

Migrate the MySQL database created on Alibaba Cloud ECS to ApsaraDB for RDS

Alibaba Cloud Professional Service Team
July 2017

Contents

Introduction	3
Restrictions	3
Environment preparation	4
Migration procedure	7
Further reading	13

Introduction

You can use the Data Transmission Service (DTS) of Alibaba Cloud to implement the data migration from a MySQL database created on Alibaba Cloud ECS to a MySQL instance of ApsaraDB for RDS. DTS provides the synchronization of incremental data, which enables the seamless migration of MySQL data without terminating the local application services.

This article introduces the solution of MySQL database migration from Alibaba Cloud ECS to ApsaraDB for RDS, including the migration of data, views, stored procedures, user-defined functions and other objects.

This solution implements full migration and incremental migration of MySQL database through DTS. From this article, you can learn how to use DTS to implement the full migration and incremental migration of MySQL databases without terminating the application services.

Restrictions

- For each migration type, the restrictions are described as follows.
 - Structure migration

Structure migration migrates the structure definition of the migrated object to the destination instance. Currently, the objects supported by DTS for structure migration include tables, views, triggers, stored procedures, and storage functions.

Structure migration does not support event migration.

- Full migration

Full migration migrates all data stored in the MySQL database on ECS to the destination RDS instance.

If you only perform full migration, the newly added business writes to the local MySQL instance during the migration will not be synchronized to the destination RDS instance.

If you select incremental migration additionally, to ensure data consistency, non-transaction tables without a primary key will be locked during the full migration process. These tables cannot be written during lock-up and the locking duration depends on the data volume of these tables. The locks will be released only after the non-transaction tables without a primary key are successfully migrated.

- Incremental migration

Incremental migration synchronizes the incremental data of the local MySQL database during the migration to the destination RDS instance, which enables the real-time synchronization of incremental data so that the dynamic data synchronization process is implemented during the data migration from the local MySQL database to ApsaraDB for RDS.

When incremental migration is selected, binlog must be enabled for the source MySQL instance and the source database must be set to "binlog_format =row".

When incremental migration is selected and the source MySQL version is 5.6 or later, the parameter binlog_row_image of the source database must be set to "full".

- DDL operations are not supported during migration.
- If object name mapping is enabled, other objects dependent on this object may fail to be migrated.

Environment preparation

- For the destination database on RDS
 - a) Create a MySQL 5.6 instance on RDS.
 - b) Create and configure the database and account used for migration in the MySQL 5.6 instance of RDS.
- For the source database on ECS
 - a) Create an ECS instance with the operating system of Linux x86 and an Internet IP address.
 - b) Install MySQL 5.6 on ECS.
 - c) Create MySQL database and configure the account used for migration on ECS.
- To perform incremental migration, check whether the binlog of the source database is enabled and whether the format meets the requirements. The items to be checked are as follows:

- a) Execute the following command to check whether binlog is enabled for the source database.

```
show global variables like 'log_bin' ;
```

If the binlog is enabled, "log_bin=ON" and the returned result is shown as the following figure.

```
mysql> show global variables like 'log_bin';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin      | ON   |
+-----+-----+
```

If "log_bin=OFF" , it indicates that binlog is disabled for the source database. If you need to perform incremental migration, modify the binlog configuration of local MySQL database as follows to enable binlog. After the modification is complete, restart the MySQL process.

- ◆ Set " log_bin=mysql_bin" to enable binlog.
- ◆ Set "binlog_format=row" to set the binlog format to row.
- ◆ Set the value of server_id to an integer greater than 1.
- ◆ If the MySQL version is 5.6, set "binlog_row_image=full" .

- b) Execute the following command to check whether the binlog format of the source database is row.

```
show global variables like 'binlog_format' ;
```

If the binlog format is row, "binlog_format=ROW" and the returned result is shown as the following figure.

```
mysql> show global variables like 'binlog_format';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
1 row in set (0.00 sec)
```

If the returned value of `binlog_format` is not `ROW`, execute the following command to change the binlog format to `row`.

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

When the modification is complete, we recommend that you kill the old connections and reconnect to MySQL. Otherwise, some other connections may still use the statement format to record binlog.

- c) If the local MySQL database version is 5.6.2 or later, check whether the `binlog_row_image` value of the source database is "full". The detailed procedure is as follows.
 - i. Execute the following command to query the local MySQL database version.

```
select version ();
```

The returned result is shown as the following figure.

