REPLICATION_SLOTS value in the source database is correct

Check Item	Whether the **MAX_REPLICATION_SLOTS** value in the source database is correct
Description	The **max_replication_slots** value of the source database must be greater than the number of used replication slots. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **max_replication_slots** value of the source database is less than or equal to the number of used replication slots. Handling suggestion: Set **max_replication_slots** to a value greater than the number of used replication slots and restart the database to apply the changes. Run the following command to query the number of used replication slots in the current database: `select count(1) from pg_replication_slots;`
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the synchronization requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.2.20 Checking Whether the Source Database Is on Standby

PostgreSQL Synchronization

Table 1-26 Checking whether the source database is on standby

Check Item	Whether the source database is on standby
Description	For a full+incremental synchronization task, the source database cannot be a standby database. Otherwise, incremental synchronization cannot be performed. For a full synchronization task, the source database can be a standby database, but **hot_standby_feedback** must be set to **on**. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: In a real-time full+incremental synchronization task, the source database cannot be a standby database. Otherwise, incremental synchronization cannot be performed. Handling suggestion: Configure the source database as the primary database.
	Failure cause: For a full synchronization task, the source database **hot_standby_feedback** is set to **off**. Handling suggestion: The source database is configured as the primary database, or **hot_standby_feedback** of the source database is set to **on**. ● Change the source database to the primary database. ● Alternatively, change the **hot_standby_feedback** value of the source database to **on** before starting full synchronization. After the full synchronization is complete, change the value of this parameter to **off**.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.2.21 Checking Whether the log_slave_updates Value of the Source Database Is Correctly Configured

MySQL Migration, Synchronization, and Disaster Recovery

Table 1-27 Checking whether the log_slave_updates value of the source database is correctly configured

Check Item	Whether the **log_slave_updates** value of the source database is correctly configured

Description	The migration will fail if the **log_slave_updates** parameter of the source database is disabled.
Failure Cause and Handling Suggestion	Failure cause: The **slave_updates_check** parameter of the source database must be enabled. Handling suggestion: In the MySQL configuration file **my.cnf**, add the "log_slave_updates=1" line under [mysqld] and restart the database for the modification to take effect.
	Failure cause: The source database is a standby database and the **log_slave_updates** value is **OFF**. Handling suggestion: On the source database, set **log_slave_updates** to **ON**. Then, restart the database for the modification to take effect.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database is a standby database and the **log_slave_updates** value is **OFF**. Handling suggestion: On the source database, set **log_slave_updates** to **ON**. Then, restart the database for the modification to take effect. If no switchover or failover will occur, no operation is required.

1.2.22 Checking Whether the BLOCK_SIZE Value of the Source Database Is the Same as That of the Destination Database

PostgreSQL Synchronization

Table 1-28 Checking whether the BLOCK_SIZE value of the source database is the same as that of the destination database

Check Item	Whether the **BLOCK_SIZE** value of the source database is the same as that of the destination database
Description	The **BLOCK_SIZE** value of the destination database must be greater than or equal to that of the source database. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **BLOCK_SIZE** value of the destination database is less than that of the source database. Handling suggestion: ● Use the destination database whose **BLOCK_SIZE** value is greater than or equal to that of the source database. ● Use the source database whose **BLOCK_SIZE** value is less than or equal to the value of destination database **BLOCK_SIZE**.

1.2.23 Checking Whether the binlog_row_image Value is FULL

MySQL

Table 1-29 Checking whether the binlog_row_image value is FULL

Check Item	Whether the **binlog_row_image** value is **FULL**
Description	If the **binlog_row_image** value of the source database is not **FULL**, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The **binlog_row_image** value of the source database is not **FULL**. Handling suggestion: ● If the source database is an RDS DB instance on the cloud, change **binlog_row_image** to **FULL** on the RDS console, and then restart the source database. ● If the source database is an on-premises database, perform the following steps: 1. Log in to the server where the MySQL source database is located. 2. Manually change the value of **binlog_row_image** in the **my.cnf** configuration file to **FULL** and save the file. `binlog_row_image=full` 3. To ensure a successful task, restart the source database during off-peak hours.

1.2.24 Checking Whether the Transaction Isolation Levels are Consistent

MySQL

Table 1-30 Checking whether the transaction isolation levels are consistent

Check Item	Whether the transaction isolation levels are consistent
Description	Check whether the transaction isolation levels of the source and destination databases are the same.

Failure Cause and Handling Suggestion	If you are migrating data to the cloud, perform the following operations: **Failure cause**: The transaction isolation levels of the source and destination databases are different. **Handling suggestion**: Change the isolation level (**tx_isolation** or **transaction_isolation**) of the destination database to be the same as that of the source database.

1.2.25 Checking Whether the lc_monetary Values of the Source and Destination Databases Are the Same

PostgreSQL Synchronization

Table 1-31 Checking whether the lc_monetary values of the source and destination databases are the same

Check Item	Whether the **lc_monetary** values of the source and destination databases are the same
Description	Check whether the **lc_monetary** values of the source and destination databases are the same. If they are inconsistent, the synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database failed to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database failed to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The **lc_monetary** values of the source and destination databases must be the same. Handling suggestion: Check whether the **lc_monetary** values of the source and destination databases meet the synchronization requirements.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the synchronization requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.2.26 Checking Whether the Source Database Contains Trigger Names with Non-ASCII Characters

MySQL

Table 1-32 Checking whether the source database contains trigger names with non-ASCII characters

Check Item	Whether the source database contains trigger names with non-ASCII characters
Description	If the source database contains non-ASCII characters, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database cannot contain view names with non-ASCII characters. Handling suggestion: To solve this problem, perform the following steps: Method 1: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the trigger name that contains non-ASCII characters. Method 2: Change the trigger name.

1.2.27 Checking Whether the Source Database Collections Contain More Than 10 Indexes

MongoDB Migration

Table 1-33 Checking whether the source database collections contain more than 10 indexes

Check Item	Whether the source database collections contain more than 10 indexes
Description	If the number of indexes in the source database exceeds 10, the migration duration is affected.
Failure Cause and Handling Suggestion	Alarm cause: The source database has collections containing more than 10 indexes, which are migrated slowly. Handling suggestion: The number of indexes affects the migration duration. Check whether all indexes need to be migrated. If the index does not need to be migrated, delete the index before starting the migration. Run the following command to delete the index: **run the db.*Collection name*.dropIndex(*Index name*)**

1.2.28 Checking Whether the Source Database Collections Contain TTL Indexes

MongoDB Migration

Table 1-34 Checking whether the source database collections contain TTL indexes

Check Item	Whether the source database collections contain TTL indexes
Description	Due to inconsistency of time zones and clocks between source and destination databases, migrating TTL indexes will cause data inconsistency.
Failure Cause and Handling Suggestion	Item to be confirmed: Due to inconsistency of time zones and clocks between source and destination databases, migrating TTL indexes will cause data inconsistency. Handling suggestion: If data consistency is required, delete TTL indexes. Alternatively, do not migrate the collections containing TTL indexes. Run the following command to delete the index: **run the db.*Collection name*.dropIndex(*Index name*)**

1.2.29 Checking Whether log_bin_trust_function_creators Is Set to On in Both the Source and Destination Databases

MySQL

Table 1-35 Checking whether log_bin_trust_function_creators is set to on in both the source and destination databases

Check Item	Whether log_bin_trust_function_creators is set to on in both the source and destination databases
Description	During the out-of-cloud migration from MySQL to MySQL, the **log_bin_trust_function_creators** value of the source database must be the same as that of the destination database. If the source database supports user-defined functions (UDFs) but the destination database does not, change the **log_bin_trust_function_creators=off** parameter of the destination database to **log_bin_trust_function_creators=on**. If the parameters of the source and destination are different, the migration may fail.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The destination database does not support custom functions. Handling suggestions: In the **my.cnf** file of the destination database, check whether **log_bin_trust_function_creators=on** exists. If it does not exist, add **log_bin_trust_function_creators=on** and restart the database for the modification to take effect.

1.2.30 Checking Whether GTID Is Enabled for the Source Database

MySQL

Table 1-36 Checking whether GTID is enabled for the source database

Check Item	Whether GTID is enabled for the source database
Description	During disaster recovery, GTID should be enabled for the source database. Otherwise, the migration fails.

Item to Be Confirmed and Handling Suggestion	Failure cause: GTID is disabled for the source database
	Handling suggestion: In the configuration file of the source database, set parameters as followings. Then, restart the source database for the modifications to take effect.
	Parameters to be configured:
	`gtid_mode = on;`
	`log_slave_updates = true;`
	`enforce_gtid_consistency = on;`

1.2.31 Checking Whether GTID Is Enabled for the Destination Database

MySQL Disaster Recovery

Table 1-37 Checking whether GTID is enabled for the destination database

Check Item	Whether GTID is enabled for the destination database
Description	During disaster recovery, GTID should be enabled for the destination database. Otherwise, the migration fails.

Item to Be Confirmed and Handling Suggestion	Failure cause: GTID is disabled for the destination database
	Handling suggestion: Check that the destination database binlog is enabled. Modify the parameter settings configuration file as follows to enable the destination database GTID, and then restart the database
	Sample command:
	gtid_mode = on;
	log_slave_updates = true;
	enforce_gtid_consistency = on;
	Sample command:
	log-bin = mysql-bin
	binlog_gtid_simple_recovery = on.

1.2.32 Checking Whether log_bin_trust_function_creators Is Set to On in the Destination Database

MySQL

Table 1-38 Checking whether log_bin_trust_function_creators is set to on in the destination database

Check Item	Whether **log_bin_trust_function_creators** is set to **on** in the destination database
Description	During the migration from RDS for MySQL to MySQL out of the cloud, the destination database does not support custom functions.
Failure Cause and Handling Suggestion	Failure cause: The destination database does not support custom functions. Handling suggestions: In the **my.cnf** file of the destination database, check whether **log_bin_trust_function_creators=on** exists. If it does not exist, add **log_bin_trust_function_creators=on** and restart the database for the modification to take effect.

1.2.33 Checking Whether the Values in the Source Oracle Database Are Out of the Ranges of the MySQL Database

Oracle Migration

Table 1-39 Checking whether the values in the source Oracle database are out of the ranges of the MySQL database

Check Item	Whether the values in the source Oracle database are out of the ranges of the MySQL database
Description	The values of the following data types in the source Oracle database are out of the ranges of the MySQL database, causing the migration failure.
Failure Cause and Handling Suggestion	**Alarm Information** 1. Ensure that the primary key or unique key column cannot contain values of character string data types when you map the MySQL data types to the character data types in Oracle. Otherwise, data inconsistency or migration failure may occur. 2. Due to differences between Oracle and MySQL databases, the migration will fail if the values of the following data types in the source Oracle database are out of the ranges of MySQL database: number, intl, float, double, date, and timestamp. Handling suggestion: Contact technical support.

1.2.34 Checking Whether the max_allowed_packet Value of the Destination Database Is too Small

MySQL

Table 1-40 Checking whether the max_allowed_packet value of the destination database is too small

Check Item	Whether the max_allowed_packet value of the destination database is too small
Description	A large amount of data cannot be written to the destination database during the migration because the max_allowed_packet value is smaller than 100 MB. As a result, the full migration failed.

Failure Cause and Handling Suggestion	Failure cause: The **max_allowed_packet** value of the destination database is too small, which may cause data fails to be written during the migration. Handling suggestions: Set the **max_allowed_packet** value greater than 100 MB

1.2.35 Checking Whether the Source Database User Has the Permission to Parse Logs

Oracle -> MySQL Migration

Table 1-41 Checking whether the source database user has the permission to parse logs

Check Item	Whether the source database user has the permission to parse logs
Description	If the source database user does not have the log parsing permission, the incremental migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database user does not have the EXECUTE_CATALOG_ROLE role. Handling suggestion: Assign the required role to the user and perform the check again. Run the **GRANT EXECUTE_CATALOG_ROLE TO** *UserName* command to assign the role.
	Failure cause: The source database user does not have the log parsing permission. Handling suggestion: Assign the required role to the user and perform the check again. Run the **GRANT LOGMINING TO** *UserName* command to grant the permission.

1.2.36 Checking Whether the ExpireLogsDays Value Is 0

MySQL Synchronization and Disaster Recovery

Table 1-42 Checking whether the expirelogsdays value is 0

Check Item	Whether the **expirelogsdays** value is 0
Description	If the **expire_logs_days** value of the source database is set to **0**, the migration may fail.

Failure Cause and Handling Suggestion	Failure cause: If **expire_logs_days** is set to **0** in the source database, operations such as startup and flush logs will trigger binlog clearance and result in a migration failure. Handling suggestion: Set the binlog storage duration by running the following command on the client: **set global expire_logs_days=value** (***value*** indicates an integer greater than 0.)

1.2.37 Checking Whether the Source Database Character Set Is Supported

Oracle -> MySQL/Oracle -> PostgreSQL/Oracle-> GaussDB(for MySQL) Migration and Synchronization

Table 1-43 Checking whether the source database character set is supported

Check Item	Whether the source database character set is supported
Description	The source database character set is not supported. A migration task from Oracle only supports the following character sets: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252.
Failure Cause and Handling Suggestion	Failure cause: The source database character set is not supported. Handling suggestion: Change the source database character set, or select a source database with supported character sets.

1.2.38 Checking Whether the Length Table and Field Names Is Not Supported

DDM -> Oracle Synchronization

Table 1-44 Checking whether the length of table and field names is supported

Check Item	Whether the length of table and field names is supported
Description	In synchronization tasks from DDM to Oracle, the lengths of table and field names cannot exceed 30 characters.

Failure Cause and Handling Suggestion	Failure cause: There are tables or fields whose name lengths exceed 30 characters in the source database. Handling suggestion: Change the lengths of the table or field names fewer than 30 characters.

1.2.39 Checking Whether the Databases and Tables Exist

All Scenarios

Table 1-45 Checking whether the databases and tables exist

Check Item	Whether the databases and tables exist
Description	There are databases and tables in the uploaded file that do not exist in the source database. The synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: Objects imported from files do not exist in the source database. Handling suggestion: Remove these objects that do not exist and import the file again.

1.2.40 Checking Whether the Supplemental Log Level of the Source Database Meets Requirements

Oracle Synchronization

Table 1-46 Checking whether the supplemental log level of the source database meets requirements

Check Item	Whether the supplemental log level of the source database meets requirements
Description	The supplemental log level of the source Oracle database does not meet requirements. The synchronization fails.

Failure Cause and Handling Suggestion	Failure cause: The supplemental logging level of the source Oracle database does not meet requirements. Handling suggestion: Perform any of the following operations in the source database: ● Enable all-level (database-level) supplemental logging: **alter database add supplemental log data (all) columns** ● Enable minimal-level supplemental logging: alter database add supplemental log data. Then run the following command to enable all-level (table-level) supplemental logging for each to-be-synchronized table: **alter table TABLE_NAME add supplemental log data(all) columns**

1.2.41 Checking Whether the Length of the Source Database Object Names Exceeds the Limit

MySQL to PostgreSQL and MySQL to GaussDB(DWS) Synchronization

Table 1-47 Checking whether the length of the source database object names exceeds the limit

Check Item	Whether the length of the source database object names exceeds the limit
Description	The destination database object name can contain a maximum of 63 characters. The length of source object names mapped to the destination ones exceeds the upper limit.
Item to Be Confirmed and Handling Suggestion	Failure cause: The destination database object name can contain a maximum of 63 characters. The length of source object names mapped to the destination ones exceeds the upper limit. Handling suggestion: 1. Deselect the objects whose name length exceeds the limit. 2. If the objects are tables, change the table names to meet the mapping conditions.

1.2.42 Checking Whether session_replication_role of the Destination Database Is correctly Set

PostgreSQL Synchronization

Table 1-48 Checking whether the session_replication_role value of the destination database is correctly set

Check Item	Whether the **session_replication_role** value of the destination database is correctly set.
Description	The **session_replication_role** parameter of the destination database is not set to **replica**. Data synchronization may fail when the synchronized table has associated foreign key constraints or triggers.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The **session_replication_role** parameter of the destination database is not set to **replica**. Handling suggestion: Before starting the synchronization task, set **session_replication_role** of the destination database to **replica**. After the synchronization is complete, change the value of this parameter to **origin**. If the destination database is an RDS instance, you can modify the parameter on the RDS console.

1.2.43 Checking the Database Compatibility Type

MySQL -> GaussDB(DWS) Synchronization

Table 1-49 Checking the database compatibility type

Check Item	Whether the database compatibility type is supported
Description	The migration of tables without primary keys does not support the Oracle (ORA) compatibility mode of DWS.
Failure Cause and Handling Suggestion	Failure cause: The migration of tables without primary keys does not support the ORA compatibility mode of GaussDB(DWS). Handling suggestion: Use the MySQL and Teradata compatibility mode of GaussDB(DWS) or add primary keys to tables that do not have primary keys. Run the following statement to add a primary key to the table: CREATE DATABASE mysql_compatible_db DBCOMPATIBILITY 'MYSQL'; Or: CREATE DATABASE td_compatible_db DBCOMPATIBILITY 'TD';

MySQL to GaussDB(for openGauss) Synchronization

Table 1-50 Checking the database compatibility type

Check Item	Whether the database compatibility type is supported
Description	The destination database is incompatible with MySQL.
Failure Cause and Handling Suggestion	Failure cause: The destination database is incompatible with MySQL. Handling suggestion: Use a destination instance that is compatible with MySQL. If the destination instance is a distributed instance, run the following statement to create a compatible database: CREATE DATABASE mysql_compatible_db DBCOMPATIBILITY 'MYSQL'; If the destination instance is a primary/standby instance, run the following statement to create a compatible database: CREATE DATABASE mysql_compatible_db DBCOMPATIBILITY 'B';

1.2.44 Checking Whether the Collation of the Destination Database Is Correct

Oracle->MySQL, Oracle->GaussDB(for MySQL), and Oracle->DDM Synchronization

Table 1-51 Checking whether the collation of the destination database is correct

Check Item	Whether the collation of the destination database is correct
Description	The primary key or unique key in the destination database contains a collation ended in _ci. The collation ending in _ci is case insensitive, so an error indicating duplicate keys may be reported during synchronization and cause the synchronization to fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database collation is not supported. Handling suggestion: Change the destination database collation to a case-sensitive collation (not ending with _ci).

1.2.45 Checking Whether the Source Database Has the Permission to Parse OGG Logs

Oracle -> GaussDB(for openGauss) Synchronization

Table 1-52 Checking whether the source database has the permission to parse OGG logs

Check Item	Whether the source database has the permission to parse OGG logs
Description	To use XStream mode to read logs on the source database, enable ENABLE_GOLDENGATE_REPLICATION on the source database.
Failure Cause and Handling Suggestion	Failure cause: XStream is not enabled on the source database. Handling suggestion: Check whether the database version supports XStream. If yes, run the ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION = TRUE SCOPE=BOTH command in the source database to enable XStream.

