

TDSQL for MySQL

Getting Started

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Getting Started

InnoDB Engine

Creating Instance

Connecting to Instance

Managing Sharded Table

Getting Started

InnoDB Engine

Creating Instance

Last updated : 2024-01-06 17:33:30

This document describes how to create a TDSQL for MySQL instance in the console.

Prerequisites

You have registered a Tencent Cloud account and completed identity verification.

To register a Tencent Cloud account:

To complete identity verification:

Directions

1. Log in to the [TDSQL for MySQL purchase page](#), configure the following instance information, and click **Buy Now**.

Billing Mode: Pay-as-you-go billing is supported.

If your business has a stable long-term demand, we recommend you select monthly subscription.

If the request volume of your business fluctuates greatly and instantaneously, we recommend you choose pay-as-you-go billing.

Region: Select the region where you want to deploy your TDSQL for MySQL instance. We recommend you use the same region as the CVM instance to be connected to. Tencent Cloud services in different regions cannot communicate with each other over the private network. The region cannot be modified after purchase.

Network: Select the network where the TDSQL for MySQL instance resides. We recommend you select the same VPC in the same region as the CVM instance to be connected to. Otherwise, the MySQL instance cannot connect to the CVM instance over the private network.

Instance Version: For more information, see [Instance Architecture](#).

Source AZ and Replica AZ: Select different source and replica AZs to protect your database from failures and AZ outages.

Kernel Version: The MySQL 5.6 kernel does not support distributed transactions (XA). If you need this feature, select v5.7. For more information, see [Overview](#).

Shard Quantity/Specification/Disk: Two, four, or eight shards are recommended to avoid data skew. For more information on shard configuration, see [Selection of TDSQL Instance and Shard Configuration](#).

Project: Select a project to which the instance belongs. The default project is used.

Tag: Categorize and manage resources with tags. For more information, see [Tag Overview](#).

Security Group: For more information on security group creation and management, see [Security Group Configuration](#).

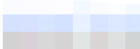

Instance Name: Name the instance now or later.

Supported character sets: UTF8, LATIN1, GBK, UTF8MB4, and GB18030 are supported.

Table Name Case Sensitivity: Select **Case sensitive (lower_case_table_names = 0)** or **Case insensitive (lower_case_table_names = 1)**. It is an initialization parameter and cannot be modified after the database is initialized.

Enable strong sync: Strong sync (downgradable) and async are supported. For more information, see [Strong Sync](#). For more information on billing, see [Pricing](#).

2. After making the payment, return to the instance list. After the status of the instance changes to **Running**, it can be used normally.

<div>Create Restart More</div> <div>Separate keywords</div>								
<input type="checkbox"/>	Instance ID/Name	Monitoring/St...	AZ	Instance ...	Project	Shard Qu...	Configuration	Database Ver...
<input type="checkbox"/>		 Running	Guangzhou Zone 7	Source Instance	DEFAULT PROJECT	2	Standard Edition (1 source, 1 replica) High IO Edition - 8-core, 16 GB memory/20 GB storage space luke-test - subnet-cp1qxb02	MySQL 8.0

Subsequent operations

For more information on how to connect to a TDSQL for MySQL instance at the private or public network address, see [Connecting to Instances](#).

Connecting to Instance

Last updated : 2024-01-06 17:33:30

Connection Methods

You can connect to a TDSQL for MySQL instance in the following ways:

Private network connection: A CVM instance can be used to connect to the private network address of a TDSQL for MySQL instance. This method utilizes the high-speed private network of Tencent Cloud and features low delay. The CVM and TencentDB instances must be under the same account and in the same [VPC](#) in the same region, or both in the classic network.

The private network address is provided by TencentDB by default and can be viewed in the instance list or on the instance details page in the [TDSQL for MySQL console](#).

Note:

CVM and TencentDB instances in different VPCs (under the same or different accounts in the same or different regions) can be interconnected over the private network through [Cloud Connect Network](#).

