Distributed Relational Database Service

Quick Start

Quick Start

Basic SQL operations

View databases

Use the SHOW DATABASES statement to display databases.

```
show databases;
```

Create, view, and delete tables

Use the CREATE TABLE statement to create tables.

```
-- Table sharding
CREATE TABLE multi_db_single_tbl(
id int auto_increment,
name varchar(30),
primary key(id)
) dbpartition by hash(id);

-- Single table
CREATE TABLE single_tbl(
id int,
name varchar(30),
primary key(id)
);
```

Use the SHOW CREATE statement to view the created tables.

```
show create table multi_db_single_tbl;

show create table single_tbl;
```

Use the DROP TABLE statement to delete tables.

```
drop table multi_db_single_tbl;
```

```
drop table single_tbl;
```

Use the SHOW TABLES statement to view all tables in the database.

```
show tables;
```

Add columns, delete columns, and change data types

Use the ALTER TABLE statement to add, modify, and delete columns.

```
alter table multi_db_single_tbl add column textcol text;
alter table multi_db_single_tbl modify column textcol varchar(40);
alter table multi_db_single_tbl drop column textcol;
```

Create, view, and delete local indexes

Create local indexes.

```
create index idx_name on multi_db_single_tbl (name);
```

or

```
alter table multi_db_single_tbl add index idx_name(name);
```

View local indexes.

```
show index from multi_db_single_tbl;
```

Delete local indexes.

```
drop index idx_name on multi_db_single_tbl;
```

or

```
alter table multi_db_single_tbl drop index idx_name;
```

Add, delete, modify, and query data

Use the INSERT statement to insert data into a table.

```

```

```
insert into multi_db_single_tbl (name) values ('test_name');
insert into multi_db_single_tbl (name) values ('test_name');
insert into multi_db_single_tbl (name) values ('test_name'),('test_namexx');
```

Use the SELECT statement to retrieve data from a table.

```
select * from multi_db_single_tbl;
```

Use the UPDATE statement to modify data in a table.

```
update multi_db_single_tbl set name='zzz' where id in (100001,100002,100003,100004);
```

Use the DELETE statement to delete data from a table.

```
delete from multi_db_single_tbl where id = 100002;
```

Create, authorize, and delete users

Use the CREATE USER statement to create a user named drdsuser with the password 123456.

```
create user drdsuser@'%' identified by '123456';
```

Authorize the drdsuser user to retrieve tables from the sample_db database.

```
grant insert,update,delete,select on sample_db.* to drdsuser@'%';
```

Query the permissions of the drdsuser user.

```
show grants for drdsuser@'%'
```

Delete the drdsuser user.

```
drop user drdsuser@'%'
```

Buy a DRDS instance

This topic describes how to buy and create a DRDS instance.

Procedure

You can enter the DRDS instance purchase portal via any one of the following two methods:

- **Method 1**

i. Log on the Alibaba Cloud homepage.

ii. Move the mouse to **Products**>**Database**>**Relational Database** in sequence, click on DRDS(the Distributed Relational Database Service), to enter the DRDS product home page.

iii. On the DRDS product home page, click **Buy Now** to enter the DRDS instance purchase page, and follow the prompts to purchase the DRDS instance.

- **Method 2**

i. Log in to the DRDS console.

ii. At the upper right corner of the console page, click **Create Instance** to enter the DRDS instance purchase page.

On the purchase page, select the payment method and corresponding instance specifications as needed, and click **Buy Now**.

After confirming the order and checking Distributed Relational Database Service (DRDS) Monthly Subscription Agreement of Service, click **Pay**. After the payment is completed, you need to wait 1-5 minutes to start the service.

After the instance is created, you can return back to the **Intsances** page of the DRDS console to view the purchased DRDS instance. If you need to create a database for an instance, see Creating a DRDS database.

Create a DRDS database

The database we created on DRDS should be built on the RDS instance. For the stability of the OLTP service, we recommend that you select a new RDS instance for the creation of the DRDS database.

Limits

You can only create DRDS databases on the console, DRDS does not support creating databases with SQL commands.

Procedure

Log in to the DRDS console.

Click **Instance List** in the left navigation pane.

Find the target instance, and click the instance name to enter its **Basic Information** page.

At the upper right corner of the **Basic Information** page, Click **Create Database**.

In the dialog box that pops up, select **Partition Mode** according to your needs and enter the basic database information.

- **Horizontal Partitioning**:DRDS can split data into multiple databases and tables according to the sharding rules. With the help of the horizontal partitioning, you can expand the database horizontally to linearly increase the overall storage capacity and concurrent throughput of the database.
  If you choose **Horizontal Partitioning** mode:
  a. After filling in the basic information of the database, click **Next** and select the RDS instance to create the DRDS database.
     - The RDS instance that can be selected as a DRDS data storage node needs to meet all of the following three conditions.
       - The engine type of the target instance is MySQL.
       - The instance is in the running state.
       - The RDS instance is in the same region as the DRDS instance.
     - If the selected RDS instance has a **Privileged Account**, you will be asked to enter the privileged account and password, which will not be accessed or stored by DRDS and will only be used temporarily in subsequent steps..

  b. Click **Next** to preview the information of the library creation.

     By default, DRDS creates 8 physical libraries on 1 RDS instance, so the total number of physical sub-libraries is 8 times the number of selected RDS instances.