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.6.36-log |
+-----+
1 row in set (0.00 sec)
```

- ii. If the database version is 5.6.2 or later, execute the following command to query the `binlog_row_image` format of the local MySQL database.

```
show global variables like 'binlog_row_image' ;
```

If "`binlog_row_image=FULL`" , the returned result is shown as the following figure.

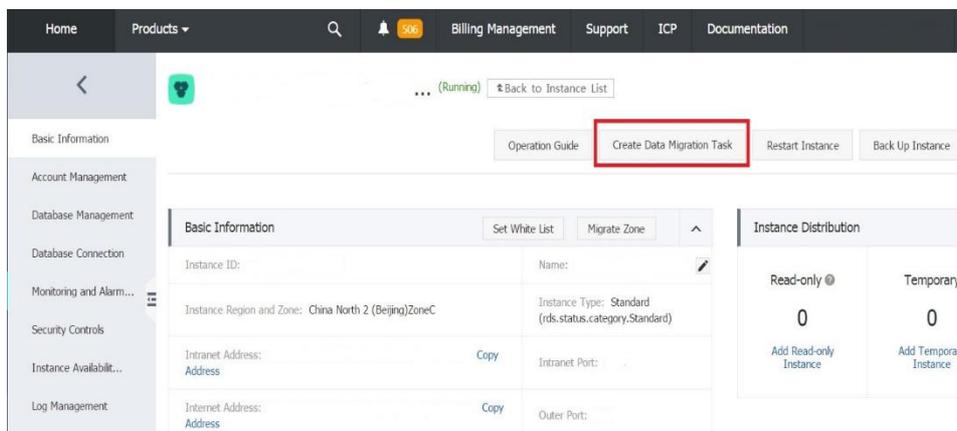
```
mysql> show global variables like 'binlog_row_image';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_row_image | FULL |
+-----+-----+
1 row in set (0.00 sec)
```

If the query result shows the value of `binlog_row_image` is not "FULL", execute the following command to change it to "FULL".

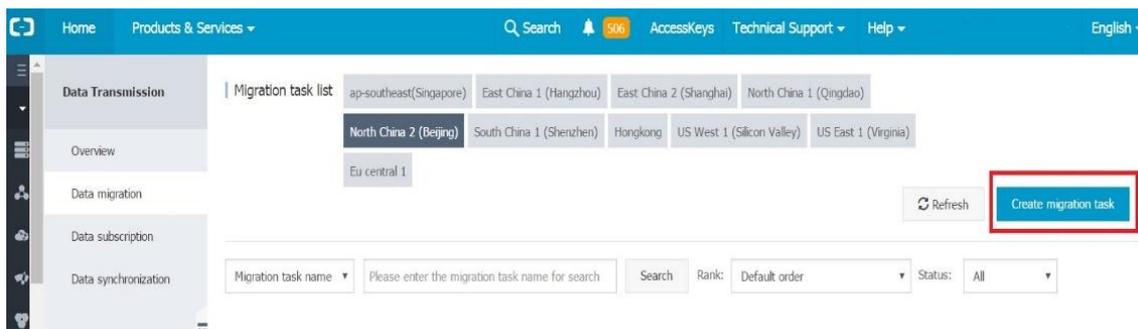
```
set global binlog_row_image=full;
```

Migration procedure

1. Log on to the [RDS console](#).
2. Select the region where the target instance is located.
3. Click the instance ID to enter the **Basic Information** page.
4. Click Create **Data Migration Task** to log on to the Data Transmission console, as shown in the following figure.



5. In the left-side navigation pane, select **Data migration** to enter the **Migration task list** page.
6. Click **Create migration task** to enter the Data Transmission task configuration page, as shown in the following figure.



7. Enter the information of the source and target databases, as shown in the following figure.

The screenshot displays a configuration interface for migrating a MySQL database from ECS to RDS. It is divided into two main sections: 'Source database' and 'Target database'. Each section contains several input fields and dropdown menus, with a 'Test the connection' button at the bottom right of each section. A red box highlights the 'Instance type', 'Instance region', and 'ECS instance ID' fields in the source database section, and the 'Instance type', 'Instance region', and 'RDS instance ID' fields in the target database section.

Source database configuration:

- * Task name: [Input field]
- * Instance type: ECS-based databases
- * Instance region: ap-southeast(Singapore)