1.2.46 Checking Whether the Index Name Is Supported

MySQL -> CSS/ES Synchronization

Table 1-53 Checking whether the index name is supported

Check Item	Whether the index name is supported
Description	Check whether the index name complies with the specifications. The index name rules are as follows: ● Uppercase letters are not allowed. ● Cannot contain /*?"<,>|# and spaces. ● Can contain colons (:). ● Cannot start with a hyphen (-), underscore (_), or plus sign (+). ● Cannot contain periods (.) or ellipsis (...). ● Can contain up to 255 characters.

Failure Cause and Handling Suggestion	Failure cause: The index name is not supported. Handling suggestion: In the migration settings, select another table for synchronization or change the index name for table mapping.

1.2.47 Checking Whether Tables Structures Are Consistent

MySQL -> CSS/ES Synchronization

Table 1-54 Checking whether tables structures are consistent

Check Item	Whether tables structures are consistent
Description	Check whether the table structures at the source end and destination are consistent.
Failure Cause and Handling Suggestion	Failure cause: The source table columns are not a subset of the destination indexes. Handling suggestion: Return to the **Set Synchronization Task** page, select the tables that meet the requirements. Alternatively, create indexes required in the destination table.

1.2.48 Checking Whether Existing Data Meets the Constraints

Oracle Synchronization

Table 1-55 Checking whether existing data meets the constraints

Check Item	Whether existing data meets the constraints
Description	The constraint type of the source database table does not check whether existing data meets constraints. Some data in the source table may not meet the constraints, and the destination database may not support the constraint types. As a result, data transmission fails or data is lost because the data written to the destination database fails the constraint check.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: Some check item constraints are not met. Handling suggestion: Ensure that all data in the corresponding table meets the current constraints. If any data does not meet the constraints, consider not configuring constraints in the destination (contact DRS online support to skip the table structure consistency check), or process the source data to ensure that the data meets the constraints.

1.2.49 Checking the Additional Column of the Destination Database

MySQL to GaussDB(for MySQL) Synchronization

Table 1-56 Checking the additional column of the destination database

Check Item	Additional column check
Description	Check whether the additional column is added to the additional column list of the destination database. If no additional column is added, the incremental synchronization task fails.
Failure Cause and Handling Suggestion	Failure cause: Some columns are missing in the additional column processing table in the destination database. Handling suggestion: Add the missing additional columns to the destination database tables. Reference statement: ALTER TABLE ` database `.` table ` ADD COLUMN ` column_name ` column_definition;

1.2.50 Checking Whether Implicit Primary Key Check Is Enabled for the Primary and Standby Databases

MySQL Disaster Recovery

Table 1-57 Checking whether implicit primary key check is enabled for the primary and standby databases

Check Item	Whether implicit primary key check is enabled for the primary and standby databases
Description	Check whether **create_default_primary_key** is enabled for the source or destination database.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The **create_default_primary_key** parameter of the source or destination database is enabled. As a result, the primary/standby switchover may fail or data may be inconsistent. You are advised to disable **create_default_primary_key**. If this parameter is disabled, the task may still fail after a primary/standby switchover. The possible cause is that a table without a primary key is created when this parameter is enabled. Handling suggestion: This parameter was discarded, but in some DB instances of earlier versions, this parameter may be enabled. As a result, an error may occur during the primary/standby switchover, which may cause the task to fail or data inconsistency, you are advised to disable this parameter or ensure that the source database does not contain tables that do not have primary keys.

1.2.51 Checking Whether the Source Table Replication Attribute Is Correct

GaussDB(for openGauss)->GaussDB(for openGauss), GaussDB(for openGauss)->Kafka, PostgreSQL->PostgreSQL and PostgreSQL- >Kafka Synchronization

Table 1-58 Checking whether the source table replication attribute is correct

Check Item	Whether the source table replication attribute is correct
Description	The replica identity attribute of the source database table must be **full**.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The selected source database table contains the primary key column, but the replication attribute is not FULL. Handling suggestion: Change the replication attribute of the preceding tables to FULL. Run the following statement in the source database: `alter table table_name replica identity full;`
Failure Cause and Handling Suggestion	Failure cause: The source database contains tables whose replication attribute is not Full. As a result, incremental synchronization may fail. Handling suggestion: Change the replication attribute of the table to FULL. Run the following statement in the source database: `alter table table_name replica identity full;`

1.2.52 Checking Whether the Source Database Is on Standby

PostgreSQL to PostgreSQL and PostgreSQL to GaussDB(for openGauss) Synchronization

Table 1-59 Checking whether the source database is on standby

Check Item	Whether the source database is on standby
Description	If the source database is a standby database, set **hot_standby_feedback** in the source database to **on**.
Failure Cause and Handling Suggestion	Failure cause: The source database is a standby database and the value of **hot_standby_feedback** in the source database is **off**. As a result, full synchronization may fail. Handling suggestion: Before full synchronization, set the **hot_standby_feedback** parameter of the source database to **on**. After the full synchronization is complete, change the parameter to the original value.

1.2.53 Checking Whether the Specified Replication Slot Exists in the Source Database

PostgreSQL to Kafka and GaussDB(for openGauss) to Kafka Synchronization

Table 1-60 Checking whether the specified replication slot exists in the source database

Check Item	Whether the specified replication slot exists in the source database
Description	The replication slot with the specified name is automatically created after the task is started and cannot be the same as an existing replication slot in the source database.
Failure Cause and Handling Suggestion	Failure cause: The specified replication slot already exists in source database. Handling suggestion: ● Delete the replication slot from the source database. ● Alternatively, change the replication slot name to a name that does not exist in the source replication slots.

1.2.54 Checking Whether the MongoDB Instance Type Matches the Migration Mode

MongoDB Migration

Table 1-61 Checking whether the MongoDB instance type matches the migration mode

Check Item	Whether the MongoDB instance type matches the migration mode
Description	Check whether the MongoDB instance type matches the migration mode. If not, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: When a DRS task is created, the source DB instance type is set to **Cluster**, but the source database is not a cluster. Handling suggestion: If the source DB instance type is set to **Cluster**, ensure that the source database is a cluster database.

1.2.55 Checking the Physical Standby Database

Oracle Synchronization

Table 1-62 Physical standby database check

Check Item	Physical standby database check
Description	If the source Oracle database is in the incremental migration phase, the instance type of the source must be the same as that of the destination. Otherwise, the migration fails.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: 1. The physical standby database does not generate logs. It replicates them from the primary database. Check whether supplemental logging of the primary database meets the incremental synchronization requirements. Handling suggestion: Check **Oracle supplemental logging**. 2. The physical standby database does not generate logs, which results in delay for the synchronization task. You can shorten the interval for archiving logs from the primary database to the physical standby database. However, extremely low values can result in a large number of logs, so your advised to synchronize data from the logical standby database. Run the following statement on the primary database to specify the log archive interval: `alter system set archive_lag_target=`*seconds*`;`

1.3 Source DB Instance Statuses

1.3.1 Checking Whether the Source and Destination Databases Are of the Same Type

MongoDB Migration

Table 1-63 Checking whether the source and destination databases are of the same type

Check Item	Whether the source and destination databases are of the same type
Description	Check whether the source and destination databases are of the same type. If they are of different types, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The destination database is a cluster but the source database is a replica set. Handling suggestion: Change the type of the source or destination database.
	Failure cause: The destination database is a replica set but the source database is a cluster. Handling suggestion: Change the type of the source or destination database.

1.3.2 Checking Whether the ChangeStream API of the source DB instance is available

MongoDB-to-DDS Migration

Table 1-64 Checking whether the ChangeStream API of the source DB instance is available

Check Item	Whether the ChangeStream API of the source DB instance is available
Description	Check whether the ChangeStream API of the source database is available.
Failure Cause and Handling Suggestion	Failure cause: The source database cannot use the ChangeStream API. Handling suggestion: 1. Check whether the source database version is MongoDB 4.0 or later. 2. Check whether the WiredTiger storage engine is enabled for the source database. If not, you are advised to create a DRS task and select the oplog mode. Run the following command (on a shard node): `db.serverStatus().storageEngine.name;`
	Failure cause: The source database cannot use the ChangeStream API. Handling suggestion: Check whether the source database version is DDS 4.0 or later. If not, upgrade the source database to DDS 4.0 or later.

1.4 Destination DB Instance Statuses

1.4.1 Checking Whether the Destination Database Is Involved in Another Migration Task

MySQL

Table 1-65 Checking whether the destination database is involved in another migration task

Check Item	Whether the destination database is involved in another migration task
Description	Check whether the destination database is being used in another migration task. If more than one migration task uses the same destination database, the migration may fail.

Failure Cause and Handling Suggestion	Failure cause: The destination RDS DB instance is being used in another migration task. Handling suggestion: Wait for the migration task to complete. You can also stop or delete an unused migration task.

1.4.2 Checking Whether the Destination Database Has a Read Replica

MySQL

Table 1-66 Checking whether the destination database has a read replica

Check Item	Whether the destination database has a read replica
Description	Check whether the destination database has read replicas. If the destination database has read replicas, the incremental migration may fail.
Failure Cause and Handling Suggestion	Failure cause: In an incremental migration, the destination database cannot have read replicas. Handling suggestion: Delete the read replicas from the destination database. After the migration is complete, create read replicas.

1.4.3 Checking Whether the Destination Database Is Read-Only

MySQL Migration, Synchronization, and Backward DR

Table 1-67 Checking whether the destination database is read-only

Check Item	Whether the destination database is read-only
Description	The destination database is read-only, and data cannot be written to the destination database.

Failure Cause and Handling Suggestion	Failure cause: The destination database is read-only.
	Handling suggestion: Run the following commands to change the destination database to read/write and then, restart the database.
	Sample commands:
	set global read_only=0;
	set global super_read_only=0;

1.4.4 Checking Whether the Extensions Are Supported

PostgreSQL Synchronization

Table 1-68 Checking whether the extensions are supported

Check Item	Whether the extensions are supported
Description	Check whether the source database has plug-ins that are not installed on the destination database.
Failure Cause and Handling Suggestion	Failure cause: Extensions installed in the source database are not supported in the destination database. Handling suggestion: ● If the source database services do not depend on those extensions, run the following statement to delete the extensions. Replace *plugin_name* with the name of the extention to be deleted. `drop extension plugin_name;` ● Alternatively, use a destination database that supports these extensions.
	Failure cause: The source database has extensions that contain tables as members. Handling suggestion: Check whether the source database extensions contain metadata generated after the extensions are created. If yes, use the dedicated syntax of the extension to rebuild the metadata after the migration is complete.

	Failure cause: The destination database user does not have the permission to create extensions.
	Handling suggestion: Grant the permission to the user in the destination database as user root. Run the following SQL statements (replace *username* with the destination database username):
	`alter user username inherit;`
	`grant root to username;`

	Failure cause: The extension version supported by the destination database is earlier than that installed in the source database. Handling suggestion: Use the destination database that supports extensions of a later version (not earlier than the source database extension version) and create a synchronization task again.

1.4.5 Checking Whether Destination Contains the Configured Database

MySQL > PostgreSQL

Table 1-69 Checking whether destination contains the configured database

Check Item	Whether the destination contains the configured database.
Description	Databases and schemas cannot be migrated. You need to manually create databases and schemas on the destination database. Otherwise, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: Databases cannot be migrated from MySQL to PostgreSQL. Handling suggestion: In the destination database, manually create databases and schemas with the same names as those of the source database.
	Failure cause: The objects to be synchronized already exist in the destination database. Handling suggestions: Delete the tables to be synchronized from the destination database or select the tables that do not exist in the destination database for synchronization.

MySQL to GaussDB(for openGauss) Synchronization

Table 1-70 Checking whether destination contains the configured database

Check Item	Whether the destination contains the configured database.
Description	In the MySQL to GaussDB(for openGauss) synchronization scenario, the mapped database must exist in the destination database. Otherwise, the synchronization fails.

Check Item	Whether the destination contains the configured database.
Failure Cause and Handling Suggestion	Failure cause: The destination database does not contain the configured database. Handling suggestion: Before the synchronization, manually create a mapped database in the destination database.

GaussDB(for openGauss) to GaussDB(for openGauss) Synchronization

Table 1-71 Checking whether destination contains the configured database

Check Item	Whether the destination contains the configured database.
Description	In the GaussDB(for openGauss) to GaussDB(for openGauss) synchronization scenario, the mapped database must exist in the destination database. Otherwise, the synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: The configured database does not exist in the destination. Handling suggestion: Before the synchronization, manually create a configured database in the destination.

1.4.6 Checking Whether the Destination DB Instance Is Available

Table 1-72 Checking whether the destination DB instance is available

Check Item	Whether the destination DB instance is available
Description	Check whether the primary instance and read replicas are available in the destination database. If not, the migration fails.
Failure Cause and Handling	Failure cause: The destination DB instance is not available. Handling suggestion: Repair the destination DB instance.
	Failure cause: Read replicas in the destination database are abnormal. Handling suggestion: Repair the abnormal read replicas in the destination database.

Suggestion	Failure cause: The RDS service is abnormal. Try again later. Handling suggestion: Try again later.

1.4.7 Checking Whether the Destination Database Is Empty

MySQL

Table 1-73 Checking whether the destination database is empty

Check Item	Whether the destination database is empty
Description	Check whether the destination database is empty. If the destination database is not empty, disaster recovery will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database is not empty. Handling suggestion: Delete all the user-created databases from the destination database.

1.5 Database User Permissions

1.5.1 Checking Whether the Source Database User Has Sufficient Permissions

MySQL Migration

Table 1-74 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	The source database user must have the required permissions. ● In a full migration, the source database user must have the SELECT, SHOW VIEW, and EVENT permissions. ● In an incremental migration, the source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. If the permissions are insufficient, the migration will fail.
Failure Cause and Handl	Failure cause: In a full migration, the source database user must have the SELECT, SHOW VIEW, and EVENT permissions. Handling suggestions: Grant the source database user the corresponding permissions.

ing Suggestion	Failure cause: In an incremental migration, the source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, In the DR scenario, the following permissions are required: CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, and INDEX. Handling suggestions: Grant the source database user the corresponding permissions.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements. **NOTE** For details about the MySQL permissions and authorization operations required by DRS, see **Which MySQL Permissions Are Required for DRS?**
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

PostgreSQL Synchronization

Table 1-75 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Different types of synchronization tasks require different permissions. ● **Full synchornization:** The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, and the SELECT permission for system table **catalog pg_catalog.pg_authid** (used for synchronizing user passwords) ● **Full+incremental synchronization**: The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, the SELECT permission for system table **pg_catalog.pg_authid** (used for synchronizing user passwords), the UPDATE, DELETE, and TRUNCATE permissions for tables that do not have primary keys, and the permission to create a replication connection. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: In a full migration, the source database user must have the SELECT, REFERENCES, TRIGGER, EXECUTE, and USAGE permissions. Handling suggestion: Change or authorize the migration account.
	Failure cause: The replication permission is not configured in **pg_hba.conf** for the replication instance and database user. Handling suggestion: Grant the replication permission to the user. Add the following to **pg_hba.conf**, and restart the database for the modification to take effect: **host replication** *XXX(dbuser)* **0.0.0.0/0** *password* After the migration is complete, delete this record and restart the database again.
	Failure cause: The **max_wal_senders** value in the source database is too small. Handling suggestion: In the **postgresql.conf** file, change the **max_wal_senders** value to a larger one, such as increasing it by 5 or 10.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains objects that can only be created by a superuser. The destination user is not a superuser, so the objects will be ignored. Handling suggestion: Use a superuser of the destination database or confirm that these objects can be ignored.

MongoDB Migration

Table 1-76 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Check whether the source database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.

Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The source database user does not have the required permissions. Handling suggestion: Check whether the source database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Oracle Migration

Table 1-77 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Check whether the source database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database user does not have sufficient permissions. Handling suggestion: Grant the CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, and SELECT ANY DICTIONARY permissions to the user and perform the check again. **NOTE** If the destination database is PostgreSQL, the SELECT ANY SEQUENCE permission is also required.

MySQL -> PostgreSQL Synchronization

Table 1-78 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Check whether the source database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.

Failure Cause and Handling Suggestion	Failure cause: The source database user does not have sufficient permissions. During the incremental synchronization, the SELECT permission on INFORMATION_SCHEMA is required. Handling suggestions: Grant the source database user the corresponding permissions. Run the following command: `grant select on [dbname];`

GaussDB(for openGauss) Synchronization

Table 1-79 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Check whether the source database user permissions meet the synchronization requirements. If the permissions are insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the synchronization requirements.

1.5.2 Checking Whether the Destination Database User Has Sufficient Permissions

MySQL Migration

Table 1-80 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.

Failure Cause and Handling Suggestion	Failure cause: The destination database user must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and WITH GRANT OPTION. If the destination database version is in the range from 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. Handling suggestion: Check whether the destination database user permissions meet the migration requirements.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

PostgreSQL Synchronization

Table 1-81 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Different permissions are granted based on the scope of objects to be synchronized. ● **Database-level synchronization objects:** The CREATEDB permission and the **root** user or a member of root are required (only in special scenarios). For details, see the following description. ● **Table-level synchronization object:** To synchronize databases: The CREATEDB permission is required. To synchronize schemas, the CONNECT and CREATE permissions for the database that contains the schema are required. To synchronize objects in a schema: The CONNECT permission for the database that contains the schema, the USAGE permission for the schema that contains the object, and CREATE permission for the schema that contains the object are required. ● **Synchronization user:** The CREATEROLE permission is required. ● **Permission to synchronize objects:** The default privilege is the default value and cannot be modified. Otherwise, the object permissions of the destination database may be different from those of the source database.
Failure Cause and Handl	Failure cause: The destination database user must have the CREATEDB permission. Handling suggestion: Grant the destination database user the CREATEDB permission. `alter role username with createdb;`

ing Suggestion	Failure cause: The user does not have the USAGE permission for schemas. Handling suggestion: Grant the destination database user the CREATEDB permission. `grant usage on schema schemaname to username;`
	Failure cause: The user does not have the permission to create tales. Handling suggestion: Grant the destination database user the permission to create tables. `grant create on schema schemaname to username;`
	Failure cause: The user does not have the permission to create schemas. Handling suggestion: Grant the destination database user the permission to create schemas. `grant create on database to username;`

MongoDB Migration

Table 1-82 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The destination database user does not have the required permissions. Handling suggestion: Check whether the destination database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Oracle -> MySQL/Oracle -> GaussDB(for MySQL) Synchronization

Table 1-83 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database user must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, TRIGGER, and ALTER. Handling suggestion: Check whether the destination database user permissions meet the migration requirements.

Oracle -> PostgreSQL Synchronization

Table 1-84 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database user does not have sufficient permissions. Handling suggestions: Run the following command to grant the permissions of creating databases and tables on the destination database. Then, perform the check again. `ALTER ROLE targetUser WITH CREATEDB;` In the preceding command, **targetUser** indicates the current user of the destination database.