Public network connection: If you fail to access the private network, you can connect to your TDSQL for MySQL instance at its public network address. The public network address needs to be [manually enabled](#). It can be viewed on the instance details page in the [TDSQL for MySQL console](#) and can be disabled if no longer needed.

Enabling the public network address will expose your database services to the public network, which may lead to database intrusions or attacks. We recommend that you use the private network to connect to the database.

Public network connection to TencentDB is suitable for development or auxiliary management of databases but not for business access in the production environment, as potentially uncontrollable factors may lead to unavailability of the public network connection, such as DDoS attacks and bursts of high-traffic access.

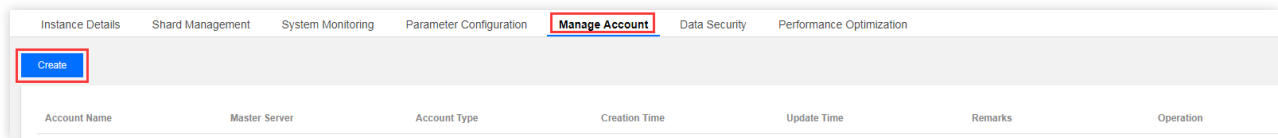
Currently, public network access can be enabled in Guangzhou, Shanghai, Beijing, Chengdu and Nanjing.

To enable public network access, associate the instance with a security group. For more information, see [Security Group Configuration](#).

Preparations

Creating an account

1. Log in to the [TDSQL for MySQL console](#). In the instance list, click an instance ID or **Manage** in the **Operation** column to enter the instance management page.
2. On the instance management page, select **Account Management** and click **Create Account**.