Vertical Partitioning: The existing RDS instances are transferred to the DRDS databases for proxied access to implement read and write splitting. You only need to change the database connection string, username, and password, without

importing data or modifying code.
You can use the DRDS instance-level account for unified authorization management. You can complete joint queries and write operations of transactions across multiple RDS instance databases through one DRDS endpoint.

> You need to upgrade the DRDS version to 5.3.8 or above to use this function.

If you choose **Vertical Partitioning** mode:

Select the databases of different RDS instances to complete the batch addition, and set the corresponding database parameters.

- When selecting an existing account, you can only select the account created on the DRDS console, and it is not supported to add an account created by the create user or grant.
- The same library name cannot be among multiple added databases.

Click **Next** to preview the parameters of creating a database.

After previewing and confirming the database parameters, click **Next** to complete the database creation.

After the creation, on the **Database Management** page, you can view all of the created database. You can also see the new account in **Account Management** page.

> The DRDS instance creates the database, account, and DRDS system tables through RDS APIs or the privileged account. Please wait until the database creation is complete with patience.

Create a DRDS table

After a DRDS database is created, you need to create tables. This topic describes how to create a DRDS table.

Procedure

Log in to DRDS Console.

In the left navigation pane, click **Instances**.

Find the target instance on the instance list page, click the instance name to enter the instance **Basic Information** page.

In the left navigation pane, click **Database Management**.

On the database list page, find the target database and click the database ID to enter the database **Basic Information** page.

In the **VPC Endpoint** area, find the **Command Line URL** which contains the DRDS connection string.

With the above DRDS connection string, you can use the following MySQL command to establish a connection to the DRDS database.

```
mysql -h${DRDS_ENDPOINT} -P${DRDS_PORT} -u${user} -p${password} -D${DRDS_DBNAME}
```

- If the DRDS instance is an exclusive instance, only the intranet address is provided by default. We recommend that you install the MySQL command line on an ECS instance that is in the same region as the DRDS instance.
- If the DRDS instance is a shared instance, you can connect to the DRDS database by using its Internet address from an Internet-based computer.

Run the DRDS DDL statement to create a table.

```
//DRDS DDL
CREATE TABLE shard_table(
id int,
name varchar(30),
primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(id) tbpartition by hash(id) tbpartitions 3;
```

Connect to DRDS

This topic describes how to connect to DRDS.

Procedure

Log in to DRDS Console.

In the left navigation pane, click **Instances**.

Find the target instance on the instance list page, click the instance name to enter the instance **Basic Information** page.

In the left navigation pane, click **Database Management**.

On the database list page, find the target database and click the database ID to enter the database **Basic Information** page.

In the **VPC Endpoint** area, find the **Command Line URL** which contains the DRDS connection string.

With the above DRDS connection string, you connect to the DRDS instance through one of the following methods.

- **MySQL command line connection** : If you have installed MySQL on your cloud server, you can run the following MySQL command to connect to the DRDS instance.

```
//Run the connection command
mysql -h${DRDS_ENDPOINT} -P${DRDS_PORT} -u${user} -p${password} -D${DRDS_DBNAME}
```

> DRDS supports MySQL official command-line 5.X series clients but not 8.0 clients at present. If you encounter a password error and other prompts, please try to install and use 5.X version clients.

Third-party tools : You can connect to DRDS through the following clients. Also,

you can go to the corresponding official websites to download and use the clients.

- MySQL Workbench (Recommended)
- SQLyog
- Sequel Pro
- Navicat for MySQL

On the third-party GUI client, you can perform basic database operations which includs data addition, deletion, modification, and DDL operations. DRDS may not support the advanced features of the tool.

Program code : You can connect to the DRDS through the following third-party program code that conforms to the official MySQL interactive protocol.

- JDBC Driver for MySQL (Connector / J)
- Python Driver for MySQL (Connector / Python)
- C ++ Driver for MySQL (Connector / C ++)
- C Driver for MySQL (Connector / C)
- ADO.NET Driver for MySQL (Connector / NET)
- ODBC Driver for MySQL (Connector / ODBC)
- PHP Drivers for MySQL (mysqli, ext / mysqli, PDO_MYSQL, PHP_MYSQLND)
- Perl Driver for MySQL (DBD :: mysql)
- Ruby Driver for MySQL (ruby-mysql)

DRDS connection examples

JDBC Driver for MySQL (Connector / J) program code example

```
//JDBC
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://drdsxxxxx.drds.aliyuncs.com:3306/doc_test","doc_test","doc_
test_password");
//...
conn.close();
```

Connection string configuration example

We recommend that you use the Druid connection pool to connect to DRDS. For more information on Druid, see Druid Github.

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-
method="close">
<property name="url" value="jdbc:mysql://drdsxxxxx.drds.aliyuncs.com:3306/doc_test" />
<property name="username" value="doc_test" />
<property name="password" value="doc_test_password" />

<property name="filters" value="stat" />

<property name="maxActive" value="100" />
<property name="initialSize" value="20" />
<property name="maxWait" value="60000" />
<property name="minIdle" value="1" />

<property name="timeBetweenEvictionRunsMillis" value="60000" />
<property name="minEvictableIdleTimeMillis" value="300000" />

<property name="testWhileIdle" value="true" />
<property name="testOnBorrow" value="false" />
<property name="testOnReturn" value="false" />

<property name="poolPreparedStatements" value="true" />
<property name="maxOpenPreparedStatements" value="20" />

<property name="asyncInit" value="true" />
</bean>
```