MySQL -> PostgreSQL Synchronization

Table 1-85 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the requirements for MySQL to PostgreSQL synchronization. If the permissions are insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: Some tables in the destination database do not have the INSERT, SELECT, UPDATE, and DELETE permissions. Handling suggestion: Add the required permissions for the tables and try again. Example command: `GRANT INSERT,SELECT,UPDATE, DELETE ON db_name.table_name TO 'user'@'host';`

PostgreSQL > PostgreSQL Synchronization

Table 1-86 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the requirements for PostgreSQL to PostgreSQL synchronization. If the permissions are insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database user must have the CREATEDB permission. Handling suggestion: Grant the destination database user the CREATEDB permission. `ALTER ROLE role_name WITH CREATEDB;`

MySQL Disaster Recovery

Table 1-87 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions

Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the disaster recovery will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database user must have the GRANT permission. Handling suggestion: Grant user the required permission. `GRANT SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'u1' WITH GRANT OPTION;`

DDM -> Oracle Synchronization

Table 1-88 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the requirements for DDM to Oracle synchronization. If the permissions are insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database user does not have sufficient permissions. Handling suggestion: Use an account with the DBA permission, or grant required permissions to the user.

GaussDB(for openGauss) as the Destination

Table 1-89 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Check whether the source database user permissions meet the migration requirements.

Failure Cause and Handling Suggestion	Failure cause: The source database user does not have sufficient permissions. The user needs the select permission for the schema. Handling suggestion: Grant the missing permissions to the object owner or system administrator.

1.5.3 Checking Whether the Destination Database Account Has Required Permissions to Migrate Definer

MySQL Migration

Table 1-90 Checking Whether the Destination Database Account Has Required Permissions to Migrate Definer

Check Item	Checking whether the destination database account has required permissions to migrate Definer
Description	To migrate Definers to the cloud, the source database user must have the all privileges permission.
Failure Cause and Handling Suggestion	Failure cause: The permission of the specified destination database account is insufficient. Handling suggestions: Migrate all Definers to the specified destination database account. Alternatively, do not migrate Definers to the specified destination database account and run the following command to grant the destination database account the all privileges permission. Example command: `grant all privileges on *.* to 'user' @' host'`
	Failure cause: The specified source database user does not have required permissions. Handling suggestion: 1. When configuring the destination database, select **Migrate Definer to User** to ensure that Definers of all the objects are under the specified user. 2. Retain the Definer settings and run the following command to grant the all privileges permission to the source database user. Example command: `grant all privileges on *.* to 'user' @' host'`

1.6 Database Versions

1.6.1 Checking Whether the Source Database Version Is Supported

MySQL Migration

Table 1-91 Checking whether the source database version is supported

Check Item	Whether the source database version is supported
Description	Check whether the source database version is MySQL 5.5.x, MySQL 5.6.x, MySQL 8.0.x, and MySQL 5.7.x.
Failure Cause and Handling Suggestion	Failure cause: The source database version is not supported. Handling suggestion: Export data and then import it following the instructions provided in the **Migrating MySQL Data Using mysqldump** section in the *Relational Database Service User Guide*.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.

Migration from MongoDB to DDS

Table 1-92 Checking whether the source database version is supported

Check Item	Whether the source database version is supported
Description	Check whether the source database version is MongoDB 3.2.x, 3.4.x, or 4.0.x.
Failure Cause and Handling	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is MongoDB 3.2x, 3.4.x, and 4.0.x.

Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The incremental data is obtained in changeStream mode but the source database version is earlier than 4.0. Handling suggestion: In changeStream mode, ensure that the source database version is 4.0 or later.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

PostgreSQL to GaussDB(DWS) Synchronization

Table 1-93 Checking Whether the Source Database Version Is Supported

Check Item	Whether the source database version is supported
Description	Check whether the source database version is PostgreSQL 9.4, PostgreSQL 9.5, PostgreSQL 9.6, PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0..
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only PostgreSQL 9.4, PostgreSQL 9.5, PostgreSQL 9.6, PostgreSQL 10.0, PostgreSQL 11.0, and PostgreSQL 12.0, and PostgreSQL 13.0. are supported.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Oracle Synchronization

Table 1-94 Checking whether source database version is supported

Check Item	Whether the source database version is supported

Description	Check whether the source database version is Oracle 10g, 11g, 12c, 18c, or 19c.
Failure Cause and Handling Suggestion	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only Oracle 10g, 11g, 12c, 18c, and 19c are supported.

MySQL -> PostgreSQL Synchronization

Table 1-95 Checking whether the source database version is supported

Check Item	Whether the source database version is supported
Description	Check whether the source database version meets the requirements for synchronization from MySQL to PostgreSQL.
Failure Cause and Handling Suggestion	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only MySQL 5.5, 5.6, 5.7, and 8.0 are supported.

1.6.2 Checking Whether the Destination Database Version Is Supported

MySQL Migration

Table 1-96 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported
Description	Check whether the destination database version is MySQL 5.6.x, MySQL 8.0.x, or MySQL 5.7.x.

Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported. Handling suggestion: Export data and then import it following the instructions provided in the **Migrating MySQL Data Using mysqldump** section in the *Relational Database Service User Guide*.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.

MongoDB to DDS Migration

Table 1-97 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported
Description	Check whether the destination database version is MongoDB 3.4.x, 4.0.x, or 4.2.x.
Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is MongoDB 3.4.x, 4.0.x, or 4.2.x.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Oracle -> MySQL Migration

Table 1-98 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported

Description	Check whether the destination database version meets the migration requirements.
Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only MySQL 5.6.x and MySQL 5.7.x are supported.

Oracle -> RDS PostgreSQL Migration

Table 1-99 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported
Description	Check whether the destination database version meets the migration requirements.
Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only PostgreSQL Enhanced Edition is supported.

MySQL -> PostgreSQL Synchronization

Table 1-100 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported
Description	Check whether the destination database version meets the requirements for synchronization from MySQL to PostgreSQL.
Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only PostgreSQL 10.x is supported.

1.6.3 Checking Whether the Migration Is from an Earlier Database Version to the Same or a Later Version

MySQL Migration

Table 1-101 Checking whether the migration is from an earlier database version to the same or a later version

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only MySQL 5.5.x, MySQL 5.6.x, MySQL 8.0.x, and MySQL 5.7.x are supported.
	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only MySQL 5.6.x, MySQL 8.0.x, and MySQL 5.7.x are supported.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The destination database version must be the same as or later than the source database version. Handling suggestion: Ensure the source database version is earlier than or the same as the destination database version.

PostgreSQL Synchronization

Table 1-102 Checking whether the migration is from an earlier database version to the same or a later version

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only PostgreSQL 9.4, 9.5, 9.6, 10.0, 11.0, 12.0, 13.0, and PostgreSQL Enhanced Edition are supported.
	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only PostgreSQL 9.5, 9.6, 10.0, 11.0, 12.0, 13.0, and PostgreSQL Enhanced Edition are supported.
	Failure cause: The major versions of the source and destination databases must be the same and the minor version of the source database must be less than or equal to that of the destination database. Handling suggestion: Ensure the source database version is earlier than or the same as the destination database version.
	Failure cause: If the source database is an RDS PostgreSQL Enhanced Edition instance, the destination database can only be an RDS PostgreSQL Enhanced Edition instance. Handling suggestion: Select RDS PostgreSQL Enhanced Edition instance as the destination database.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

	Failure cause: The destination database version and source database version do not meet the requirements of the selected migration mode. Handling suggestion: Check whether the versions of the destination database and source database meet the migration mode requirements.

MySQL Disaster Recovery

Table 1-103 Checking whether the migration is from an earlier database version to the same or a later version

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version.
Failure Cause and Handling Suggestion	Failure cause: The versions of the source and destination databases are inconsistent or do not meet the version requirements. Handling suggestion: The versions of the source and destination databases must be the same and be 5.6, , 5.7, or 8.0. Select the databases whose versions meet the version requirements.

MongoDB Migration

Table 1-104 Checking Whether the Source Database Version Is Supported

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version. The destination database version can be 4.0.x or later. The migration must be performed from an older database version to the same or a newer version Check whether the features of the newer version are used in the source database. If the new features are not supported in the destination database, the synchronization may fail.
Failure Cause and Handling Suggestion	Item to be confirmed: The versions of the source and destination databases are different. Potential problem: The source database version is 4.4.5 and the destination database version is 4.0.3, which does not comply with the migration version requirements. If the source database supports features of the newer version but the destination database does not, the synchronization may fail.

	Failure cause: The destination database version does not meet the requirements. Handling suggestion: Select version 4.0.x for the destination database because the source database version is later than 4.0.x and the destination database version is 3.4.x.

1.7 Networks

1.7.1 Checking Whether the Source Database Is Connected

MySQL Migration

Table 1-105 Checking whether the source database is connected

Check Item	Whether the source database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the source database.
Failure Cause and Handling Suggestion	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Run the following command to create a user that allows remote connections. After the migration, delete this user. `CREATE USER 'Account' @ '%' IDENTIFIED BY 'Password'`
	Failure cause: The SSL CA root certificate is invalid. Handling suggestion: Upload a valid SSL CA certificate.
	Failure cause: No SSL CA root certificate exists. Handling suggestion: Contact Huawei technical support.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.

PostgreSQL Synchronization

Table 1-106 Checking whether the source database is connected

Check Item	Whether the source database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the source database.
Failure Cause and Handling Suggestion	Failure cause: The IP address is inaccessible. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Configure the remote connection permission for the user in the **pg_hba.conf** file. Open **pg_hba.conf**, add the following, and restart the database for the modification to take effect: `host all xxx(dbuser) 0.0.0.0/0 password` After the migration is complete, delete this record and restart the database again.
	Failure cause: Failed to connect to the database. Handling suggestion: The **listen_addresses** parameter value or port number in the **postgres.conf** file is incorrect. In the **postgres.conf** file, set the **listen_addresses** value to **'*'** or set the port number to the correct value. Then, restart the database for the modification to take effect.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The user does not have the login permission. Handling suggestion: Run the following command to grant the login permission to the user: **alter role** *xxx(dbuser)* **login**
	Failure cause: The postgres database does not exist in the source database. Handling suggestion: Create a postgres database.

MongoDB Migration

Table 1-107 Checking whether the source database is connected

Check Item	Whether the source database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the source database.
Failure Cause and Handling Suggestion	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: The username, password, or authentication database of the source database is incorrect. Handling suggestion: Check that the input username, password, and authentication database for the connection test are correct.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: SSL connection parameters are not configured. Handling suggestion: Contact Huawei technical support.
	Failure cause: The source database cannot connect to the port of the replication instance. Handling suggestion: Modify the firewall, security group, or ACL configurations of the source and destination databases to enable the source database to connect to the port of the replication instance.

1.7.2 Checking Whether the Destination Database Is Connected

MySQL Migration and Synchronization

Table 1-108 Checking whether the destination database is connected

Check Item	Whether the destination database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the destination database.

Failure Cause and Handling Suggestion	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Run the following command to create a user that allows remote connections. After the migration, delete this user. **CREATE USER** '*Account*' **@ '%' IDENTIFIED BY** '*Password*'
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.

PostgreSQL Synchronization

Table 1-109 Checking whether the destination database is connected

Check Item	Whether the destination database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the destination database.
Failure Cause and Handling Suggestion	Failure cause: The IP address is inaccessible. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Grant the remote connection permission for the user in the **pg_hba.conf** file because the replication instance and user are not configured in the **pg_hba.conf** configuration file. Add the following to the **pg_hba.conf** configuration file. After the migration is complete, delete this record and restart the database for the modification to take effect. host all xxx(dbuser) 0.0.0.0/0 password

	Failure cause: Failed to connect to the database. The failure may be caused by the incorrect **listen_addresses** parameter value or port number in **postgres.conf**. Handling suggestion: In the **postgres.conf** file, set **listen_addresses** to "*" or set the port number to the correct value. Then, restart the database for the modification to take effect.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The user does not have the login permission. Handling suggestion: Run the following command to grant the login permission to the user: **alter role** *xxx(dbuser)* **login**
	Failure cause: The postgres database does not exist in the source database. Handling suggestion: Create a postgres database.

MongoDB Migration

Table 1-110 Checking whether the destination database is connected

Check Item	Whether the destination database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the destination database.
Failure Cause and Handling Suggestion	Failure cause: The connection fails. Handling suggestion: See network preparations in the Data Replication Service Best Practices.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: SSL connection parameters are not configured. Handling suggestion: Contact Huawei technical support.

1.7.3 Checking Whether the Destination Database Can Connect to the Source Database

MySQL Migration and Synchronization

Table 1-111 Checking whether the destination database can connect to the source database

Check Item	Whether the destination database can connect to the source database
Description	Check whether the destination database can connect to the source database.
Failure Cause and Handling Suggestion	Failure cause: The destination database fails to connect to the source database. Handling suggestion: See network preparations in the Data Replication Service Best Practices.

1.8 Database Objects

1.8.1 Checking Whether the Source Database Contains a MyISAM Table

MySQL

Table 1-112 Checking whether the source database contains a MyISAM table

Check Item	Whether the source database contains a MyISAM table
Description	If the source database contains a MyISAM table, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains MyISAM tables that are not supported by the destination database, which may cause the migration to fail. Handling suggestion: Convert the tables in the source database to InnoDB tables and try again. Alternatively, contact technical support.

1.8.2 Checking Whether the Source Database Contains Unsupported Table Field Types

Oracle Migration and Synchronization

Table 1-113 Checking whether the source database contains unsupported table field types

Check Item	Whether the source database contains unsupported table field types
Description	Unsupported table field types are as follows: BFILE, XMLType, SDO_GEOMETRY, TIMESTAMP (x), INTERVAL DAY TO SECOND (x), and UROWID. The source database contains unsupported table field types, resulting in migration failures. **NOTE** ● *x* indicates the precision. TIMESTAMP and INTERVAL DAY TO SECOND do not support the column type with precision between 7 to 9. ● Due to internal restrictions, DRS cannot filter out special fields during data processing.
Failure Cause and Handling Suggestion	Failure cause: The source database contains unsupported table field types. Handling suggestion: Delete the columns containing the unsupported field types. Alternatively, do not migrate the tables containing the unsupported table field types.

MySQL -> PostgreSQL Synchronization

Table 1-114 Checking whether the source database contains unsupported table field types

Check Item	Whether the source database contains unsupported table field types
Description	The following table fields types are not supported: geometry, point, lineString, polygon, geometrycollection, multipoint, multilinestring, and multipolygon. The source database contains unsupported table field types, resulting in migration failures.

Failure Cause and Handling Suggestion	Failure cause: The source database contains unsupported table field types. Handling suggestion: Delete the columns containing the unsupported field types. Alternatively, do not migrate the tables containing the unsupported table field types.

1.8.3 Checking Whether the Source Database Contains the Functions or Stored Procedures that the Source Database User Is Not Authorized to Migrate

MySQL

Table 1-115 Checking whether the source database contains the functions or stored procedures that the source database user is not authorized to migrate

Check Item	Whether the source database contains the functions or stored procedures that the source database user is not authorized to migrate.
Description	The source database contains the functions or stored procedures that the source database user is not authorized to migrate.
Failure Cause and Handling Suggestion	Failure cause: The source database user does not have the permission to migrate functions and stored procedures. Handling suggestion: Ensure that the source database user has the highest-level right.

1.8.4 Checking Whether Objects with the Same Names Exist in the Source Database

MySQL -> PostgreSQL Synchronization

Table 1-116 Checking whether objects with the same names exist in the source database

Check Item	Whether objects with the same names exist in the source database
Description	Failure cause: The source databases selected are not mapped to the same database, or tables with same names exist in the selected databases.

Failure Cause and Handling Suggestion	Failure cause: PostgreSQL does not support synchronization of multiple databases, or objects have the same names in the source databases. Handling suggestion: Select one database for migration, or map the selected databases to the same database and ensure that the object in the databases have unique names.

1.8.5 Checking Whether the Source Database Contains Unlogged Tables

PostgreSQL as the Source

Table 1-117 Checking whether the source database contains unlogged tables

Check Item	Whether the source database contains unlogged tables
Description	Check whether the source database contains unlogged tables. If yes, the synchronization fails.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains unlogged tables and modifications to these tables are not recorded in logs. As a result, incremental data of unlogged tables cannot be synchronized. Handling suggestions: Check whether incremental data of the unlogged tables needs to be synchronized. If yes, run the following command to change the unlogged tables to logged: ALTER TABLE TABLE_NAME SET LOGGED."

1.8.6 Checking Whether the Names of Views to Be Migrated Are the Same

Oracle Migration and Synchronization

Table 1-118 Checking whether the source database meets integrity constraints

Check Item	Whether the source database meets constraint integrity
Description	Check the constraint integrity of the source database. If the check result does not meet the migration requirements, the migration fails.

Item to Be Confirmed and Handling Suggestion	Failure cause: Tables to be migrated are dependent but the referenced tables are not to be migrated. Handling suggestion: Select the referenced tables in the migration object list.
	Failure cause: Views that cannot be migrated are selected in the migration object list. Handling suggestions: Check all objects which the views reference and select them for migration.
	Failure cause: The names of the views to be migrated are the same except for letter cases. Handling suggestions: Change the view names or do not migrate these views.
	Failure cause: Source database constraint names are the same except for letter cases. Handling suggestions: Change the constraint names or do not migrate the constraints whose names are the same except for letter cases.

1.8.7 Checking Whether the _Id Fields in the Collection of the Source Database Have Indexes

MongoDB Migration

Table 1-119 Checking whether the **_id** fields in the collections of the source database have indexes

Check Item	Whether the **_id** fields in the collections of the source database have indexes
Description	Check whether the **_id** fields in the collections of the source database have indexes. If yes, the migration fails.
Item to Be Confirmed and Handling Suggestion	Failure cause: The **_id** fields in the collections of the source database must have indexes. Handling suggestion: Run the **db.*Collection name*.ensureIndex({_id: 1})** command to add an index. If the index fails to be added and the system displays a message indicating that the **_id** value already exists, the collection cannot be migrated.

1.8.8 Checking Whether the Index Length of the Source Database Exceeds the Limit

Oracle -> MySQL/Oracle -> GaussDB(for MySQL) Migration and Synchronization

Table 1-120 Checking whether the index length of the source database exceeds the limit

Check Item	Whether the index length of the source database exceeds the limit
Description	The migration fails because the index length of the source database exceeds the column length limit of the destination database.
Failure Cause and Handling Suggestion	Failure cause: There are indexes in the source database exceed the column length limit of the destination database. Handling suggestion: 1. Delete the tables from the migration object. 2. Modify the index length.

MySQL Migration, Synchronization, and Disaster Recovery

Table 1-121 Checking whether the index length of the source database exceeds the column length limit

Check Item	Whether the index length of the source database exceeds the limit
Description	The migration fails because the index length of the source database exceeds the column length limit of the destination database.
Failure Cause and Handling Suggestion	Failure cause: There are indexes in the source database exceed the column length limit of the destination database. Handling suggestion: Set **innodb_large_prefix** of the destination database to **ON**, or return to the previous step and select the table to be migrated again.