- * ECS instance ID: [Input field]
- * Database engine: MySQL
- * Port: 3306
- * Database account: [Input field]
- * Database password: [Input field]
- Test the connection

Target database configuration:

- * Instance type: RDS instance
- * Instance region: ap-southeast(Singapore)
- * RDS instance ID: Select DRDS instance
- * Database account: [Input field]
- * Database password: [Input field]
- Test the connection

Parameters description:

- Task name: By default, Data Transmission generates a name for every task automatically. The task name is not required to be unique. You can modify the name. A name indicating the specific services of the task is recommended to facilitate task identification.
- Source database
 - ◆ Instance type: Select **ECS-based databases**.
 - ◆ Instance region: Select the region where the ECS instance is located.
 - ◆ ECS instance ID: Select the ECS instance ID.
 - ◆ Database engine: Select **MySQL**.
 - ◆ Port: The listener port of the MySQL instance.
 - ◆ Database account: The account used to access the MySQL instance.
 - ◆ Database password: The password corresponding to the account used to access the MySQL instance.

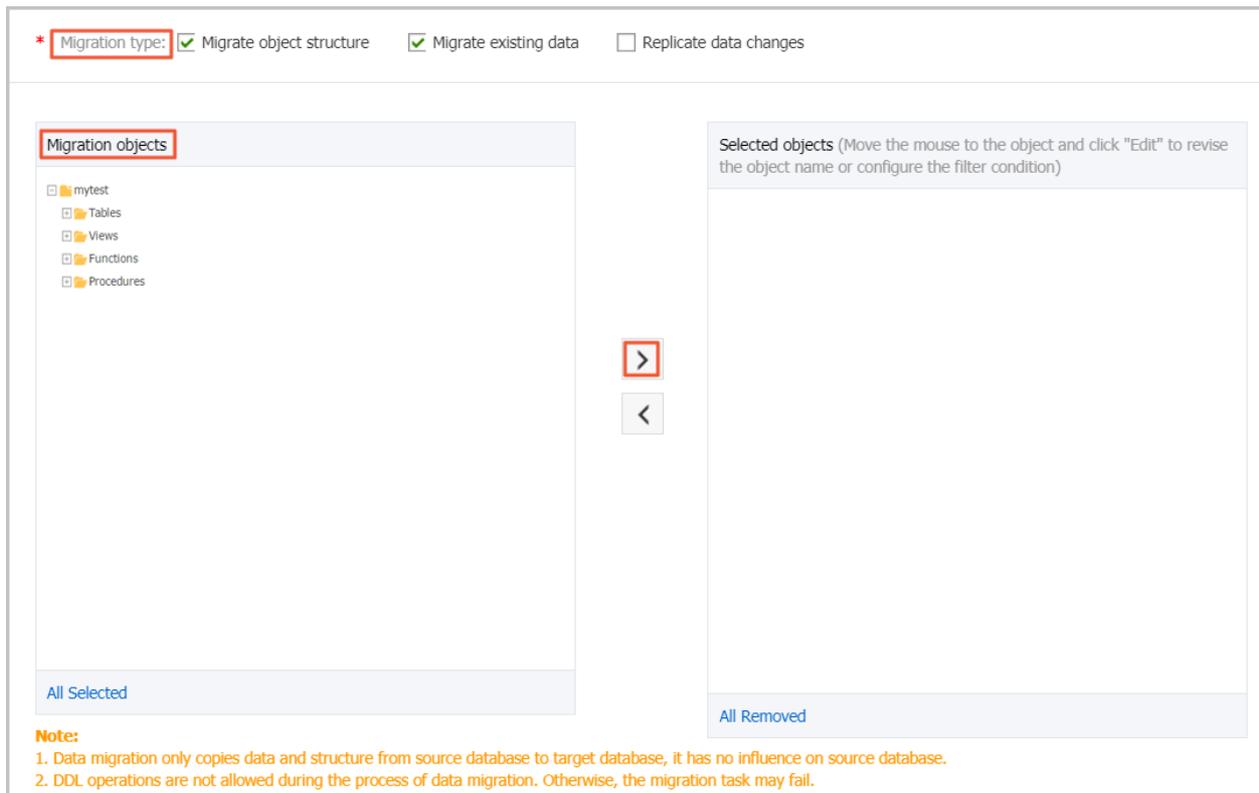
- Target database

- ◆ Instance type: Select **RDS instance**.
- ◆ Instance region: Select the region where the RDS instance is located.
- ◆ RDS instance ID: Select the ID of the target RDS instance to be migrated.
- ◆ Database account: The account used to access the RDS instance.
- ◆ Database password: The password corresponding to the account used to access the RDS instance.

8. Click **Test the connection** in the Source database and Target database areas respectively to test whether the connection information is correct.
9. If **Test passed** appears, click **Authorize whitelist and enter into next step** to enter **the Migration class and list** page.

Note: If the connection test fails, view the diagnosis and rectify the faults accordingly first.

10. Select the migration type and migration objects, as shown in the following figure.



Parameters description:

- Migration type

- ◆ DTS supports structure migration, full migration and incremental migration.