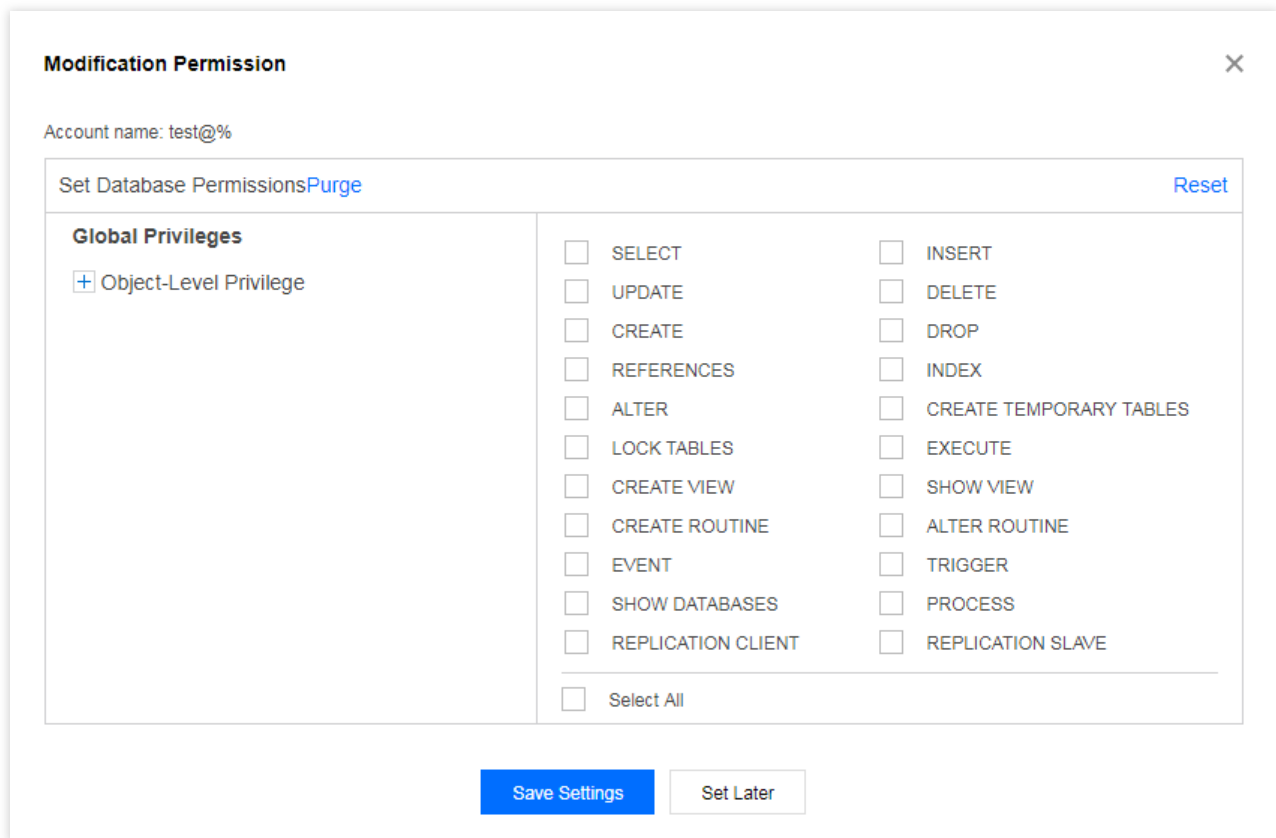


3. In the pop-up dialog box, enter the account name, host, and password. After confirming that everything is correct, click **Next**.

The host name is actually the network egress address. If needed, you can enter % to indicate that this instance is accessible to every IP.

4. In the **Modify Permissions** pop-up window, grant permissions as needed and click **Modify**. To discard the changes, click **Cancel Modification**.

The navigation bar on the left provides a graphical interface fully compatible with MySQL management. Permissions can be managed at the column level.

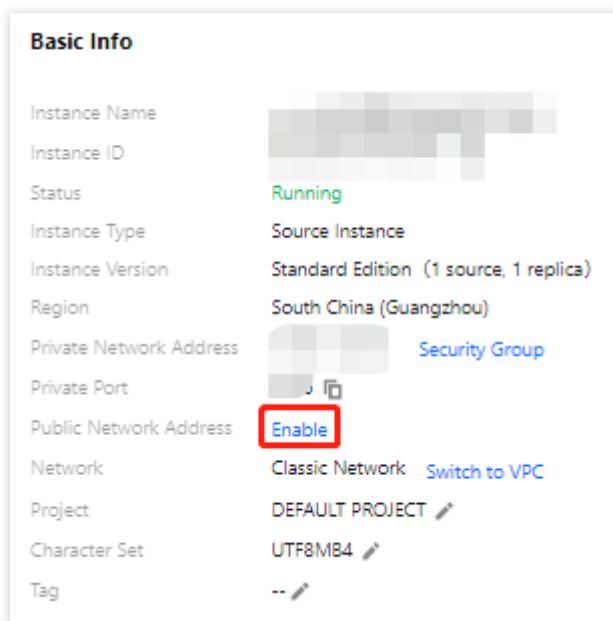


5. Return to the account list, click **Modify Permissions** to modify user permission, click **Clone Account** to completely copy the current account permissions to create a new account. Click **More** to reset the password and delete the account.

帐号名	主机	帐号类型	创建时间	更新时间	备注	操作
hao	%	常规帐号	2020-06-24 16:57:54	2020-06-24 16:57:54		修改权限 克隆帐号 更多 ▼

Enabling the public network address (optional)

1. Log in to the [TDSQL for MySQL console](#), click an instance ID in the instance list to enter the instance details page, and click **Enable** next to **Public IP** in the **Basic Info** section.



2. After this, the public network address and port number can be obtained. TDSQL for MySQL provides a unique IP and port for access and use.

After creating an account and obtaining the public/private network address, you can connect to TDSQL for MySQL through third-party tools and program drivers.

On Windows, connection methods of command line, client, and JDBC driver are taken as examples.

On Linux, connection method of command line is taken as an example.