1.8.9 Checking Whether the Source Database Tables Use Storage Engines Not Supported by the Destination Database

MySQL Migration and Synchronization

Table 1-122 Checking whether the source database tables use storage engines not supported by the destination database

Check Item	Whether the source database tables use storage engines not supported by the destination database
Description	Check whether the source database tables use storage engines not supported by the destination database. If yes, the migration fails.
Item to Be Confirmed and Handling Suggestion	Failure cause: The source database tables use the storage engines that are not supported by the destination database. Handling suggestion: Go back to the previous page and deselect the tables that use the storage engines not supported by the destination database.

1.8.10 Checking Whether the Database Names Mapped to the Destination DB Instance Contain Unsupported Characters

MySQL

Table 1-123 Checking whether the database names mapped to the destination database contain unsupported characters.

Check Item	Whether the database names mapped to the destination DB instance contain unsupported characters.
Description	The following characters are not supported in the database names mapped to the destination DB instance: .<>\'
Item to Be Confirmed and Handling Suggestion	Failure cause: The database names mapped to the destination DB instance contain unsupported characters. Handling suggestion: Go back to the object selection page and change the source database names to be mapped to the destination DB instance.

1.8.11 Checking Whether the Source Database Tables Contain Primary Keys

MySQL Migration and Disaster Recovery

Table 1-124 Checking whether the source database tables contain primary keys

Check Item	Whether the source database tables contain primary keys
Description	If tables to be migrated in the source database do not contain primary keys, the migration may fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The tables to be migrated in the source database do not contain primary keys. Handling suggestion: Create a primary key for the table. If the table does not have a primary key to uniquely identify every row and the network connection is unstable, the data in the destination database may be inconsistent with that in the source database.

MySQL Synchronization

Table 1-125 Checking whether the source database tables contain primary keys

Check Item	Whether the source database tables contain primary keys
Description	If tables to be synchronized in the source database do not contain primary keys, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The tables to be synchronized in the source database do not contain primary keys. Handling suggestion: Create primary keys for the tables to improve performance.
	Failure cause: In a many-to-one task, tables with no primary key that have the same name as those in the destination database are not allowed in the source database. Handling suggestion: Modify tables without primary keys, delete tables with no primary key from the destination database, or do not migrate tables without primary keys.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The tables to be synchronized in the source database do not contain primary keys. Handling suggestion: Create a primary key for the table. If the table does not have a primary key to uniquely identify every row and the network connection is unstable, the data in the destination database may be inconsistent with that in the source database.

1.8.12 Checking Whether the Source Database Contains Triggers or Events

MySQL Migration

Table 1-126 Checking whether the source database contains triggers or events

Check Item	Whether the source database contains triggers or events
Description	To prevent unexpected operations on the destination database automatically triggered by triggers or events, this task starts the trigger or event migration only after you stop the task. If you close or disconnect the source database connection during the task running, triggers or events are not migrated.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains triggers or events. Handling suggestion: Stop the task first and then disconnect the network to ensure the completeness of the migration.

1.8.13 Checking Whether the Source Database Referenced Roles Pass the Check

MongoDB Migration

Table 1-127 Checking whether the source database referenced roles pass the check

Check Item	Whether the source database referenced roles pass the check
Description	If the roles referenced by accounts to be migrated are not migrated to the destination database, the migration may fail.
Failure Cause and Handling Suggestion	Failure cause: The databases referenced by source database roles do not exist in the destination database, and are not displayed in the object selection list. Handling suggestion: Select the databases referenced by roles for migration or do not migrate the roles.
	Failure cause: The roles referenced by source database roles do not exist in the destination database, and are not displayed in the role selection list. Handling suggestion: Select the referenced roles or do not migrate the roles that fail the check.

1.8.14 Checking Whether the Source Database Referenced Accounts Pass the Check

MongoDB Migration

Table 1-128 Checking whether the source database referenced accounts pass the check

Check Item	Whether the source database referenced accounts pass the check
Description	If the roles referenced by accounts to be migrated are not migrated to the destination database, the migration may fail.
Failure Cause and Handling Suggestion	Failure cause: The databases referenced by source database account do not exist in the destination database, and are not displayed in the object selection list. Handling suggestion: Select the roles referenced by accounts for migration or do not migrate the accounts.
	Failure cause: The databases referenced by source database account do not exist in the destination database, and are not displayed in the object selection list. Handling suggestion: Select the databases referenced by accounts for migration or do not migrate the accounts.

1.8.15 Checking Database Mapping Objects

MySQL -> MySQL

Table 1-129 Checking database mapping objects

Check Item	Whether the database mapping objects are supported
Description	Database mapping does not support views or stored procedures. If the source database contains views or stored procedures, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: Database mapping does not support views or stored procedures. Handling suggestions: Cancel the mapping of databases containing views or stored procedures, or delete the views or stored procedures from the databases.

1.8.16 Checking Whether Associated Objects Are Selected

PostgreSQL to RDS PostgreSQL Migration and Synchronization, and PostgreSQL to GaussDB(DWS) Synchronization

Table 1-130 Checking whether associated objects are selected

Check Item	Whether the associated objects are selected
Description	The associated objects must be selected for migration. Otherwise, the migration may fail.
Failure Cause and Handling Suggestion	Failure cause: Tables referenced by the foreign key in the table to be migrated are not selected for migration. Handling suggestion: Select the associated objects.
	Failure cause: The selected objects contain views associated with some tables or views that are not selected for migration. Handling suggestion: Select the associated objects.
	Failure cause: The tables associated with the child tables to be migrated are not selected for migration. Handling suggestion: Select the associated objects.

1.8.17 Checking Whether the Specified Objects Exist In the Destination Database

PostgreSQL to RDS PostgreSQL Migration and Synchronization, and PostgreSQL to GaussDB(DWS) Synchronization

Table 1-131 Checking whether the specified migration objects exist in the destination database.

Check Item	Whether the specified migration objects exist in the destination database.
Description	The objects with the same names as the source objects cannot exist in the destination database. Otherwise, the migration may fail.
Failure Cause and Handling Suggestion	Failure cause: The objects to be synchronized exist in the destination database. Handling suggestion: Delete the objects with the same names as the source objects and perform the verification again.

1.8.18 Checking Whether the Source Table Contains Column Types that Cannot Be Used as Distribution Keys

PostgreSQL to GaussDB(DWS) Synchronization

Table 1-132 Checking whether the source table contains column types that cannot be used as distribution keys

Check Item	Whether the source table contains column types that cannot be used as distribution keys
Description	The source table cannot contain column types used as distribution keys. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The source table contains column types that cannot be used as distribution keys. Handling suggestion: Select another table that can be migrated.

1.8.19 Checking Whether the Source Table Contains Unsupported Table Field Types

PostgreSQL to GaussDB(DWS) Synchronization

Table 1-133 Checking whether the source table contains unsupported table field types

Check Item	Whether the source table contains unsupported table field types
Description	The source tables cannot contain unsupported field types. Otherwise, the synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: The source database table contains unsupported table field types. Handling suggestion: Select another table that can be migrated.

1.9 Database Configuration Items

1.9.1 Checking Whether the Source Database Name Is Valid

MySQL Migration

Table 1-134 Checking whether the source database name is valid

Check Item	Whether the source database name is valid
Description	The source database name cannot contain invalid characters. It must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). If the source database name contains any invalid character, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The source database name cannot contain invalid characters. It must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). Handling suggestion: Change the source database names that contain unsupported characters or go back to the previous page and select the databases that do not contain unsupported characters.

MongoDB Migration

Table 1-135 Checking whether the source database name is valid

Check Item	Whether the source database name is valid
Description	If the source database name contains invalid characters, the migration fails.
	Failure cause: The source database names cannot contain the following special characters: .<>' Handling suggestion: Change the source database names that contain unsupported characters or go back to the previous page and select the databases that do not contain unsupported characters.

Oracle Migration

Table 1-136 Checking whether the source database name is valid

<table>
<tr><th>Check Item</th><th>Whether the source database name is valid</th></tr>
<tr><td rowspan="2"><b>Description</b></td><td>The source database names cannot contain non-ASCII characters or special characters .><\`|,?'!"
If the source database name contains invalid characters, the migration fails.</td></tr>
<tr><td>Failure cause: The source database names contain unsupported characters.
Handling suggestion: Change the source database names that contain unsupported characters or go back to the previous page and select the databases that do not contain unsupported characters.</td></tr>
</table>

1.9.2 Checking Whether the Source Database Table Name Is Valid

MySQL Migration

Table 1-137 Checking whether the source database table name is valid

Check Item	Whether the source database table name is valid
Description	If the source database table name contains invalid character, the synchronization task fails.
Failure Cause and Handling Suggestion	Failure cause: The source database table names contain unsupported characters, non-ASCII characters, or the following characters: ></\ Handling suggestion: To solve this problem, perform the following steps: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the table that contains unsupported characters. Method 2: Change the table name.

PostgreSQL Migration

Table 1-138 Checking whether the source database table name is valid

Check Item	Whether the source database table name is valid

Description	The source database table name cannot contain single quotation marks ('), double quotation marks ("), and periods (.). Ensure that the source database table name does not contain invalid characters. Otherwise, the synchronization task fails.
Failure Cause and Handling Suggestion	Failure cause: The source database names contain unsupported characters. Handling suggestion: Run the following command to change the source database names that contain unsupported characters: alter table old_name rename to new_name;

Oracle Migration

Table 1-139 Checking whether the source database table name is valid

Check Item	Whether the source database table name is valid
Description	The source database names cannot contain non-ASCII characters or special characters .><\`|,?'!" If the table name or view name of the source database contains any invalid character, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The table name or view name of the source database contains unsupported characters. Handling suggestion: ● Run the following command to change the table name that contains unsupported characters: alter table old_name rename to new_name; ● Run the following command to change the view name that contains unsupported characters: rename old_view_name to new_view_name;

1.9.3 Checking Whether the Source Database View Name Is Valid

MySQL

Table 1-140 Checking whether the source database contains view names with non-ASCII characters

Check Item	Whether the source database contains view names with non-ASCII characters

Description	If the source database contains non-ASCII characters, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Failure cause: The source database view names contain unsupported characters, non-ASCII characters, or the following characters: ></\ Handling suggestion: To solve this problem, perform the following steps: Method 1: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the view name that contains unsupported characters. Method 2: Change the view name.

1.9.4 Checking Whether the Source Database Collection Name Is Valid

MongoDB Migration

Table 1-141 Checking whether the source database collection name is valid

Check Item	Whether the source database collection name is valid
Description	The source database collection names cannot contain slashes (/) or backslashes (\).
	Failure cause: The source database collection names cannot contain the following characters: ,<> Handling suggestion: Modify the source database collection names that contain invalid characters or go back to the **Select Migration Type** page and select the collections that do not contain invalid characters.

1.9.5 Checking Whether the Shard Key Can Be Obtained from the Source Database

MongoDB Migration

Table 1-142 Checking whether the shard keys can be obtained from the source database

Check Item	Whether the shard keys can be obtained from the source database

Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Item to be confirmed: The source database is a replica set but the shard keys have not been configured in the destination database. Handling suggestion: If the destination database cannot obtain the shard keys of the source database, the data in the source database will be migrated to the primary shard node of the sharded cluster in the destination database. To fully utilize the read/write performance, storage capability, and high availability of the cluster, see **FAQs**.
	Item to be confirmed: The source database type is unknown, and the shard keys have not been configured in the destination database. Handling suggestion: If the destination database cannot obtain the shard keys of the source database, the data in the source database will be migrated to the primary shard node of the sharded cluster in the destination database. To fully utilize the read/write performance, storage capability, and high availability of the cluster, see **FAQs**.
	Item to be confirmed: The source database contains collections that do not have shard keys configured. Handling suggestion: If the destination database cannot obtain the shard keys of the source database, the data in the source database will be migrated to the primary shard node of the sharded cluster in the destination database. To fully utilize the read/write performance, storage capability, and high availability of the cluster, see **FAQs**.

1.9.6 Checking Whether the Source Database Schema Name Is Valid

PostgreSQL

Table 1-143 Checking whether the source database schema name is valid

Check Item	Whether the source database schema name is valid
Description	The source database schema name cannot contain single quotation marks ('), double quotation marks ("), and periods (.). Ensure that the source database schema name does not contain invalid characters. Otherwise, the synchronization task fails.
Failure Cause and Handling Suggestion	Failure cause: The source database schema names contain unsupported characters. Handling suggestion: Run the following command to change the source database schema names that contain unsupported characters: `alter schema old_name rename to new_name;`

1.9.7 Checking Whether the Maximum Number of Chunks in the Destination Database Is Sufficient

MongoDB Migration

Table 1-144 Checking whether the maximum number of chunks in the destination database is sufficient

Check Item	Whether the maximum number of chunks in the destination database is sufficient
Description	The maximum number of chunks in the destination database is insufficient to support sharding and splitting of the source database. If the maximum number of chunks is reached, chunks are not split and the write performance is negatively affected.
Failure Cause and Handling Suggestion	Item to be confirmed: The maximum number of chunks in the destination database is insufficient to support sharding and splitting of the source database. If the maximum number of chunks is reached, chunks are not split and the write performance is negatively affected. Handling suggestion: Select a destination DB instance of higher specifications.

1.9.8 Checking Whether Archive Logs Are Enabled on the Source Oracle Database

Oracle -> MySQL Migration and Synchronization

Table 1-145 Checking whether archive logs are enabled on the source Oracle database

Check Item	Whether archive logs are enabled on the source Oracle database
Description	During incremental migration from Oracle to MySQL, archive logs must be enabled on the source Oracle database.
Failure Cause and Handling Suggestion	Failure cause: Archive logs are not enabled on the source Oracle database. Handling suggestion: Run the **alter database archivelog** command to enable archive logs on the source Oracle database.

1.9.9 Checking Whether Supplemental Logging of the Source Oracle Database Meets Requirements

Oracle -> MySQL/Oracle -> PostgreSQL/Oracle -> GaussDB(for MySQL)/ Oracle -> GaussDB Migration and Synchronization

Table 1-146 Checking whether supplemental logging of the source oracle database meets requirements

Check Item	Whether supplemental logging of the source Oracle database meets requirements
Description	The supplemental logging of the source Oracle database must meet requirements. Otherwise, the migration or synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: The supplemental logging level of the source Oracle database does not meet requirements. Handling suggestion: Perform any of the following operations in the source database: ● Enable all supplemental logging at the database level: **alter database add supplemental log data (all) columns** ● Enable primary key and unique key supplemental logging: **alter database add supplemental log data (primary key,unique) columns** ● Enable minimal supplemental logging: **alter database add supplemental log data**; then run the following command to enable all, primary key, or unique key supplemental logging at the table level for each to-be-synchronized table: **alter table TABLE_NAME add supplemental log data(all) columns**

1.10 Conflicts

1.10.1 Checking Whether the Names of the Source and Destination Databases Are the Same

MySQL Migration

Table 1-147 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	Check whether the names of the source and destination databases are the same.

Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Handling suggestion: ● If you are migrating data to the cloud, determine whether to delete the databases with the same names as the source databases or specify a new destination DB instance based on site requirements. ● If you are migrating data out of the cloud, determine whether to use the original destination database or specify a new destination DB instance based on site requirements.
	Failure cause: During an incremental migration, the source and destination databases cannot have the same names. Handling suggestion: Determine whether to retain these databases in the destination RDS DB instance or specify another destination RDS DB instance.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

Oracle -> MySQL/Oracle -> GaussDB(for MySQL) Migration and Synchronization

Table 1-148 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	Check whether the source and destination databases have the same names to prevent existing databases from being overwritten. If the source and destination databases have the same name, the migration cannot be performed.
Failure Cause and Handl	Failure cause: Source database names are the same except for letter cases. Handling suggestion: Change table names or do not migrate the tables with the same names.

ing Sugge stion	Failure cause: Source database names are the same except for letter cases. Handling suggestion: Select only one database among the databases whose names are the same except for letter cases.
	Failure cause: The source and destination databases cannot have the same names. Handling suggestion: Determine whether to retain these databases in the destination RDS DB instance or specify another destination RDS DB instance.

Oracle -> MySQL/Oracle -> GaussDB(for MySQL) Synchronization

Table 1-149 Checking Whether the Names of the Source and Destination Databases Are the Same

Check Item	Whether the names of the source and destination databases are the same
Description	The mapping names of the tables to be synchronized in the source databases are the same as the names of the destination database tables.
Failure Cause and Handling Suggestion	Failure cause: The names of the tables to be synchronized are already mapped to the destination database. Handling suggestion: Select other tables to migrate or change the mapping names of the tables to be synchronized.

Oracle -> PostgreSQL Migration

Table 1-150 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	The migration fails when the names of the source and destination databases are different.
Failure Cause and Handl	Failure cause: The source and destination databases must have the same names, except that the destination database must use only lowercase letters. Handling suggestion: Create a database with name in lowercase letters in the destination database.

ing Suggestion	Failure cause: The names of the objects to be migrated are the same except for letter cases. Handling suggestion: Select only one database among the databases whose names are the same except for letter cases.
	Failure cause: The names of the tables to be migrated are the same except for letter cases. Handling suggestion: Change table names or do not migrate the tables with the same names.
	Failure cause: The names of the tables to be migrated are the same as those in the destination database and use only lowercase letters. Handling suggestion: Change table names or do not migrate the tables with the same names.

Oracle -> GaussDB T Migration

Table 1-151 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	The migration fails when the names of the source and destination databases are different.
Failure Cause and Handling Suggestion	Failure cause: Some tables cannot be migrated because tables with the same names already exist in the destination database. Handling suggestion: Select the tables that do not contain same table names or rename these tables. Command example: **alter table old_name rename to new_name**
	Failure cause: Some databases or users cannot be migrated because the destination databases do not have the databases or users with the same name. Handling suggestion: Create these databases or users before the migration, or do not migrate these databases or users.

PostgreSQL to PostgreSQL Migration and Synchronization

Table 1-152 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same

Description	Check whether the source and destination databases have the same names to prevent existing databases from being overwritten. If the source and destination databases have the same name, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The destination database names cannot be the same. Handling suggestion: Change the destination database names to prevent the database from being overwritten.

PostgreSQL to GaussDB(DWS) Synchronization

Table 1-153 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	Check whether the source and destination databases have the same names to prevent existing databases from being overwritten. If the source and destination databases have the same name, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The destination database names cannot be the same. Handling suggestion: Change the destination database names to prevent the database from being overwritten.

DDM -> Oracle Synchronization

Table 1-154 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	Before synchronization from DDM to Oracle, databases or users with the same names must be created in the destination database.

Failure Cause and Handling Suggestion	Failure cause: Some databases cannot be synchronized because the databases with the same names do not exist in the destination databases. Handling suggestion: Create these database or users in the destination databases or do not migrate these databases.