- ◆ If you need migration without stopping services, select all the three migration types.

- ◆ If you only need full migration, select **Migrate object structure** and **Migrate existing data**.

- Migration objects

- ◆ The migration object can be a database, a table or a column.
- ◆ By default, after the object is migrated to an RDS instance, the object name remains the same with that in the local MySQL instance. If the object you migrate has different names on the source and destination instances, you need to use the object name mapping feature provided by DTS. Detailed usage can be found in [Database Table Column Mapping](#).

11. Click **Pre-check and start**.

Note: A pre-check will be performed before a migration task is formally started. Migration can be started only after the pre-check is successful. If the pre-check fails, check the failure details by clicking the button after the specific check items, rectify the faults accordingly, and perform a pre-check again.

12. If the pre-check is successful, click **Buy and start now** to start the migration task.

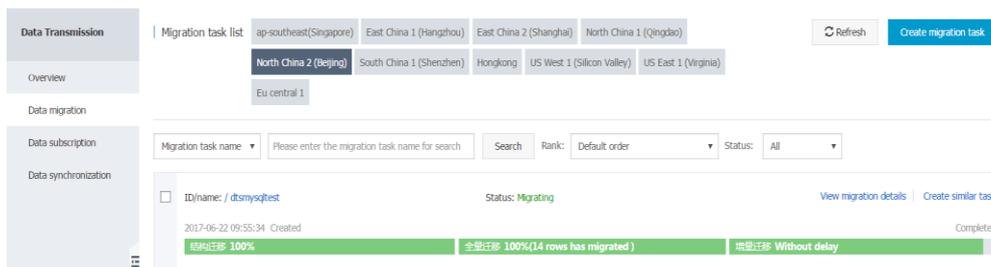
13. Go back to the Data Transmission console.

14. In the left-side navigation pane, select **Data migration** to enter the **Migration task list** page.

15. Select the region where the target RDS instance is located.

16. Find the migration task and view the progress, as shown in the following figure. If you want to view the migration details, click **View migration details**.

Note: Incremental migration is a process of dynamic synchronization. We recommend that you verify the business on the destination database when the incremental migration achieves no latency. If the verification succeeds, you can stop the migration task and switch the business to the destination database.



17. Verify whether the changes to the source database can be synchronized to the destination database during the migration process. The detailed procedure is as follows.

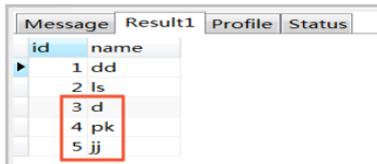
- a) Execute the following command to insert three records to the source database.