Connecting from a Windows Device

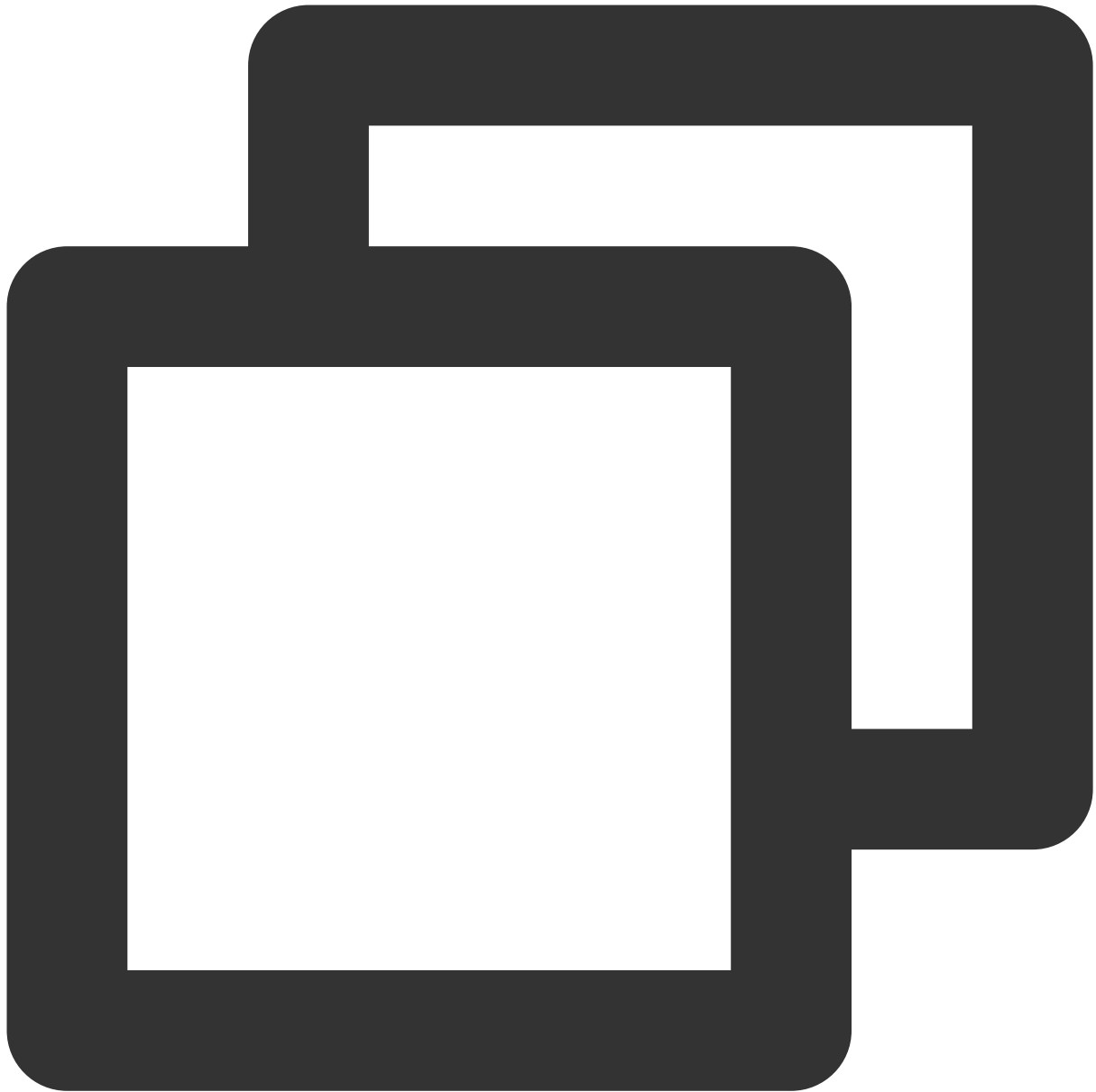
Connecting with Windows command lines

1. Open the Windows command line tool, and enter the following commands under the correct path of mysql.



```
mysql -h public/private IP -P port number -u username -p
Enter password: *****
```

2. If the following information is displayed, the database is connected successfully, and you can operate it.



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

Connecting with the Windows client

1. Download a standard SQL client such as MySQL Workbench and SQLyog. Here we take SQLyog as an example.
2. Open SQLyog, select **File > New Connection**, enter the host address, port, username, and password, and click **Connect**.