1.10.2 Checking Whether the Same View Names Exist in Both the Source and Destination Databases

Migration from MongoDB to DDS

Table 1-155 Checking whether the same view names exist in both the source and destination databases

Check Item	Whether the same view names exist in both the source and destination databases
Description	Check whether the source and destination databases have the same view names to prevent existing views from being overwritten. If view of the same name exists in the destination database, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The same view names exist in both the source and destination databases. Handling suggestions: Delete the destination database views that have the same names as those in the source database. Alternatively, do not migrate the views with the same names.

1.10.3 Checking Whether the Destination Database Contains a Non-Empty Collection with the Same Name As the Source Database

MongoDB Migration

Table 1-156 Checking whether the destination database contains a non-empty collection with the same name as the source database

Check Item	Whether the destination database contains a non-empty collection with the same name as the source database

Description	Check whether the source and destination databases have the same non-empty collections to prevent existing databases from being overwritten. If the source and destination databases have the same collections, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database failed to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database failed to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The same non-empty collections exist in both the source and destination databases. Handling suggestion: Determine whether to delete the same non-empty collections from the destination DB instance or specify a new destination DB instance.
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.

1.10.4 Checking Whether Destination Database Contains the Same Table Names As the Synchronization Objects

MySQL > MySQL

Table 1-157 Checking whether destination database contains the same table names as the synchronization objects

Check Item	Whether destination database contains the same table names as the synchronization objects. (table name conflicts)
Description	The destination database contains objects with the same name as those in the source database. If table of the same name exists in the destination database, the migration cannot be performed. Exceptions: If the names and structures of tables in the source and destination databases are the same, the system determines that no conflict occurs.
Failure Cause and Handling Suggestion	Failure cause: The source and destination database tables cannot have the same names. Handling suggestions: Check whether the tables with the same names need to be retained. If yes, select another object for synchronization. If no, delete the tables with the same names.

	Failure cause: The destination database contains the same table names as those of the synchronization objects. Handling suggestions: Check whether the tables with the same names need to be retained. If yes, select another object for synchronization. If no, delete the tables with the same names.
	Failure cause: Both the source and destination databases are RDS DB instance and do not have mapped databases. Handling suggestion: Create mappings for the databases that are not mapped.

1.10.5 Checking Whether the Destination Database Contains Objects with the Same Name As Those in the Source Database

MySQL -> PostgreSQL Synchronization

Table 1-158 Checking whether the destination database contains objects with the same name as those in the source database

Check Item	Whether destination database contains objects with the same name as those in the source database
Description	The destination database contains objects with the same name as those in the source database. If a table of the same name exists in the destination database, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The source and destination database tables cannot have the same names. Handling suggestions: Check whether the tables with the same names need to be retained. If yes, select another object for synchronization. If no, delete the tables with the same names.
	Failure cause: The destination database contains the same table names as those of the synchronization objects. Handling suggestions: Check whether the tables with the same names need to be retained. If yes, select another object for synchronization. If no, delete the tables with the same names.

Table-Level Synchronization from PostgreSQL to PostgreSQL

Table 1-159 Checking whether destination database contains objects with the same name as those in the source database

Check Item	Whether destination database contains objects with the same name as those in the source database
Description	The destination database contains objects with the same name as those in the source database If the same object names exist, the migration cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The destination database contains objects with the same name as those in the source database. Handling suggestions: Check whether the objects with the same names need to be retained. If yes, select another object for migration. If no, delete the objects with the same names.

1.10.6 Checking Whether Collections in Both the Source and Destination Databases Are Not Capped

MongoDB Migration

Table 1-160 Checking whether collections in both the source and destination databases are not capped

Check Item	Whether collections in both the source and destination databases are not capped
Description	Check whether collections in both the source and destination databases are not capped. If not, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: the destination database has a collection whose name is the same as that of the capped collection of the source database. Handling suggestion: To ensure data consistency, you can delete the collection in the destination database with the same name as the capped collection in the source database. Alternatively, you can choose not to migrate the capped collection that will contradict with that in the destination database.
	Failure cause: The collections to be migrated are capped collections and already exist in the destination database. Handling suggestion: To ensure data consistency, you can delete the collection in the destination database with the same name as the capped collection in the source database. Alternatively, you can choose not to migrate the capped collection that will contradict with that in the destination database.

1.11 SSL Connections

1.11.1 Checking Whether the SSL Connection Is Correctly Configured

MySQL

Table 1-161 Checking whether the SSL connection is correctly configured

Check Item	Whether the SSL connection is correctly configured
Description	Check whether the SSL connection is correctly configured for the source database.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The database is unavailable. Handling suggestion: Contact Huawei technical support.
	Item to be confirmed: The source database user must have the REQUIRE SSL permission when using the SSL connection. Handling suggestion: This alarm does not affect the migration process. If you require the SSL connection, you are advised to grant the REQUIRE SSL permission to the source database user.
	Item to be confirmed: The destination database user must have the REQUIRE SSL permission when using the SSL connection. Handling suggestion: This alarm does not affect the migration process. If you require the SSL connection, you are advised to grant the REQUIRE SSL permission to the destination database user.
	Failure cause: The source database user has the REQUIRE SSL permission but did not upload the encryption certificate. The SSL connection cannot be used. Handling suggestion: On the **Configure Source and Destination Databases** page, enable the SSL connection and upload the certificate, or change the source database user.

	Failure cause: The destination database user has the REQUIRE SSL permission but did not upload the encryption certificate. The SSL connection cannot be used. Handling suggestion: On the **Configure Source and Destination Databases** page, enable the SSL connection and upload the certificate, or change the destination database user.
	Item to be confirmed: Currently, the SSL connection is not enabled. DRS must ensure that the source database account allows non-SSL connections to the source database. Handling suggestion: Manually check whether the source database account allows non-SSL connections, or try to perform a migration. (By default, the source database account allows non-SSL connections.)
	Failure cause: An internal error occurs. Handling suggestion: Contact Huawei technical support.
	Failure cause: The SSL connection is enabled for the source database but no certificate has been uploaded. Handling suggestion: On the **Configure Source and Destination Databases** page, upload a certificate or disable the SSL connection for the source database.

MongoDB Migration

Table 1-162 Checking whether the SSL connection is enabled for the source database

Check Item	Whether the SSL connection is enabled for the source database
Description	Check whether the SSL connection is enabled for the source database.
Failure Cause and Handling Suggestion	Failure cause: SSL is enabled for the source database but no certificate has been uploaded. Handling suggestion: Upload a certificate on the **Configure Source and Destination Databases** page, or disable SSL for the source database.

1.11.2 Checking Whether the SSL Connection Is Enabled for the Source Database

PostgreSQL

Table 1-163 Checking whether the SSL connection is enabled for the source database

Check Item	Whether the SSL connection is enabled for the source database
Description	Check whether the SSL connection is enabled for the source database.
Failure Cause and Handling Suggestion	Failure cause: The source database SSL connection is disabled. Handling suggestion: In the **postgresql.conf** file, set **ssl_ca_file** to the directory of an SSL root CA certificate and set **ssl** to **on** to enable the SSL connection. Then, restart the database for the modifications to take effect.

1.11.3 Checking Whether the SSL Certificate of the Source Database Exists

MySQL > MySQL

Table 1-164 Checking whether the SSL certificate of the source database exists

Check Item	Whether the SSL certificate of the source database exists
Description	Check whether the SSL certificate type of the source database is correct during MySQL to MySQL synchronization. Otherwise, the synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: The source database uses SSL to encrypt connections but the SSL certificate does not exist. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the source database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.

	Failure cause: The SSL certificate type of the source database is not supported. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the source database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.

1.11.4 Checking Whether the SSL Certificate of the Destination Database Exists

MySQL

Table 1-165 Checking whether the SSL certificate of the destination database exists

Check Item	Whether the SSL certificate of the destination database exists
Description	Check whether the SSL certificate type of the destination database is correct during migration. Otherwise, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The SSL certificate of the destination database does not exist. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the destination database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.
	Failure cause: The SSL certificate type of the destination database is not supported. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the destination database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.

1.11.5 Checking Whether Both the Source and Destination Databases Use SSL

MongoDB Migration

Table 1-166 Checking whether both the source and destination databases use SSL to encrypt connections

Check Item	Whether both the source and destination databases use SSL to encrypt connections
Description	Check whether both the source and destination databases use SSL to encrypt connections.
Failure Cause and Handling Suggestion	Failure cause: Both the source and destination databases use SSL to encrypt connections. Handling suggestion: When migrating data to the cloud, disable SSL on the destination database. When migrating data out of the cloud, disable SSL on the source database. To disable SSL, go to the DDS basic information page.

1.12 Object Dependencies

1.12.1 Checking Whether the Objects Referenced by Views Are Selected for Migration

Migration from MongoDB to DDS

Table 1-167 Checking whether the objects referenced by views are selected for migration

Check Item	Whether the objects referenced by views are selected for migration
Description	The views and referenced objects should be migrated together. Otherwise, the migration fails.

Item to Be Confirmed and Handling Suggestion	Failure cause: The views to be migrated have dependencies on the objects that are not to be migrated. Handling suggestion: Select the referenced objects in the migration object list. Alternatively, do not migrate the dependent views.

1.12.2 Checking Whether Referenced Tables Are Selected for Migration

MySQL Migration and Synchronization

Table 1-168 Checking whether the tables referenced by the foreign key in the table to be migrated are selected for migration.

Check Item	Whether the tables referenced by the foreign key in the table to be migrated are selected for migration.
Description	The tables referenced by the foreign key in the table to be migrated are not selected for migration.
Item to Be Confirmed and Handling Suggestion	Failure cause: Tables referenced by the foreign key in the table to be migrated are not selected for migration. Handling suggestion: Select the referenced tables.

1.13 Source Database Information

1.13.1 Checking Whether the Shards and Mongos Are in the Same Cluster

MongoDB Migration

Table 1-169 Checking whether the shards and mongos are in the same cluster

Check Item	Whether the shards and mongos are in the same cluster
Description	If the shards and mongos are not in the same cluster, the migration will fail.

Item to Be Confirmed and Handling Suggestion	Failure cause: The shards are in different clusters from mongos. Handling suggestion: On the connection test page, enter the shards which are in the same cluster as the mongos.

1.13.2 Checking Whether the Balancers of the Source Database Is Enabled

MongoDB Migration

Table 1-170 Checking whether the balancers of the source database is enabled

Check Item	Whether the balancers of the source database are enabled
Description	If the source database contains the collections whose balancers are enabled, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Failure cause: Balancers are enabled for the collections in the source database. Handling suggestions: Disable balancers for the collections.

1.13.3 Checking Whether the Source and Destination Database Types Match

MongoDB Migration

Table 1-171 Checking whether the source and destination database types match

Check Item	Whether the source and destination database types match
Description	If the source database type does not match the destination database type, the migration task will fail.

Failure Cause and Handling Suggestion	Failure cause: The source and destination database types do not match. Handling suggestion: If the source DB instance type is cluster, ensure that the object type corresponding to the input IP address and port of the source database cluster is mongos, the source shard database type is replica set, and the destination database type is cluster.

2 Failure Cases

2.1 Case Overview

Table 2-1 Overview

Link	Related Documents
Real-Time Migration from MongoDB to DDS	Full Migration Error: Prematurely reached end of stream
	Full Migration Error: not authorized on riskevaluation to execute command {***}
	Full Migration Error: GC overhead limit exceeded
	Full Migration Error: The background process is unavailable. Maybe it has been killed manually or by the operating system. Please restart the task if possible or wait for restarting by itself within 5 minutes."
	Full Migration Error: Timed out after 60000 ms while waiting to connect
	Full or Incremental Migration Error: Timed out after 60000 ms while waiting to connect
	Full or Incremental Migration Error: Invalid BSON field name XXX
	Incremental Migration Error: Timed out after 60000 ms while waiting for a server that matches com.mongodb.client.internal
	Incremental Migration Error: Command failed with error ***(***):***. The full response is {***}"
Real-Time Migration and Synchron	Full Phase Error: Table '%.%' doesn't exist
	Full Phase Error: The background process is unavailable

Link	Related Documents
ization from MySQL to MySQL	Full Phase Error: Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
	Full Phase Error: Error writing file '***' (errno: 28 - No space left on device)
	Full Phase Error: The MySQL server is running with the --super-read-only option so it cannot execute this statement
	Full Phase Error: The table 't_diary_mem_font' is full
	Full Phase Error: Unknown column '%s' in 'field list'
	Full Phase Error: Lock wait timeout exceeded; try restarting transaction
	Full Phase Error: Java heap space
	Full Phase Error: Table '%s ' already exists
	Full Phase Error: temp table: %s.%s not exist
	Full Phase Error: failed to create new session
	Full Phase Error: load table: \`%s\`.\`%s\` failed
	Full Phase Error: read table=\`%s\`.\`%s\` failed
	Full Phase Error: CANNOT UPDATE USER WITH NULL PASSWORD
	Full Phase Error: Access denied for user '***' to database '***'
	Full Phase Error: The MySQL server is running with the --super-read-only option so it cannot execute this statement
	Full Phase Error: Temporary file write failure.
	Full Phase Error: Incorrect prefix key; the used key part isn't a string, the used length is longer than the key part, or the storage engine doesn't support unique prefix keys
	Full Phase Error: Unknown database '%s'
	Full Phase Error: Access denied; you need (at least one of) the SUPER privilege(s) for this operation
	Full Phase Error: retry structures failed events and Table '%s.%s' doesn't exist
	Full Phase Error: shard table=\`%s\`.\`%s\` failed
	Full Phase Error: Column name 'AUTO_PK_ROW_ID' is reserved.
	Full Phase Error: transfer account failed, can not find password from src DB

Link	Related Documents
	Full Phase Error: Access denied for user %s
	Full or Incremental Phase Error: binlog is not existed
	Full or Incremental Phase Error: Can not read response from server
	Full or Incremental Phase Error: Communications link failure
	Full or Incremental Phase Error: EOF Packet received, master disconnected
	Full or Incremental Phase Error: Extract db create sql failed
	Full or Incremental Phase Error: load table
	Full or Incremental Phase Error: Reached end of input stream
	Full or Incremental Phase Error: Read timed out
	Full or Incremental Phase Error: The background process is unavailable
	Full or Incremental Phase Error: Duplicate entry '%s' for key 'PRIMARY'
	Full or Incremental Phase Error: cause by: Index: %s, Size: %s
	Full or Incremental Phase Error: The offset and file name between src and parser is inconsistency
	Full or Incremental Phase Error: core process is not healthy or crashed
	Full or Incremental Phase Error: table info of table \`%s\`.\`%s\` from metadata miss
	Full or Incremental Phase Error: table[%s.%s]record field size for insert/delete dml
	Incremental Phase Error: write table %s.%s failed: Duplicate entry '%s' for key 'PRIMARY'
	Incremental Phase Error: not equals to target db column count
	Incremental Phase Error: The MySQL server is running with the --super-read-only option
	Incremental Phase Error: you need (at least one of) the SUPER privilege(s) for this operation
	Incremental Phase Error: Can't DROP '%s'; check that column/key exists
	Incremental Phase Error: Can't find file: '%s' (errno: 2 - No such file or directory)

Link	Related Documents
	Incremental Phase Error: Data truncation: Data too long for column
	Incremental Phase Error: Failed to read file header from
	Incremental Phase Error: Lock wait timeout exceeded
	Incremental Phase Error: Must seek before attempting to read next event
	Incremental Phase Error: Table '%s' already exists
	Incremental Phase Error: Table '%s' doesn't exist
	Incremental Phase Error: Table %s not found in database
	Incremental Phase Error: source has more columns than target
	Incremental Phase Error: Unknown storage engine
	Incremental Phase Error: Unknown table
	Incremental Phase Error: You have an error in your SQL syntax
	Incremental Phase Error: not illegal for mariaDb gtid position
	Incremental Phase Error: without PK execute failed
	Incremental Phase Error: Deadlock found when trying to get lock
	Incremental Phase Error: current serverUUID not equals to this session
Real-time Synchronization from MySQL to GaussDB(DWS)	Full Synchronization Error: Table %s.%s not found in database
	Full Synchronization Error: column 'database_table' of relation '%s' does not exist
	Full Synchronization Error: value too long for type character varying
	Full Synchronization Error: int1 has not implemented
	Full Synchronization Error: column name 'tid' conflicts with a system column name
	Full Synchronization Error: date/time field value out of range
	Full or Incremental Synchronization Error: service LOGMANAGER failed
	Full or Incremental Synchronization Error: service CAPTURER failed
	Full or Incremental Synchronization Error: ERROR: pooler

Link	Related Documents
	Incremental Synchronization Error: dn_%s_%s: column '%s' contains null values
	Incremental Synchronization Error: source has more columns than target
	Incremental Synchronization Errors: Connection to *.*.*.98:8000 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections
	Incremental Synchronization Error: Table %s not found in target database
	Incremental Synchronization Error: in a read-only transaction
	Incremental Synchronization Error: relation '%s' does not exist
	Incremental Synchronization Error: %s doesn't in the target table
	Incremental Synchronization Error: %s doesn't in the target table
	Incremental Synchronization Error: syntax error at or near
	Incremental Synchronization Error: schema '%s' does not exist
Real-Time Synchronization from PostgreSQL to PostgreSQL	Full Synchronization Error: function uuid_generate_v4() does not exist
	Full Synchronization Error: relation '%s' does not exist
	Full Synchronization Error: GC overhead limit exceeded
	Full Synchronization Error: Java heap space
	Full Synchronization Error: column '%s' of relation '%s' does not exist
	Full Synchronization Error: column '%s' does not exist
	Full Synchronization Error: type 'hstore' does not exist
	Full Synchronization Error: Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections
	Full Synchronization Error: invalid locale name
	Incremental Synchronization Error: Table %s not found in target database
	Incremental Synchronization Error: remaining connection slots are reserved
	Incremental Synchronization Error: PL/pgSQL function %s line %s at SQL statement

Link	Related Documents
Backup migration	**Backup Migration Failed Because Backup Files Cannot Be Found**
	Backup Migration Failed Because a Backup Database Cannot Be Found in the Backup Files

2.2 Real-Time Migration from MongoDB to DDS

2.2.1 Full Migration Error: Prematurely reached end of stream

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during full migration. The log information is as follows: service DATAMOVE failed, cause by: Prematurely reached end of stream.

Possible Causes

- The number of connections to the source or destination database is insufficient. Check the maximum number of connections to the source or destination database and the number of used connections. Generally, the number of connections used by DRS is about 10 on the source database and 20 on the destination database.

Solution

Adjust the number of connections supported by the source MongoDB database. For details, see the official documents of the source MongoDB database or the destination DDS database user guide.

2.2.2 Full Migration Error: not authorized on riskevaluation to execute command {***}

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during full migration. The log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=1, table_schema=***, table_name=***, record_num=2720] occur error, msg=Command failed with error 13 (Unauthorized):'not authorized on riskevaluation to execute command {***}.

Possible Causes

- The migration account used by DRS does not have the write permission on the destination database.

Solution

Grant the destination database write permission to the DRS migration account. For details, see the MongoDB official documents or DDS user guide.