```
insert into t2 values (3, 'd' ),(4,' pk' ),(5,' jj' );
```

- b) Execute the following command to query whether the new records are added to the source database.

```
select * from t2;
```

If the following result is returned, it indicates that the three records are added to Table t2 of the source database successfully.

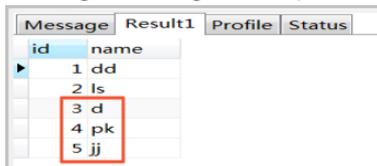


id	name
1	dd
2	ls
3	d
4	pk
5	jj

- c) Execute the following command to query the new records in the destination database.

```
select * from t2;
```

If the following result is returned, it indicates that the records added to the source database during the migration process are synchronized to the destination database in real time.

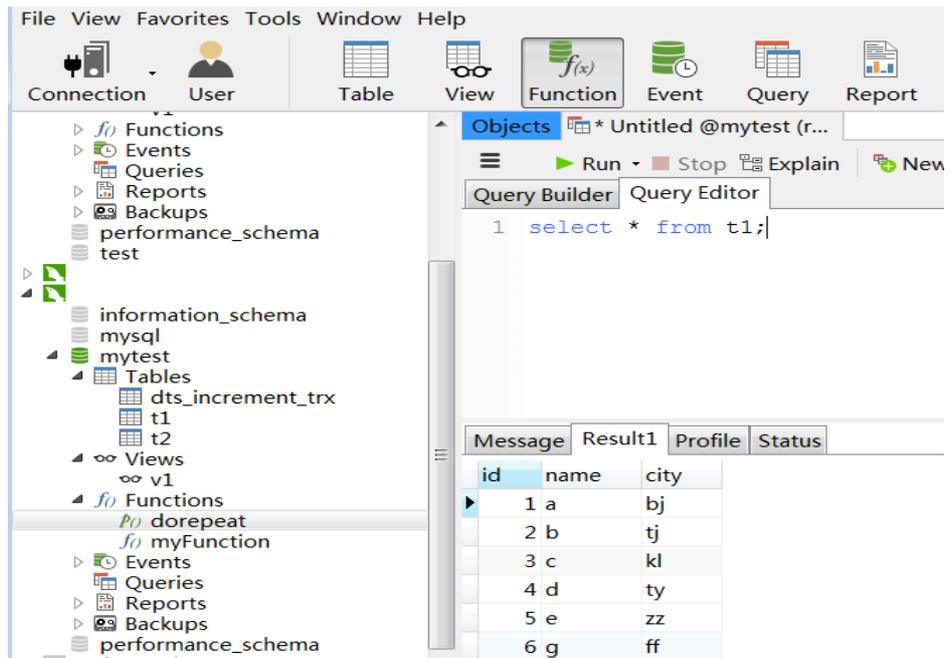


id	name
1	dd
2	ls
3	d
4	pk
5	jj

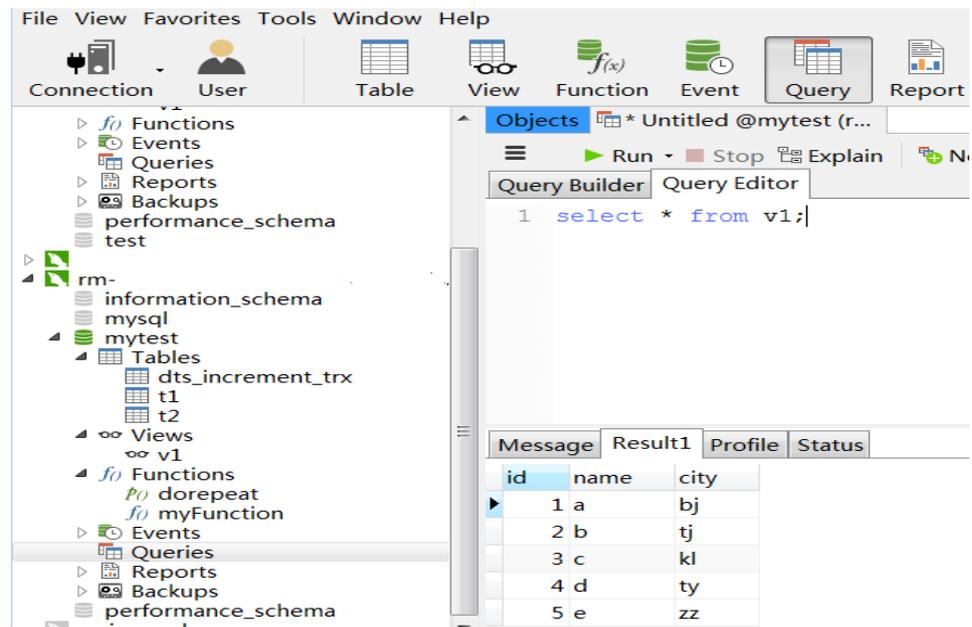
Data validation after migration

- Table data validation

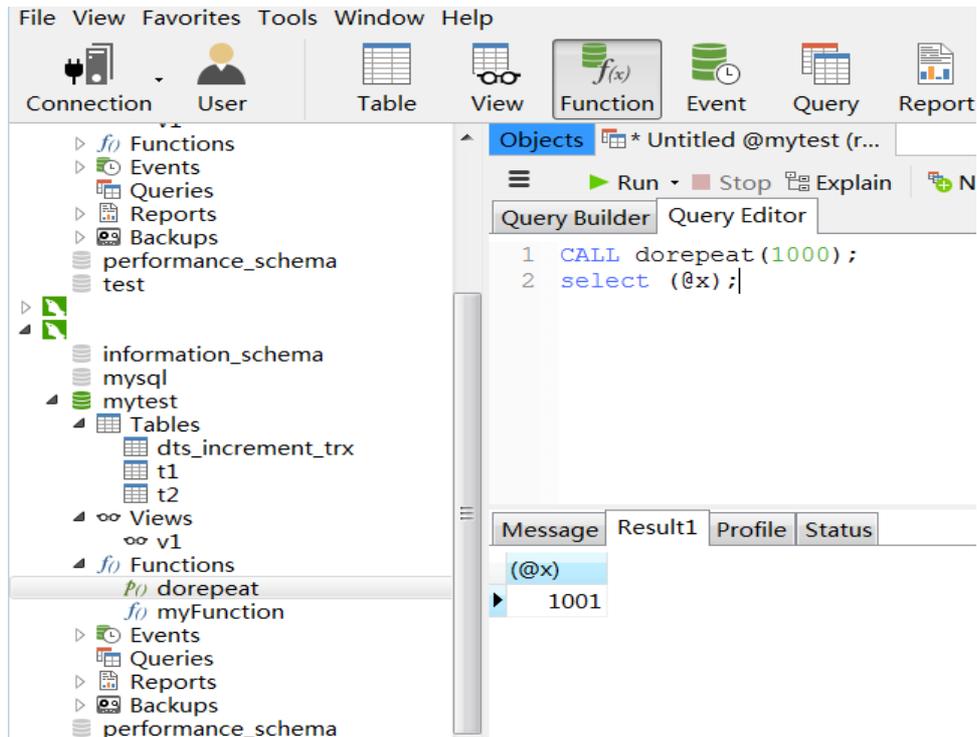
Alibaba Cloud - Migrate the MySQL database created on Alibaba Cloud ECS to ApsaraDB for RDS



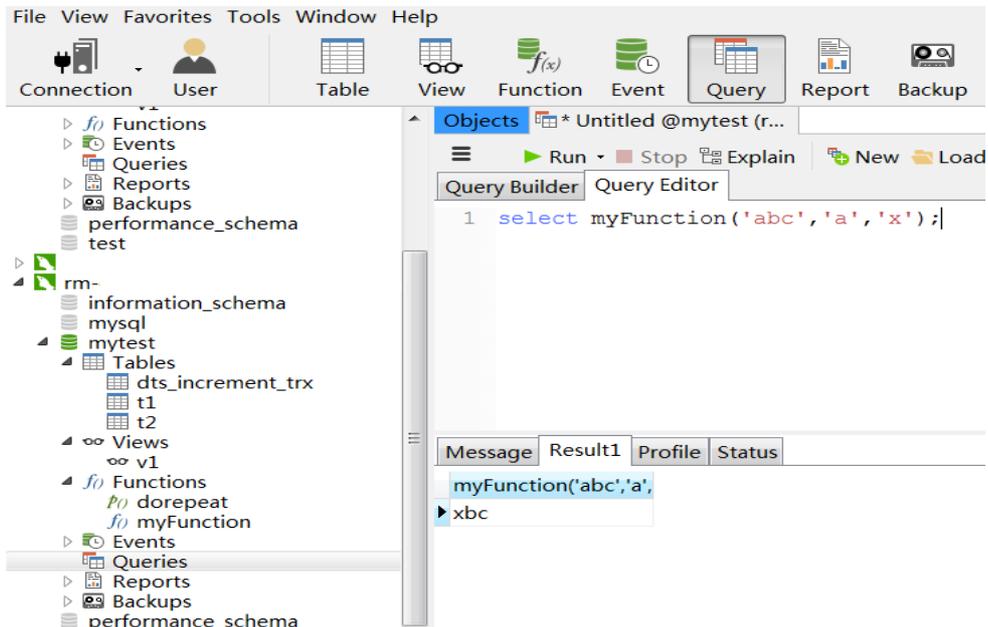
- Views



- Stored procedures



- User-defined functions



Further reading

Migrate Azure Database for MySQL to MySQL of ApsaraDB for RDS