MySQL host address: Enter the public/private IP of the instance obtained earlier.

Username: Enter the account name created earlier.

Password: Enter the password of the account. If you forgot the password, reset it in the [console](#).

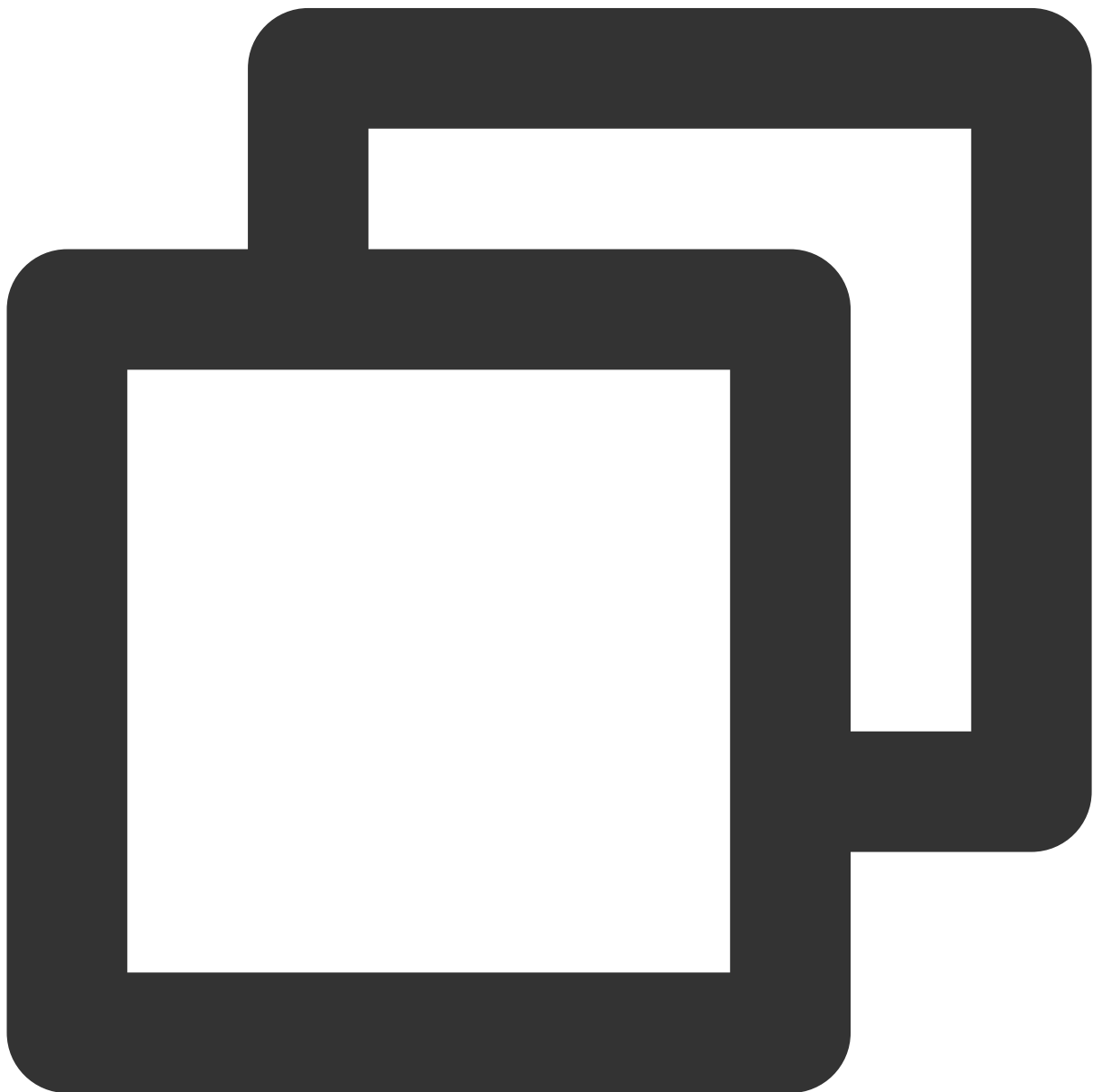
Port: Enter the port number of the instance obtained earlier.