2.2.3 Full Migration Error: GC overhead limit exceeded

Scenarios

During real-time MongoDB to DDS migration, an error is reported during full migration. The log information is as follows: service DATAMOVE failed, cause by: GC overhead limit exceeded.

Possible Causes

- The size of a single data record in the source database is too large.
- The replication instance specifications are too small.

Solution

Contact Huawei engineers to rectify the fault.

2.2.4 Full Migration Error: The background process is unavailable. Maybe it has been killed manually or by the operating system. Please restart the task if possible or wait for restarting by itself within 5 minutes."

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during full migration. The log information is as follows: service DATAMOVE failed, cause by: The background process is unavailable. Maybe it has been killed manually or by the operating system. Please restart the task if possible or wait for restarting by itself within 5 minutes.

Possible Causes

- The synchronization process is abnormal.

Solution

No further operation is required. The DRS daemon process automatically resumes the task. The migration and synchronization services are not affected, and data is resumed from the breakpoint. If the task is abnormal, click **Resume** in the **Operation** column to resume the task.

2.2.5 Full Migration Error: Timed out after 60000 ms while waiting to connect

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during full migration. The log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=144, table_schema=***, table_name=***, record_num=8510] occur error, msg=Timed out after 60000 ms while waiting to connect. Client view of cluster state is {type=UNKNOWN, servers=[{*** type=UNKNOWN, state=CONNECTING, exception={com.mongodb.MongoSocketOpenException: Exception opening socket}, caused by {java.net.ConnectException: Connection refused (Connection refused)}}].

Possible Causes

- The network is unstable. As a result, the connection times out when data is written to the destination database.
- The destination database is busy. As a result, the connection times out.

Solution

1. Check whether the destination database is running properly.
2. Check whether packet loss or retransmission occurs on the network between the DRS replication instance and the destination database.
3. Contact Huawei engineers to change the default timeout interval.

2.2.6 Full or Incremental Migration Error: Timed out after 60000 ms while waiting to connect

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during full or incremental migration. The log information is as follows: service LOGMANAGER failed, cause by: Timed out after 60000 ms while waiting to connect. Client view of cluster state is {type=UNKNOWN, servers=[{***, type=UNKNOWN, state=CONNECTING, exception={com.mongodb.MongoSocketOpenException: Exception opening socket}, caused by {java.net.ConnectException: Connection refused (Connection refused)}}]

Possible Causes

- The network is unstable. As a result, the connection to the source database times out.
- The source database is busy. As a result, the connection times out.

Solution

1. Check whether the source database is running properly.

2. Check whether packet loss or retransmission occurs on the network between the DRS replication instance and the source database.
3. Contact Huawei engineers to change the default timeout interval.

2.2.7 Full or Incremental Migration Error: Invalid BSON field name XXX

Scenarios

During real-time migration from MongoDB to DDS, an error is reported during full or incremental migration. The log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard={***}, table_schema={***}, table_name={***}, record_num={***}] occur error, msg=Invalid BSON field name {***}

Possible Causes

The field contains invalid characters, such as periods (.) and dollar signs ($).

Solution

Check and remove invalid symbols in the source database. In the task list, locate the target task and click **Resume** in the **Operation** column to resume the task.

2.2.8 Incremental Migration Error: Timed out after 60000 ms while waiting for a server that matches com.mongodb.client.internal

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during incremental migration. The log information is as follows: service INCREMENT failed, cause by: Timed out after 60000 ms while waiting for a server that matches com.mongodb.client.internal.MongoClientDelegate$1@27105e1a. Client view of cluster state is {type=REPLICA_SET, servers=[{address=***, type=UNKNOWN, state=CONNECTING, exception={com.mongodb.MongoSocketOpenException: Exception opening socket}, caused by {java.net.ConnectException: Connection timed out (Connection timed out)}}, {address=***, type=UNKNOWN, state=CONNECTING, exception={com.mongodb.MongoSocketOpenException: Exception opening socket}, caused by {java.net.NoRouteToHostExceptio.

Possible Causes

- The network is unstable. As a result, the connection times out when data is written to the destination database.
- The destination database is busy. As a result, the connection times out.

Solution

1. Check whether the destination database is running properly.

2. Check whether packet loss or retransmission occurs on the network between the DRS replication instance and the destination database.
3. Contact Huawei engineers to change the default timeout interval.

2.2.9 Incremental Migration Error: Command failed with error *** (***):***. The full response is {***}"

Scenarios

During real-time MongoDB-to-DDS migration, an error is reported during incremental migration. The log information is as follows: service INCREMENT failed, cause by: Command failed with error *** (***):***. The full response is {***}.

Possible Causes

The destination database returns an error. Common error codes are as follows:

- Error 91: The destination database service is abnormal.
- Error 133: The destination databaseshard is abnormal.

Solution

Contact destination database engineers to rectify the fault.

2.3 Real-Time Migration and Synchronization from MySQL to MySQL

2.3.1 Full Phase Error: Table '%.%' doesn't exist

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: Table ' %. %' doesn't exist.

Possible Causes

The possible causes are as follows:

- During the full migration phase, DDL statements are executed in the source database to delete tables.

Solution

Solution 1

During the full migration and synchronization phases, the DELETE command cannot be performed. For details about how to recreate a task, see **From MySQL to MySQL**.

Solution 2

Create a table with the same structure as the deleted table in the source database. In the task list on the **Online Migration Management** page, locate the task and click **Resume** in the **Operation** column.

2.3.2 Full Phase Error: The background process is unavailable

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: The background process is unavailable. Maybe it has been killed manually or by the operating system. Please restart the task if possible or wait for restarting by itself within 5 minutes.

Possible Causes

The possible causes are as follows:

- During the full migration or synchronization phase, the DRS process is terminated unexpectedly.

Solution

No further operation is required. The DRS daemon process automatically resumes the task. The migration and synchronization services are not affected, and data is resumed from the breakpoint. If the task is abnormal, click **Resume** in the **Operation** column to resume the task.

2.3.3 Full Phase Error: Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: Unable to connect to DBMS: url=jdbc:mysql://*** user=root, Caused by: Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.

Possible Causes

The possible causes are as follows:

- The connection to the source or destination database fails to be established.

Solution

1. Check whether the source or destination database is running properly.
2. Check whether the network connection between the DRS instance and the source or destination database is normal.

3. Check whether the whitelist of the source or destination database allows access from DRS instance IP addresses.

2.3.4 Full Phase Error: Error writing file '***' (errno: 28 - No space left on device)

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply data of table=` %s`.` %s` failed: Error writing file '***' (errno: 28 - No space left on device).

Possible Causes

The possible causes are as follows:

- The destination database storage space is insufficient. As a result, data fails to be written to the destination database.

Solution

Contact destination database O&M engineers to rectify the fault.

2.3.5 Full Phase Error: The MySQL server is running with the --super-read-only option so it cannot execute this statement

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply data of table=` %s`.` %s` failed: The MySQL server is running with the --super-read-only option so it cannot execute this statement.

Possible Causes

The possible causes are as follows:

- The destination database is read-only. The possible cause is that the space of the destination database is insufficient.

Solution

Contact destination database O&M engineers to rectify the fault.

2.3.6 Full Phase Error: The table 't_diary_mem_font' is full

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=1, table_schema= %s, table_name=

%s, record_num=3585] occur error, msg=apply data of table=` %s`.` %s` failed: The table 't_diary_mem_font' is full.

Possible Causes

The possible causes are as follows:

- The destination database storage space is insufficient. As a result, data fails to be written to the destination database.

Solution

Contact destination database O&M engineers to rectify the fault.

2.3.7 Full Phase Error: Unknown column '%s' in 'field list'

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=1, table_schema= %s, table_name= %s, record_num=5] occur error, msg=apply data of table=` %s`.` %s` failed: Unknown column ' %s' in'field list'.

Possible Causes

The possible causes are as follows:

- The table structures of the source and destination databases are inconsistent. The possible cause is that DDL is executed on the columns of the destination database table during full synchronization or the table consistency check is skipped during pre-check.

Solution

Contact the O&M personnel to modify the table structure of the destination database to be the same as that of the source database. After the modification is complete, click **Resume** in the **Operation** column to resume the task.

2.3.8 Full Phase Error: Lock wait timeout exceeded; try restarting transaction

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=***, table_schema= %s, table_name= %s, record_num=***] occur error, msg=apply data of table=` %s `.` %s ` failed: Lock wait timeout exceeded; try restarting transaction.

Possible Causes

A lock wait times out when data is written to the destination database. The possible causes are as follows:

- The service connection of the destination database holds the lock for a long time.
- The performance of the destination database is insufficient or the load is heavy, and the execution is slow.

Solution

Contact the customer's O&M personnel to check the lock usage, slow SQL statements, or load status of the destination database.

2.3.9 Full Phase Error: Java heap space

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=***, table_schema= %s, table_name= %s, record_num=***] occur error, msg=apply data of table=` %s `.` %s ` failed: Java heap space.

Possible Causes

The possible causes are as follows:

- The size of a single record exceeds 50 MB.

Solution

Contact Huawei technical support engineers.

2.3.10 Full Phase Error: Table '%s ' already exists

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_rename_or_copy, index=null, schema_name= %s, object_name= %s] occur error, msg=rename table %s. %s error: Table '%s ' already exists.

Possible Causes

The possible causes are as follows:

- The table without a primary key already exists in the destination database.

Solution

Contact the O&M engineers to delete the tables that do not have primary keys from the destination database table. After that, click **Resume** in the **Operation** column to resume the task.

2.3.11 Full Phase Error: temp table: %s.%s not exist

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_rename_or_copy, index=null, schema_name= %s, object_name= %s] occur error, msg=temp table: %s. %s not exist

Possible Causes

The possible causes are as follows:

- An exception occurred when a table without a primary key is being migrated.

Solution

Contact Huawei technical support engineers.

2.3.12 Full Phase Error: failed to create new session

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: com.continuent.tungsten.replicator.ReplicatorException: Unable to connect to DBMS: url=jdbc:mysql://*** user=***, Caused by: failed to create new session.

Possible Causes

The possible causes are as follows:

- The connection to the source or destination database fails to be established.

Solution

1. Check whether the source or destination database is running properly.
2. Check whether the network connection between the DRS instance and the source or destination database is normal.
3. Check whether the whitelist of the source or destination database allows access from the IP address of the DRS instance.

2.3.13 Full Phase Error: load table: `%s`.`%s` failed

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: load table: ` %s`.`%s` failed.

Possible Causes

The possible causes are as follows:

- The table structure of the source database fails to be loaded. The possible cause is that the user deletes tables during synchronization or the source database user does not have required permissions.

Solution

1. Contact the O&M engineers to check whether the table structure of the source database is normal. The common commands are as follows:
   ```
   SELECT * FROM `%s`.`%s` LIMIT 1
   SHOW CREATE TABLE `%s`.`%s`
   ```
2. Contact the source database administrator to check whether the source database and tables have been deleted. If they were deleted, recreate the task by referring to **Precautions**.
3. Check whether the migration account has the SHOW CREATE TABLE permission on the source database tables. If the account does not have the permission, grant the permission to the source database migration account by referring to **Precautions**. Then, in the task list, click **Resume** in the **Operation** column to resume the task.

2.3.14 Full Phase Error: read table=`%s`.`%s` failed

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: read table=` %s`.`%s` failed.

Possible Causes

The possible causes are as follows:

- Failed to read table data from the source database due to poor source database performance or unstable network connection.

Solution

Contact Huawei technical support to adjust the timeout interval for accessing the source database.

2.3.15 Full Phase Error: CANNOT UPDATE USER WITH NULL PASSWORD

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=account, index=0, schema_name=mysql, object_name='***']reason:[CANNOT UPDATE USER WITH NULL PASSWORD].

Possible Causes

The possible causes are as follows:

- The source database account password cannot be empty.

Solution

Contact the O&M engineer to add a password for the account that reports the error in the source database. Then, in the task list, click **Resume** in the **Operation** column to resume the task.

2.3.16 Full Phase Error: Access denied for user '***' to database '***'

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=account, index=0, schema_name=mysql, object_name='***']reason:[Access denied for user '***' to database '***']

Possible Causes

The possible causes are as follows:

- The DRS migration account does not have sufficient permissions on the destination database.

Solution

Contact the O&M engineers to add the schema permission to the migration account in the destination database. Then, in the task list, locate the row that contains the target task and click **Resume** in the **Operation** column to resume the task.

2.3.17 Full Phase Error: The MySQL server is running with the --super-read-only option so it cannot execute this statement

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=constraint_data, index=0, schema_name= %s, object_name= %s]reason:[The MySQL server is running with the --super-read-only option so it cannot execute this statement]

Possible Causes

The possible causes are as follows:

- When DRS migrates indexes, the destination database is in the read-only state. The possible cause is that the space of the destination database is insufficient.

Solution

Contact the O&M engineers to check the running status and disk space of the destination database.

2.3.18 Full Phase Error: Temporary file write failure.

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=constraint_data, index=0, schema_name= %s, object_name= %s]reason:[Temporary file write failure.]

Possible Causes

The possible causes are as follows:

- The temporary space of the destination database is insufficient when DRS migrates indexes.

Solution

Contact the O&M engineers to check the temporary space of the destination database.

2.3.19 Full Phase Error: Incorrect prefix key; the used key part isn't a string, the used length is longer than the key part, or the storage engine doesn't support unique prefix keys

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=constraint_data, index=106, schema_name= %s, object_name= %s]reason:[Incorrect prefix key; the used key part isn't a string, the used length is longer than the key part, or the storage engine doesn't support unique prefix keys]

Possible Causes

The possible causes are as follows:

- Table structures in the source and destination databases are inconsistent.

Solution

Contact the O&M engineers to change the table structure of the destination database to be the same as that of the source database. Then, in the task list, locate the target task and click **Resume** in the **Operation** column to resume the task.

2.3.20 Full Phase Error: Unknown database '%s'

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=constraint_data, index=4, schema_name= %s, object_name= %s]reason:[Unknown database ' %s']

Possible Causes

The possible causes are as follows:

- The schema corresponding to the destination database structure does not exist.

Solution

Check whether the O&M engineers have deleted the destination database schema and manually create a schema that is the same as that of the source database. After the schema is created, click **Resume** in the **Operation** column to resume the task.

2.3.21 Full Phase Error: Access denied; you need (at least one of) the SUPER privilege(s) for this operation

Scenarios

During a full migration, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=function, index=2, schema_name= %s, object_name= %s]reason:[Access denied; you need (at least one of) the SUPER privilege(s) for this operation]

Possible Causes

The possible causes are as follows:

- The user retained the definer for migration, but the migration account does not have the super permission.

Solution

Grant the super permission to the destination database user and submit the task again. For details, see **RDS FAQs**. Alternatively, you can choose not to migrate definers when recreating a task.

2.3.22 Full Phase Error: retry structures failed events and Table '%s.%s' doesn't exist

Scenarios

During a full migration, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=view, index=2, schema_name= %s, object_name= %s]reason:[Table ' %s. %s' doesn't exist]

Possible Causes

The possible causes are as follows:

- The user retained the definer for migration, but the definer is abnormal or does not exist.

Solution

Recreate a task and do not migrate definers.

2.3.23 Full Phase Error: shard table=`%s`.`%s` failed

Scenarios

During a full migration or synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: shard table=`%s`.`%s` failed

Possible Causes

The possible causes are as follows:

- The source database performance is insufficient or the network is unstable. As a result, the source database sharding times out.

Solution

Contact DRS O&M personnel.

2.3.24 Full Phase Error: Column name 'AUTO_PK_ROW_ID' is reserved.

Scenarios

An error is reported during full migration or synchronization, and the following log information is displayed: service LOGMANAGER failed, cause by: create table xxx error. Column name 'AUTO_PK_ROW_ID' is reserved. Operation 'CREATE' is not permitted.

Possible Causes

The possible cause is as follows:

- The **AUTO_PK_ROW_ID** column name is a reserved column name for the RDS for MySQL database and cannot be created by users.

Solution

- Check the tables whose column names contain **AUTO_PK_ROW_ID** in the source database, change the column names, and resume the task.
- Create a task again and do not select the tables whose column names contain **AUTO_PK_ROW_ID**.

2.3.25 Full Phase Error: transfer account failed, can not find password from src DB

Scenarios

During a full migration, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: transfer account failed, can not find password from src DB.

Possible Causes

- The RDS security policy does not allow the user password to be empty. However, if the source database is an on-premises MySQL database, the user password can be empty.

Solution

Step 1 Run the following SQL statement in the source database to query users whose passwords are empty:

- MySQL 5.7 or later versions:

```
SELECT USER,HOST,authentication_string FROM MYSQL.user WHERE authentication_string
IS NULL OR authentication_string='';
```

- MySQL 5.6 and earlier versions

```
SELECT USER,HOST,`password` FROM MYSQL.user WHERE `password` IS NULL OR
`password`='';
```

Step 2 Run the following SQL statement to delete the users whose passwords are empty from the source database or set passwords for the users:

- Delete a user whose password is empty.

```
DROP USER XXX@XXX;
```

- Set a password for a user.

```
ALTER USER xxx@xxx IDENTIFIED BY xxx;
```

----End

2.3.26 Full Phase Error: Access denied for user %s

Scenarios

During a full or increment migration or synchronization, an error is reported. The log information is as follows: service %s failed, cause by: Unable to connect to DBMS: url=***?useUnicode=true&allowLoadLocalInfile=false&characterEncoding=UTF-8&connectTimeout=5000&useSSL=false&allowPublicKeyRetrieval=true&verifyServerCertificate=false&serverTimezone=UTC user=%s, Caused by: Access denied for user %s

Possible Causes

The possible cause is as follows:

- Failed to connect to the source or destination database.

Solution

1. Check whether the source or destination database is running properly.
2. Check whether the password for connecting to the source or destination database is correct.
3. Check whether the network connection between the DRS instance and the source database is normal.
4. Check whether the IP address of the DRS instance is allowed to access the source or destination database.

2.3.27 Full or Incremental Phase Error: binlog is not existed

Scenarios

During a full or incremental migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: binlog is not existed

Possible Causes

The possible causes are as follows:

- The binlog files in the source database were deleted. DRS cannot obtain logs from the source database.

Solution

Recreate a DRS task.

2.3.28 Full or Incremental Phase Error: Can not read response from server

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: Can not read response from server. Expected to read 4 bytes, read 0 bytes before connection was unexpectedly lost.

Possible Causes

The possible causes are as follows:

- The network bandwidth between the DRS replication instance and the source database is too small or unstable.
- The source database is overloaded.

Solution

Contact the source database administrator to check the source database load and check whether packet loss occurs on the network between the source database and the replication instance.

2.3.29 Full or Incremental Phase Error: Communications link failure

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service %s failed, cause by: Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.

Possible Causes

The possible causes are as follows:

- Failed to connect to the source database.

Solution

1. Check whether the source database is running properly.
2. Check whether the network connection between the DRS instance and the source database is normal.

2.3.30 Full or Incremental Phase Error: EOF Packet received, master disconnected

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: EOF Packet received, master disconnected

Possible Causes

The possible causes are as follows:

- The source database is abnormal.
- The binlog file is abnormal.

Solution

Contact the source database administrator to check whether the source database is running properly.

2.3.31 Full or Incremental Phase Error: Extract db create sql failed

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: Extract db create sql failed, dbName = %s

Possible Causes

The possible causes are as follows:

- During full migration, the user deleted databases from the source.
- The source database user does not have the permission to perform operations on the source database.

Solution

- Contact the source database administrator to check whether the source database has been deleted. If the source database was deleted, recreate the task by referring to the related section in the product documentation.
- Check whether the source database user has the SHOW CREATE TABLE permission on the source database table. If the user does not have the permission, grant the permission to the user and retry the DRS task.

2.3.32 Full or Incremental Phase Error: load table

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: load table: ` %s`.` %s` failed

Possible Causes

The possible causes are as follows:

- During full migration, the user deleted databases or tables from the source.
- The source database user does not have the permission to perform operations on the source database or table.

Solution

- Contact the source database administrator to check whether the source database or table has been deleted. If the source database or table was deleted, recreate the task by referring to the related section in the product documentation.
- Check whether the source database user has the SHOW CREATE TABLE permission on the source database table. If not, grant the permission to the user and retry the DRS task.

2.3.33 Full or Incremental Phase Error: Reached end of input stream

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: Reached end of input stream

Possible Causes

The possible causes are as follows:

- The source database is abnormal.
- The binlog file is abnormal.

Solution

Contact the source database administrator to check whether the source database is running properly.

2.3.34 Full or Incremental Phase Error: Read timed out

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, cause by: Read timed out

Possible Causes

The possible causes are as follows:

- Failed to connect to the source database.

Solution

1. Check whether the source database is running properly.
2. Check whether the network connection between the DRS instance and the source database is normal.

2.3.35 Full or Incremental Phase Error: The background process is unavailable

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service %s failed, cause by: The background process is unavailable. Maybe it has been killed manually or by the operating system. Please restart the task if possible or wait for restarting by itself within 5 minutes.

Possible Causes

The possible causes are as follows:

- During the migration, the DRS process stops unexpectedly.

Solution

No further operation is required. The DRS daemon process automatically resumes the task. The migration and synchronization services are not affected, and data is resumed from the breakpoint. If the task is abnormal, click **Resume** in the **Operation** column to resume the task.

2.3.36 Full or Incremental Phase Error: Duplicate entry '%s' for key 'PRIMARY'

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, cause by: Duplicate entry '120' for key 'PRIMARY'.

Possible Causes

The possible causes are as follows:

- **binlog_format** in the source database is not set to **ROW**.
- The **binlog_format** setting of the source database does not take effect immediately.

Solution

Step 1 Log in to the source database using the MySQL official client or other tools.

Step 2 Run the following command for setting global parameters in the source database.

```
set global binlog_format = ROW;
```

Step 3 Run the following command on the source database and check whether the preceding operation is successful:

```
select @@global.binlog_format;
```

Step 4 You can use either of the following methods to ensure that the modified binlog format of the source database takes effect immediately:

Method 1

1. Select a non-service period to disconnect all service connections on the current database.
   a. Run the following command to query all service threads (excluding all binlog dump threads and current threads) in the current database:
   ```
   show processlist;
   ```
   b. Stop all the service threads queried in the previous step.

   NOTE

   Do not create or start a migration task before the preceding operations are complete. Otherwise, data may be inconsistent.
2. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.
   ```
   binlog_format=ROW
   ```

Method 2

1. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the

startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.

```
binlog_format=ROW
```

2. Ensure that the **binlog_format** parameter is successfully added or modified. Then, restart the source database at a non-service period.

----End

2.3.37 Full or Incremental Phase Error: cause by: Index: %s, Size: %s

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, cause by: Index: 8, Size: 7

Possible Causes

The possible causes are as follows:

- **binlog_format** in the source database is not set to **ROW**.
- The binlog_format setting of the source database does not take effect immediately.

Solution

Step 1 Log in to the source database using the MySQL official client or other tools.

Step 2 Run the following command for setting global parameters in the source database.

```
set global binlog_format = ROW;
```

Step 3 Run the following command on the source database and check whether the preceding operation is successful:

```
select @@global.binlog_format;
```

Step 4 You can use either of the following methods to ensure that the modified binlog format of the source database takes effect immediately:

Method 1

1. Select a non-service period to disconnect all service connections on the current database.
   a. Run the following command to query all service threads (excluding all binlog dump threads and current threads) in the current database:
   ```
   show processlist;
   ```
   b. Stop all the service threads queried in the previous step.

   NOTE

   Do not create or start a migration task before the preceding operations are complete. Otherwise, data may be inconsistent.

2. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the

startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.

```
binlog_format=ROW
```

Method 2

1. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.

   ```
   binlog_format=ROW
   ```

2. Ensure that the **binlog_format** parameter is successfully added or modified. Then, restart the source database at a non-service period.

----End

2.3.38 Full or Incremental Phase Error: The offset and file name between src and parser is inconsistency

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, cause by: The offset and file name between src and parser is inconsistency

Possible Causes

The possible causes are as follows:

- The source database is abnormal.
- The binlog file is abnormal.

Solution

Contact the source database administrator to check whether the source database is running properly.

2.3.39 Full or Incremental Phase Error: core process is not healthy or crashed

Scenarios

During a full or increment migration or synchronization, a DRS task error is reported, and the following log information is displayed: service CAPTURER failed, cause by: core process is not healthy or crashed

Possible Causes

The possible causes are as follows:

- During the migration, the DRS process stops unexpectedly.

Solution

No further operation is required. The DRS daemon process automatically resumes the task. The migration and synchronization services are not affected, and data is resumed from the breakpoint. If the task is abnormal, click **Resume** in the **Operation** column to resume the task.

2.3.40 Full or Incremental Phase Error: table info of table \`%s\`.\`%s\` from metadata miss

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, cause by: table info of table \`%s\`.\`%s\` from metadata miss

Possible Causes

The possible causes are as follows:

- The table may fail to be created due to DDL syntax incompatibility.

Solution

Contact Huawei technical support engineers.

2.3.41 Full or Incremental Phase Error: table[%s.%s]record field size for insert/delete dml

Scenarios

During a full or increment migration or synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, cause by: table[%s.%s]record field size for insert/delete dml=%s, the column size in dictionary=%s

Possible Causes

The possible causes are as follows:

- Full image is not enabled for the source database binlog.

Solution

For details, see **How Do I Set binlog_row_image=FULL to Take Effect Immediately?**

2.3.42 Incremental Phase Error: write table %s.%s failed: Duplicate entry '%s' for key 'PRIMARY'

Scenarios

During an incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: write table %s. %s failed: Duplicate entry ' %s' for key 'PRIMARY'

Possible Causes

The possible causes are as follows:

- A primary key conflict is reported when the synchronization policy is selected.

Solution

The migration task needs to be recreated.

2.3.43 Incremental Phase Error: not equals to target db column count

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: write table %s. %s failed: table= %s. %s has field list size=[***] not equals to target db column count= %s

Possible Causes

The possible causes are as follows:

- DDL is executed on the destination database table, causing the table structure in the destination database to be inconsistent with that in the source database.

Solution

Contact the O&M engineers of the destination database to change the table structure of the destination database to be the same as that of the source database. Then, in the task list, locate the target task and click **Resume** in the **Operation** column to resume the task.

2.3.44 Incremental Phase Error: The MySQL server is running with the --super-read-only option

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: write table %s.

%s failed: record tid: %s,seqno: %s with PK applied failed in table %s. %s, The MySQL server is running with the --super-read-only option so it cannot execute this statement

Possible Causes

The possible causes are as follows:

- The destination database is in the read-only state. Generally, the destination database storage is insufficient.

Solution

Contact the customer's O&M engineers to check the running status and disk space of the destination database.

2.3.45 Incremental Phase Error: you need (at least one of) the SUPER privilege(s) for this operation

Scenarios

During an incremental migration, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Access denied; you need (at least one of) the SUPER privilege(s) for this operation; sql is: CREATE DEFINER= %s

Possible Causes

The possible causes are as follows:

- The user retained the definer for migration, but the definer is abnormal or does not exist.

Solution

Contact the customer's O&M engineers to check the running status and disk space of the destination database.

2.3.46 Incremental Phase Error: Can't DROP '%s'; check that column/key exists

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Can't DROP ' %s'; check that column/key exists; sql is %s

Possible Causes

The possible causes are as follows:

- DDL is executed on the destination database table, causing the table structure in the destination database to be inconsistent with that in the source database.

Solution

Contact the O&M engineers of the destination database to change the table structure of the destination database to be the same as that of the source database. Then, in the task list, locate the target task and click **Resume** in the **Operation** column to resume the task.

2.3.47 Incremental Phase Error: Can't find file: '%s' (errno: 2 - No such file or directory)

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Can't find file: ' %s' (errno: 2 - No such file or directory); sql is: %s

Possible Causes

The possible causes are as follows:

- The destination database table file is damaged.

Solution

Contact destination database O&M engineers of the customer to check whether the corresponding table exists and is normal.

2.3.48 Incremental Phase Error: Data truncation: Data too long for column

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Data truncation: Data too long for column ' %s' at row %s; sql is: %s

Possible Causes

The possible causes are as follows:

- The DDL statement fails to be executed because the statement length is too long.

Solution

Contact destination database O&M engineers to check the structure of the table where the synchronization error occurs, adjust the length and data type of the column where the error is reported. Then, in the task list, click **Resume** in the **Operation** column of the corresponding task to retry the task.

2.3.49 Incremental Phase Error: Failed to read file header from

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Failed to read file header from thl.data.0000000011

Possible Causes

The possible causes are as follows:

- The format of the DRS data file is damaged.

Solution

Contact Huawei technical support.

2.3.50 Incremental Phase Error: Lock wait timeout exceeded

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Lock wait timeout exceeded; try restarting transaction

Possible Causes

The possible cause is as follows:

- The lock wait times out when the destination database is accessed.

Solution

Contact destination database O&M engineers to check the status and load of the destination database.

2.3.51 Incremental Phase Error: Must seek before attempting to read next event

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Must seek before attempting to read next event

Possible Causes

The possible causes are as follows:

- The task is interrupted for a long time. Historical DRS data files were deleted and the task cannot be continued.

Solution

Contact the user to re-create the task.

2.3.52 Incremental Phase Error: Table '%s' already exists

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Table ' %s' already exists

Possible Causes

The possible cause is as follows:

- A table has been created in the destination database. As a result, an error is reported when the DDL statement for creating a table in the source database is executed.

Solution

Contact destination database O&M engineers to delete the destination database table. Then, in the task list, locate the row that contains the task and click **Resume** in the **Operation** column to resume the task.

2.3.53 Incremental Phase Error: Table '%s' doesn't exist

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Table ' %s' doesn't exist; sql is: create %s like matches

Possible Causes

The possible causes are as follows:

- Tables are deleted from the destination database. As a result, the synchronization statement reports an error.

Solution

Contact the destination database O&M engineers to create a table in the destination database based on the table structure of the source database. After the table is created, click **Resume** in the **Operation** column to resume the task.

2.3.54 Incremental Phase Error: Table %s not found in database

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Table %s not found in database

Possible Causes

The possible causes are as follows:

- Tables are deleted from the destination database. As a result, the synchronization statement reports an error.

Solution

Contact the destination database O&M engineers to create a table in the destination database based on the table structure of the source database. After the table is created, click **Resume** in the **Operation** column to resume the task.

2.3.55 Incremental Phase Error: source has more columns than target

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Table %s in source has more columns than target

Possible Causes

The possible causes are as follows:

- The table structure is modified in the destination database. As a result, the synchronization statement reports an error.

Solution

Contact destination database O&M engineers to change the table structure of the destination database to be the same as that of the source database. After the modification, click **Resume** in the **Operation** column resume the task.

2.3.56 Incremental Phase Error: Unknown storage engine

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Unknown storage engine 'FEDERATED'; sql is: %s

Possible Causes

The possible causes are as follows:

- When the DDL table creation statement of the source database is replayed in destination database, the source DB engine is not supported by the destination database.

Solution

Contact the user to create a table that supports the DB engine in the destination database and contact Huawei technical support to skip the DDL statement that reports the error. Then, click **Resume** in the **Operation** column to resume the task.

2.3.57 Incremental Phase Error: Unknown table

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Unknown table ' %s'; sql is %s

Possible Causes

The possible causes are as follows:

- The table does not exist in the destination database.

Solution

Contact the user to create the table in the destination database based on the table structure of the source database. After the table is created, click **Resume** in the **Operation** column to resume the task.

2.3.58 Incremental Phase Error: You have an error in your SQL syntax

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near'START TRANSACTION' at line 39

Possible Causes

The possible causes are as follows:

- During the pre-check, the system skips version check. The syntax of the later version fails to be executed in the earlier version.

Solution

Contact the user to modify the statement based on the destination database syntax and run the statement in the destination database. Then, contact Huawei engineers to skip this error and click **Resume** in the **Operation** column to resume the task.

2.3.59 Incremental Phase Error: not illegal for mariaDb gtid position

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: %s not illegal for mariaDb gtid position

Possible Causes

The possible causes are as follows:

- The gtid mode is changed during task creation.

Solution

Contact the user to recreate the task.

2.3.60 Incremental Phase Error: without PK execute failed

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: record tid: %s,seqno: %s without PK execute failed in table %s

Possible Causes

The possible causes are as follows:

- A conflict occurs during data synchronization for tables that do not have primary keys.

Solution

Contact Huawei technical support.

2.3.61 Incremental Phase Error: Deadlock found when trying to get lock

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: record tid:

%s,seqno: %s with PK applied failed in table %s, Deadlock found when trying to get lock; try restarting transaction

Possible Causes

The possible causes are as follows:

- A deadlock occurs in the destination database.

Solution

Contact destination database O&M engineers to check the status and load of the destination database.

2.3.62 Incremental Phase Error: current serverUUID not equals to this session

Scenarios

During an incremental migration or synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: write table %s failed: current serverUUID not equals to this session

Possible Causes

The possible causes are as follows:

- The destination database had a switchover.

Solution

Contact destination database O&M engineers to check the destination database status and retry the DRS task.

2.4 Real-time Synchronization from MySQL to GaussDB(DWS)

2.4.1 Full Synchronization Error: Table %s.%s not found in database

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard=1, table_schema= %s, table_name= %s, record_num= %s] occur error, msg=Table %s. %s not found in database. Unable to generate a valid statement.

Possible Causes

The possible causes are as follows:

- The destination database table is not created. As a result, the synchronization statement reports an error.

Solution

Contact destination database O&M engineers to create the corresponding table in the destination database.

2.4.2 Full Synchronization Error: column 'database_table' of relation '%s' does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard= %s, table_schema= %s, table_name= %s, record_num= %s] occur error, msg=apply to %s failed: ERROR: column 'database_table' of relation ' %s' does not exist.

Possible Causes

The possible causes are as follows:

- No additional column is added when a table is created in the destination database.

Solution

Contact destination database O&M engineers to add additional columns to the destination database table.

2.4.3 Full Synchronization Error: value too long for type character varying

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard= %s, table_schema= %s, table_name= %s, record_num= %s] occur error, msg=apply to %s failed: ERROR: value too long for type character varying(%s) Where: COPY %s, line 1, column remarks.

Possible Causes

The possible causes are as follows:

- The length of the varchar column in the user-defined table in the destination database is insufficient. MySQL and GaussDB(DWS) define the length of the varchar data type differently. MySQL defines the length of the varchar data type as the number of characters, while GaussDB (DWS) defines the length of the varchar data type as the number of bytes.

Solution

Contact destination database O&M engineers to increase the field precision in the destination database table.

2.4.4 Full Synchronization Error: int1 has not implemented

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard= %s, table_schema= %s, table_name= %s, record_num= %s] occur error, msg=apply to %s failed: column: %s, invalid value: %s, int1 has not implemented, type is %s.

Possible Causes

The possible causes are as follows:

- User-defined table fields in the destination database do not support int1.

Solution

Contact destination database O&M engineers to create a table in the destination database based on the synchronization link mappings.

2.4.5 Full Synchronization Error: column name 'tid' conflicts with a system column name

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=table_structure, index=%s, schema_name=%s, object_name= %s]reason:[ERROR: column name 'tid' conflicts with a system column name]

Possible Causes

The possible causes are as follows:

- tid is a reserved character of GaussDB(DWS) and exists in the source database table.

Solution

This scenario is not supported due to the restrictions of the destination database. You can change the column names of the source database table.

2.4.6 Full Synchronization Error: date/time field value out of range

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=table_structure, index= %s, schema_name= %s, object_name= %s]reason:[ERROR: date/time field value out of range: '0000-00-00 00:00:00'].

Possible Causes

The possible causes are as follows:

- GaussDB(DWS) does not support 0000-00-00 00:00:00.

Solution

This scenario is not supported due to the restrictions of the destination database. You can change the column names of the source database table.

2.4.7 Full or Incremental Synchronization Error: service LOGMANAGER failed

Scenarios

During full or incremental synchronization, an error is reported, and the log information is as follows: service LOGMANAGER failed, ***.

Solution

For synchronization from MySQL to GaussDB(DWS), the cause of the LOGMANAGER error is the same as that of the MySQL to MySQL synchronization. You can rectify the fault based on the error keyword.

- **Full Phase Error: Column name 'AUTO_PK_ROW_ID' is reserved.**
- **Full or Incremental Phase Error: binlog is not existed**
- **Full or Incremental Phase Error: Can not read response from server**
- **Full or Incremental Phase Error: EOF Packet received, master disconnected**
- **Full or Incremental Phase Error: Extract db create sql failed**
- **Full or Incremental Phase Error: Reached end of input stream**
- **Full or Incremental Phase Error: Read timed out**

2.4.8 Full or Incremental Synchronization Error: service CAPTURER failed

Scenarios

During full or incremental synchronization, an error is reported, and the log information is as follows: service CAPTURER failed, ***.

Solution

For synchronization from MySQL to GaussDB(DWS), the cause of the CAPTURER error is the same as that of the MySQL to MySQL synchronization. You can rectify the fault based on the error keyword.

- **Full or Incremental Phase Error: Duplicate entry '%s' for key 'PRIMARY'**
- **Full or Incremental Phase Error: cause by: Index: %s, Size: %s**
- **Full or Incremental Phase Error: The offset and file name between src and parser is inconsistency**
- **Full or Incremental Phase Error: core process is not healthy or crashed**
- **Full or Incremental Phase Error: table info of table `%s`.`%s` from metadata miss**
- **Full or Incremental Phase Error: table[%s.%s]record field size for insert/delete dml**

2.4.9 Full or Incremental Synchronization Error: ERROR: pooler

Scenarios

During full or incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: %s failed:tid: %s, sqno: %s, ERROR: pooler: %s.

Possible Causes

The possible causes are as follows:

- The GaussDB(DWS) database returns an error.

Solution

Contact the destination GaussDB(DWS) database O&M engineers to rectify the fault.