3. If the connection is successful, you will be prompted and can use the database from the page that appears.

Connecting with the Windows JDBC driver

TDSQL for MySQL supports connecting with program drivers. This document uses Java JDBC Driver for MySQL (Connector/J) as an example.

1. [Download](#) a JDBC jar package from the MySQL official website, and import it to the library referenced in Java.
2. Run the following code to call JDBC:



```

public static final String url = "public/private network address";
public static final String name = "com.mysql.jdbc.Driver"; // Call JDBC
public static final String user = "username";
public static final String password = "password";
//JDBC
Class.forName("com.mysql.jdbc.Driver");
Connection conn=DriverManager.getConnection("url, user, password");
//
conn.close();

```

3. After the connection is successful, you can perform other operations on the database.

Connecting from a Linux Device

Connecting with Linux command lines

This document takes CentOS 7.2 64-bit on a CVM instance as an example. For more information on CVM instance purchase, see [Purchasing Channels](#).

1. Log in to the Linux CVM, enter `yum install mysql`, and download and install the MySQL client from Tencent Cloud image source with CentOS's package management software Yum.

```

CentOS Linux 7 (Core)
Kernel 3.10.0-327.36.3.el7.x86_64 on an x86_64

UM_213_43_centos login: root
Password:
[root@UM_213_43_centos ~]# yum install mysql
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.52-1.el7 will be installed
--> Processing Dependency: mariadb-libs(x86-64) = 1:5.5.52-1.el7 for package: 1:mariadb-5.5.52-1.el7.x86_64
--> Running transaction check
---> Package mariadb-libs.x86_64 1:5.5.50-1.el7_2 will be updated
---> Package mariadb-libs.x86_64 1:5.5.52-1.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository
=====
Installing:
mariadb                                x86_64                              1:5.5.52-1.el7                          os
Updating for dependencies:
mariadb-libs                           x86_64                              1:5.5.52-1.el7                          os
=====
Transaction Summary
=====
Install 1 Package
Upgrade (1 Dependent package)

Total download size: 9.5 M
Is this ok [y/d/N]:

```

2. When `complete` is displayed on the command line, the MySQL client is successfully installed.

3. Enter `mysql -h public/private IP -P port -u username -p` to connect to TDSQL for MySQL.

Then, you can perform sharding.

Below uses `show databases;` as an example:

```
[root@UM_3_86_centos ~]# mysql -hcdcbt [REDACTED] myqcloud.com -P12 -uexample1 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 493901845
Server version: 10.0.10-proxy Source distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database                |
+-----+
| SysDB                   |
| information_schema      |
| mysql                   |
| performance_schema      |
| test                    |
+-----+
5 rows in set (0.01 sec)

MySQL [(none)]> _
```

Managing Sharded Table

Last updated : 2024-01-06 17:33:30

This document describes how to perform simple database operations on a sharded table.

Creating a table

For the specific differences among sharded, non-sharded, and broadcast tables, see [Overview](#).

For the restrictions on selecting the shardkey, see [Creating Tables](#).

You need to specify the shardkey when creating a sharded table. Below is the sample code:



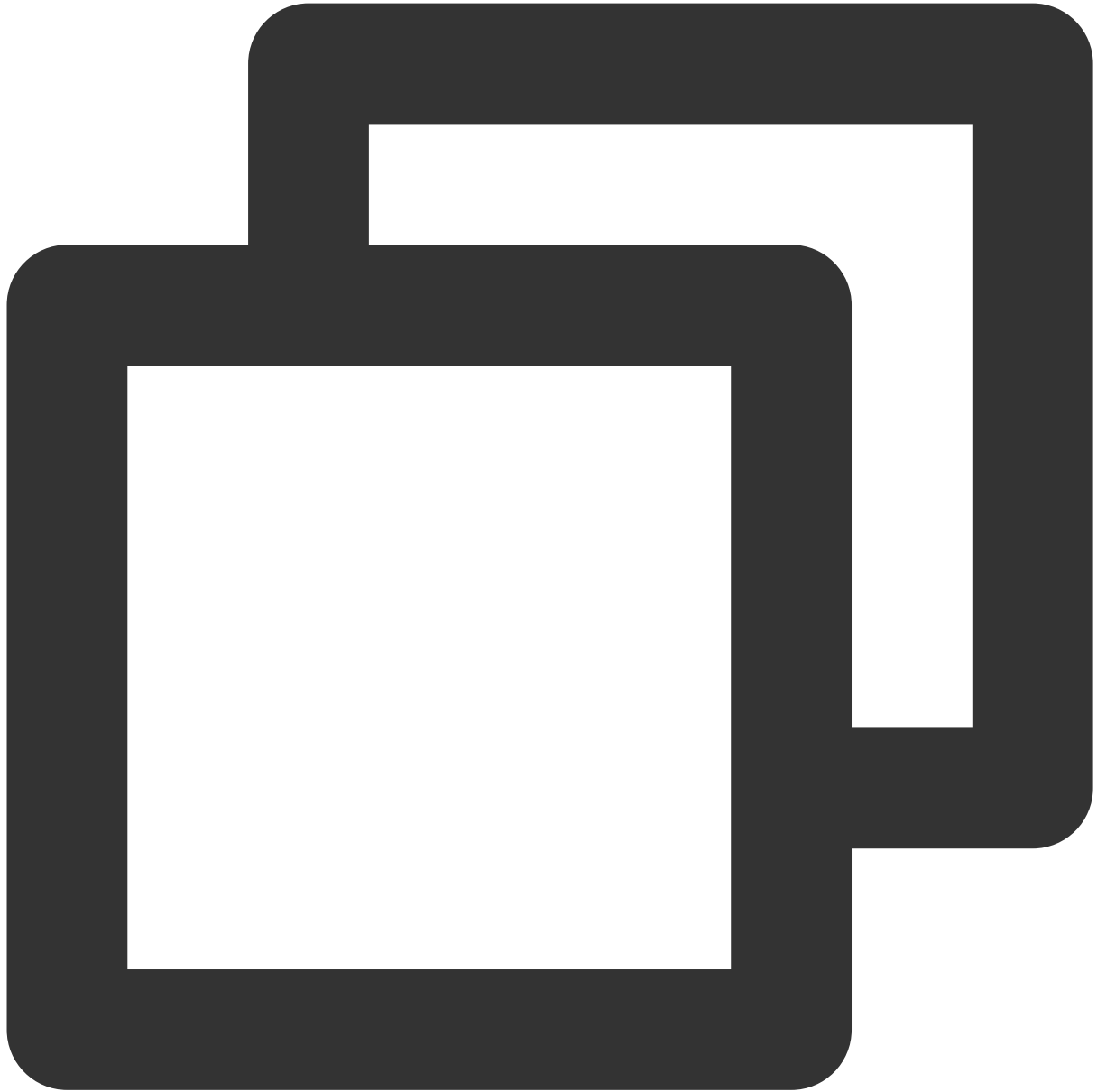
```
mysql> create table test1(id int primary key,name varchar(20),addr varchar(20)) shard
Query OK, 0 rows affected (0.15 sec)
```

Inserting data

Note:

The `insert` field must contain a shardkey; otherwise, the statement will be rejected.