2.4.10 Incremental Synchronization Error: dn_%s_%s: column '%s' contains null values

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: table %s execute the ddl failed ERROR: dn_ %s_ %s: column ' %s' contains null values.

Possible Causes

The possible causes are as follows:

- The destination GaussDB(DWS) database has the NOT NULL constraint, so the default value needs to be set.

Solution

Modify the DDL statement and manually run it in the destination database. Then contact Huawei engineers to skip the synchronization of this DDL statement.

2.4.11 Incremental Synchronization Error: source has more columns than target

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: INCREMENT failed, cause by: Table %s in source has more columns than target, the columns is= [%s, %s].

Possible Causes

The possible causes are as follows:

- The filter conditions of DDL statements for adding columns in the source database are not synchronized to the destination database.
- As a result, columns are deleted in the destination database.

Solution

Supplement the missing columns in the destination database based on the table structure of the source database.

2.4.12 Incremental Synchronization Errors: Connection to *.*.*.98:8000 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Unable to connect to DBMS: url=jdbc:dws://*.*.*.115:8000,*.*.*.98:8000,*.*.*.133:8000/tanzhou_prod?client_encoding=UTF-8&loadBalanceHosts=true&targetServerType=any&reWriteBatchedInserts=true&socketTimeout=600&connectTimeout=300&binaryTransfer=true&ssl=false&sslmode=prefer user=dbadmin, Caused by: Connection to *.*.*.98:8000 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections

Possible Causes

The possible causes are as follows:

- The connection to the source or destination database fails to be established.

Solution

1. Check whether the source or destination database is running properly.
2. Check whether the network connection between the DRS replication instance and the source or destination database is normal.
3. Check whether the IP address of the DRS instance is allowed to access the source or destination database.

2.4.13 Incremental Synchronization Error: Table %s not found in target database

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: INCREMENT failed, cause by: Table %s not found in target database.

Possible Causes

The possible causes are as follows:

- The filter conditions of DDL statements for creating tables in the source database are not synchronized to the destination database.
- As a result, tables are deleted from the destination database.
- Some DDL statements (such as rename) are filtered out during online DDL.

Solution

For the first two causes, create a table in the destination database based on the table structure in the source database.

If some DDLs are filtered out, rename the temporary table to the correct table name in the destination database.

2.4.14 Incremental Synchronization Error: in a read-only transaction

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: insert %s failed:tid: %s, sqno: %s, ERROR: dn_ %s_ %s: cannot execute INSERT in a read-only transaction.

Possible Causes

The possible causes are as follows:

- The destination database is in read-only mode. The possible cause is that the disk space is full.

Solution

Contact GaussDB(DWS) O&M personnel.

2.4.15 Incremental Synchronization Error: relation '%s' does not exist

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: insert %s failed:tid: %s, sqno: %s, ERROR: relation ' %s' does not exist.

Possible Causes

The possible causes are as follows:

- The filter conditions of the DDL statement for creating tables in the source database are not executed in the destination database during synchronization. As a result, the tables do not exist in the destination database.
- The tables are deleted from the destination database.
- Filtering conditions are set for DDL statements during synchronization. Some DDL statements (such as rename) are filtered out. As a result, the destination database table does not exist.

Solution

For the first two causes, create a table in the destination database based on the table structure in the source database.

If some DDL statements are filtered out, rename the temporary table to the correct table name in the destination database.

2.4.16 Incremental Synchronization Error: %s doesn't in the target table

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: insert %s failed: %s doesn't in the target table: %s.

Possible Causes

The possible causes are as follows:

- The filter conditions of DDL statements for adding columns in the source database are not synchronized to the destination database.
- Columns are deleted in the destination database.

Solution

Supplement the missing columns in the destination database based on the table structure of the source database.

2.4.17 Incremental Synchronization Error: %s doesn't in the target table

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: insert %s failed: %s doesn't in the target table: %s.

Possible Causes

The possible causes are as follows:

- The filter conditions of DDL statements for adding columns in the source database are not synchronized to the destination database.
- Columns are deleted in the destination database.

Solution

Supplement the missing columns in the destination database based on the table structure of the source database.

2.4.18 Incremental Synchronization Error: syntax error at or near

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Check whether dws supports the DDL. For support, please contact dws Services. DDL: CREATE TABLE xxx like xxx. failed by:ERROR: syntax error at or near '%s'

Possible Causes

The possible causes are as follows:

- The DDL statement conversion in the source database does not comply with the syntax of the destination database.

Solution

DRS does not convert the syntax for incremental DDL synchronization from MySQL to GaussDB (DWS). The DDL executed in the MySQL database will be executed in GaussDB(DWS). Contact Huawei technical support.

2.4.19 Incremental Synchronization Error: schema '%s' does not exist

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: update %s failed: ERROR: schema ' %s' does not exist.

Possible Causes

The possible causes are as follows:

- The filter conditions of DDL statements for adding tables in the source database are not synchronized to the destination database.
- Schemas are deleted from the destination database.

Solution

Create schemas in the destination database based on the source database structure.

2.5 Real-Time Synchronization from PostgreSQL to PostgreSQL

2.5.1 Full Synchronization Error: function uuid_generate_v4() does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_structure, index= %s, schema_name= %s, object_name= %s] occur error, msg=ERROR: function uuid_generate_v4() does not exist Hint: No function matches the given name and argument types. You might need to add explicit type casts.

Possible Causes

- The destination database does not have the uuid-ossp extension, and uuid_generate_v4() is the function of this extension.

  NOTE

  Log in to the destination RDS PostgreSQL database and run the following SQL statement:

  ```
  select * from pg_extension where extname = 'uuid-ossp';
  ```

Solution

Extensions are not synchronized. Before synchronization, install the corresponding extension in the destination database. Perform the following steps to install the extension on the destination database and retry the DRS task:

Step 1 Log in to the destination RDS PostgreSQL database as the **root** user.

Step 2 Run the following SQL statement to install uuid-ossp:

```
create extension "uuid-ossp";
```

Step 3 Retry the DRS task.

----End

2.5.2 Full Synchronization Error: relation '%s' does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: ERROR: relation '%s' does not exist Position: 15

Possible Causes

- During table-level synchronization, objects with dependencies are not synchronized. For example, the source database contains tables A and B and table A depends on table B, but only table A is synchronized.

Solution

Step 1 Clear data in the destination database.

Step 2 Create a synchronization task again and select the objects to be synchronized and all dependent objects.

Step 3 Start the synchronization task.

----End

2.5.3 Full Synchronization Error: GC overhead limit exceeded

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: GC overhead limit exceeded.

Possible Causes

- Too many large objects exist in the service. As a result, the memory usage of the synchronization task exceeds the threshold.

Solution

Contact Huawei technical support.

2.5.4 Full Synchronization Error: Java heap space

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: Java heap space.

Possible Causes

- A large number of fields exist in the service. As a result, the memory usage of the synchronization task exceeds the threshold.

Solution

Contact Huawei technical support.

2.5.5 Full Synchronization Error: column '%s' of relation '%s' does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: apply event=[type=table_data, batch_index_in_shard= %s, table_schema= %s, table_name= %s, record_num= %s] occur error, msg=apply table %s data failed: %s: ERROR: column ' %s' of relation ' %s' does not exist Position: 1043 Call getNextException to see other errors in the batch.

Possible Causes

- During the full synchronization, DDL operations are executed in the destination database. As a result, the table structure in the destination database is inconsistent with that in the source database.
- During the full synchronization, DDL operations are executed in the source database. As a result, the table structure in the destination database is inconsistent with that in the source database.

Contact the customer to confirm whether they executed DDL operations.

Solution

Create a synchronization task again. During the full synchronization, ensure that no DDL operation is executed on the source database and no data is written to the destination database. Otherwise, data may be inconsistent or the synchronization may fail.

2.5.6 Full Synchronization Error: column '%s' does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=function, index=0, schema_name= %s, object_name=' %s']reason:[ERROR: column ' %s' does not exist Position: %s].

Possible Causes

- During the full synchronization, DDL operations are executed in the destination database. As a result, the table structure in the destination database is inconsistent with that in the source database.
- During the full synchronization, DDL operations are executed in the source database. As a result, the table structure in the destination database is inconsistent with that in the source database.

Contact the customer to confirm whether they executed DDL operations.

Solution

Create a synchronization task again. During the full synchronization, ensure that no DDL operation is executed on the source database and no data is written to the destination database. Otherwise, data may be inconsistent or the synchronization may fail.

2.5.7 Full Synchronization Error: type 'hstore' does not exist

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: retry structures failed events=the fail structures are [type=operator, index=2, schema_name=public, object_name=?|]reason:[ERROR: type 'hstore' does not exist].

Possible Causes

- The hstore plug-in is not installed on the destination database.

  NOTE

  Run the following SQL statement in the destination RDS PostgreSQL database:

  ```
  select * from pg_extension where extname = 'hstore';
  ```

Solution

Extensions are not synchronized. Before synchronization, install the corresponding extension in the destination database. Perform the following steps to install the extension and retry the DRS task.

Step 1 Log in to the destination RDS PostgreSQL database as the **root** user.

Step 2 Run the following SQL statements to install hstore:

```
create extension "hstore";
```

Step 3 Retry the DRS task.

----End

2.5.8 Full Synchronization Error: Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: source engine postgresql client initialize failed, detail: Unable to connect to DBMS: url= %s/position3.0?client_encoding=UTF-8&ssl=false&sslmode=prefer user= %s, Caused by: Connection to %s refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.

Possible Causes

- The connection to the source or destination database fails to be established.

Solution

Perform the following operations:

1. Check whether the source or destination database is running properly.
2. Check whether the DRS instance IP address is allowed by the listening port of the source or destination database.
   - For DRS tasks performed over a public network, the source database must allow access from the DRS instance EIP, and the destination database must allow access from the private IP address of the DRS instance.

- For DRS tasks performed in a VPC, VPN, or Direct Connect network, both the source and destination databases must allow access from the private IP addresses of DRS instance.

2.5.9 Full Synchronization Error: invalid locale name

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service DATAMOVE failed, cause by: target engine postgresql client initialize failed, detail: Unable to connect to DBMS: url= %s/position3.0?client_encoding=UTF-8&autosave=always&stringtype=unspecified&ssl=false&sslmode=prefer user= %s, Caused by: ERROR: invalid locale name:'Chinese (Simplified)_China.936'.

Possible Causes

- The source database region type is not supported by the destination database.

Solution

Contact the customer to check whether the region type can be changed to another one (UTF-8 by default). The region type may affect the sorting rules of different languages. If the encoding format can be changed to UTF-8, contact Huawei technical support.

2.5.10 Incremental Synchronization Error: Table %s not found in target database

Scenarios

During full synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Table %s not found in target database.

Possible Causes

- The user did not select to synchronize DDL, so the CREATE TABLE statement is not synchronized to the destination database.
- The user selected to synchronize DDL, but the source database uses the DDL statement that is not supported by DRS to create a table.
- The table is deleted from the destination database.

Solution

- Method 1: Create a task again and comply with the following DRS usage rules:
  - If you select to synchronize DDL, do not execute DDL statements that are not supported by DRS in the source database.
  - If you do not synchronize DDL, do not execute DDL statements in the source database, or execute DDL statements in the destination database before executing the same DDL statements in the source database.

- During full and incremental synchronization, do not write data to the destination database. Otherwise, data may be inconsistent or the synchronization may fail.

- Method 2: Perform the following operations to restore the DRS task:
  - **Possible cause:** The user did not select to synchronize DDL, so the CREATE TABLE statement in the source database is not synchronized to the destination database.

    Solution: Create a table in the destination database based on the table structure of the source database and retry the DRS task.
  - **Possible cause:** The user selected to synchronize DDL, but the source database uses the DDL statement that is not supported by DRS to create a table.

    Solution: Create a table in the destination database based on the table structure of the source database and retry the DRS task.
  - **Possible cause:** The table is deleted from the destination database.

    Solution: Re-create the table in the destination database based on the structure of the deleted table and retry the DRS task.

    NOTE

    If both the table and the data in the table are deleted, re-creating the table may lead to data inconsistency or cause the task to fail again.

2.5.11 Incremental Synchronization Error: remaining connection slots are reserved

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: Unable to connect to DBMS: url= %s user= %s, Caused by: FATAL: remaining connection slots are reserved for non-replication superuser connections.

Possible Causes

- The number of user connections to the destination database reaches the upper limit. As a result, the connection to the destination database fails to be established.

  NOTE

  Log in to the destination RDS PostgreSQL database and run the following SQL statement:

  - View max_connections.
    ```
    show max_connections;
    ```
  - Check the current number of connections.
    ```
    select count(*) from pg_stat_activity;
    ```

Solution

On the destination RDS PostgreSQL database console, change the value of **max_connections** to a larger value and make it take effect. Each DRS task requires about 100 connections.

2.5.12 Incremental Synchronization Error: PL/pgSQL function %s line %s at SQL statement

Scenarios

During incremental synchronization, an error is reported, and the log information is as follows: service INCREMENT failed, cause by: insert %s failed:tid: %s, sqno: %s, ERROR: %s' PL/pgSQL function %s line %s at SQL statement.

Possible Causes

- The destination database **session_replication_role** is not set to **replica**, and the destination database trigger is not disabled.

  NOTE

  You can log in to the destination database RDS PostgreSQL and run the following SQL statement to check the value of **session_replication_role**:

  ```
  show session_replication_role;
  ```

Solution

Step 1 On the RDS PostgreSQL console, change the value of **session_replication_role** to **replica** and apply the changes.

Step 2 Retry the DRS task.

----End

2.6 Backup Migration Failed Because Backup Files Cannot Be Found

Scenarios

When you migrate full backups from self-built OBS buckets to clouds, the following error message is displayed: restore:null.

Possible Causes

The possible causes are as follows:

- Backup files are deleted after you submit a backup migration task.
- When you upload backup files to a self-built OBS bucket, you select **Archive** for **Storage Class**. OBS archive storage offers cloud storage for rarely accessed data. An archive file uploaded for the first time is in the **Not restored** status. As a result, a Microsoft SQL Server DB instance cannot download the file.

Solutions

Based on the previous analysis, solutions are provided as follows:

Solution 1

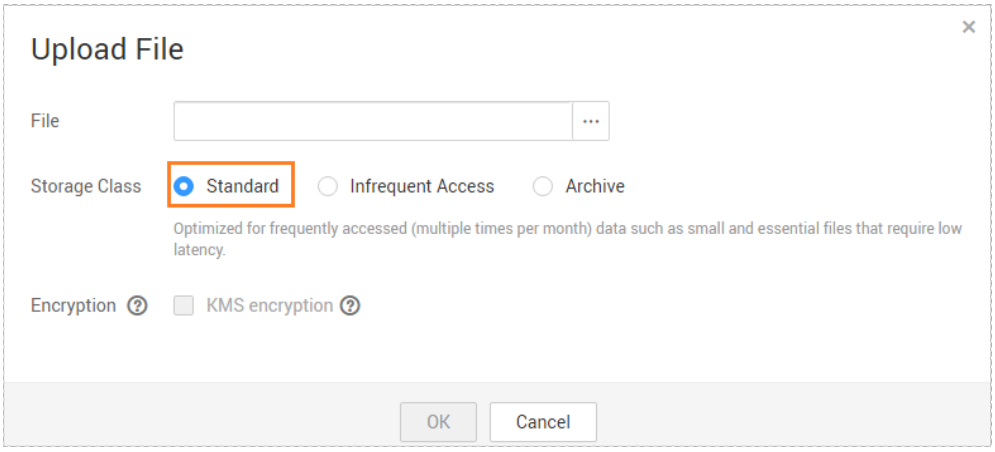
If the migration fails because you delete the backup files, you can upload the deleted backup files again to a self-built OBS bucket and select **Standard** for **Storage Class**. For details, see **Uploading a File** in *Object Storage Service Console Operation Guide*.

Solution 2

- If the migration failed because the storage class of your backup files is **Archive**, perform the following steps. If the size of backup files is small, upload the backup files again to an OBS bucket and select **Standard** for **Storage Class**.

  For details, see the "Uploading a File or Folder" section in the *Object Storage Service Console Operation Guide.*

  Figure 2-1 Uploading a file

  Upload File

  File

  Storage Class ● Standard ○ Infrequent Access ○ Archive

  Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

  Encryption □ KMS encryption

- If the backup files are large in size, log in to the OBS console and click the bucket to which the backup files are uploaded. On the displayed page, choose **Objects** in the navigation pane on the left. On the **Objects** page, select the object to be restored and click **Restore** above the file list. After the status of the backup files becomes **Restored**, submit an offline migration task again.

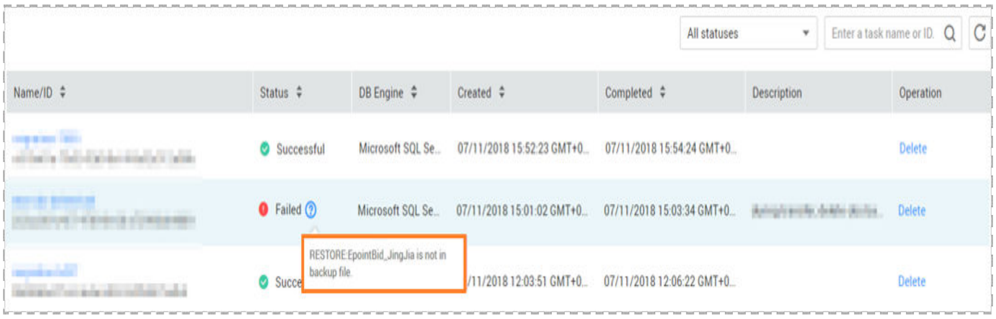
  For details, see the "Restoring an Archive File on OBS" section in the *Object Storage Service Console Operation Guide*.

2.7 Backup Migration Failed Because a Backup Database Cannot Be Found in the Backup Files

Scenarios

When you migrate full backups from self-built OBS buckets to clouds, the system displays an error message indicating that the migration failed because the source database cannot be found in the backup files.

Figure 2-2 Backup migration



Name/ID	Status	DB Engine	Created	Completed	Description	Operation
[redacted]	Successful	Microsoft SQL Se...	07/11/2018 15:52:23 GMT+0...	07/11/2018 15:54:24 GMT+0...		Delete
[redacted]	Failed	Microsoft SQL Se...	07/11/2018 15:01:02 GMT+0...	07/11/2018 15:03:34 GMT+0...	[redacted]	Delete
[redacted]	Succe		/11/2018 12:03:51 GMT+0...	07/11/2018 12:06:22 GMT+0...		Delete

RESTORE:EpointBid_JingJia is not in backup file.

Possible Cause

The name of a .bak backup file uploaded to a self-built OBS bucket is too long.

Solution

Based on the previous analysis, a solution is provided as follows:

Step 1 Check the name of the backup file to be uploaded to an OBS bucket by referring to **Preparing Backup Files** in the *Backup Migration*.

Step 2 Change the name of the backup file in the local database and upload the file to a self-built OBS bucket again.

----End