Insert data to the newly created table. Below is the sample code:



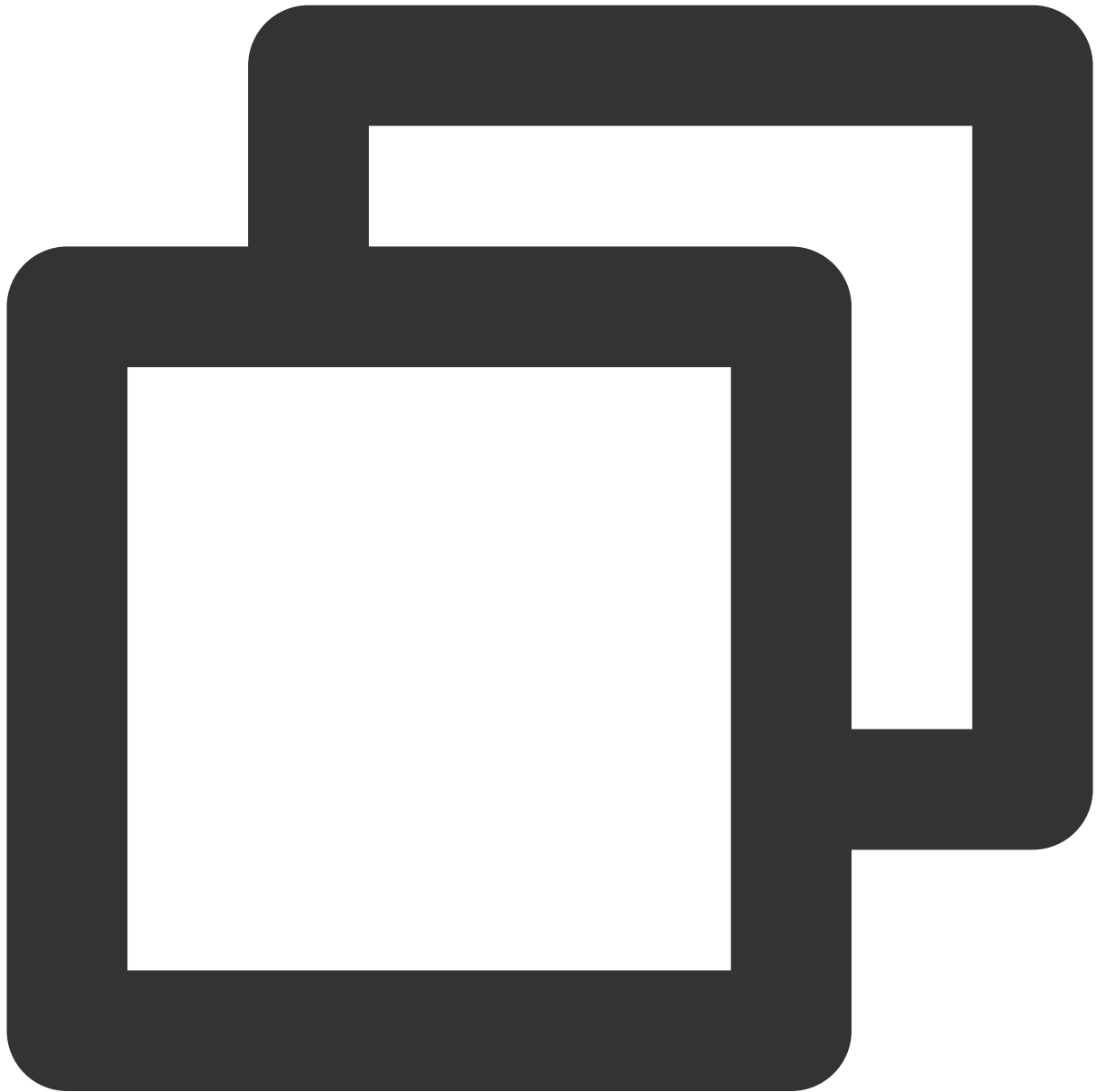
```
mysql> insert into test1(id,name) VALUES(1,'test');  
Query OK,1 rows affected(0.08 sec)  
mysql> insert into test3(name,addr) values('example','shenzhen');  
ERROR 7013 (HY000): Proxy ERROR:get_shardkeys return error
```

Querying data

Note:

We recommend that you carry a shardkey when querying data, so that the corresponding shard will be automatically redirected to through distributed routing, delivering the highest efficiency; otherwise, the distributed system will automatically perform a full-table scan and aggregate the results on the gateway, which is less efficient.

Below is the sample code for data query:



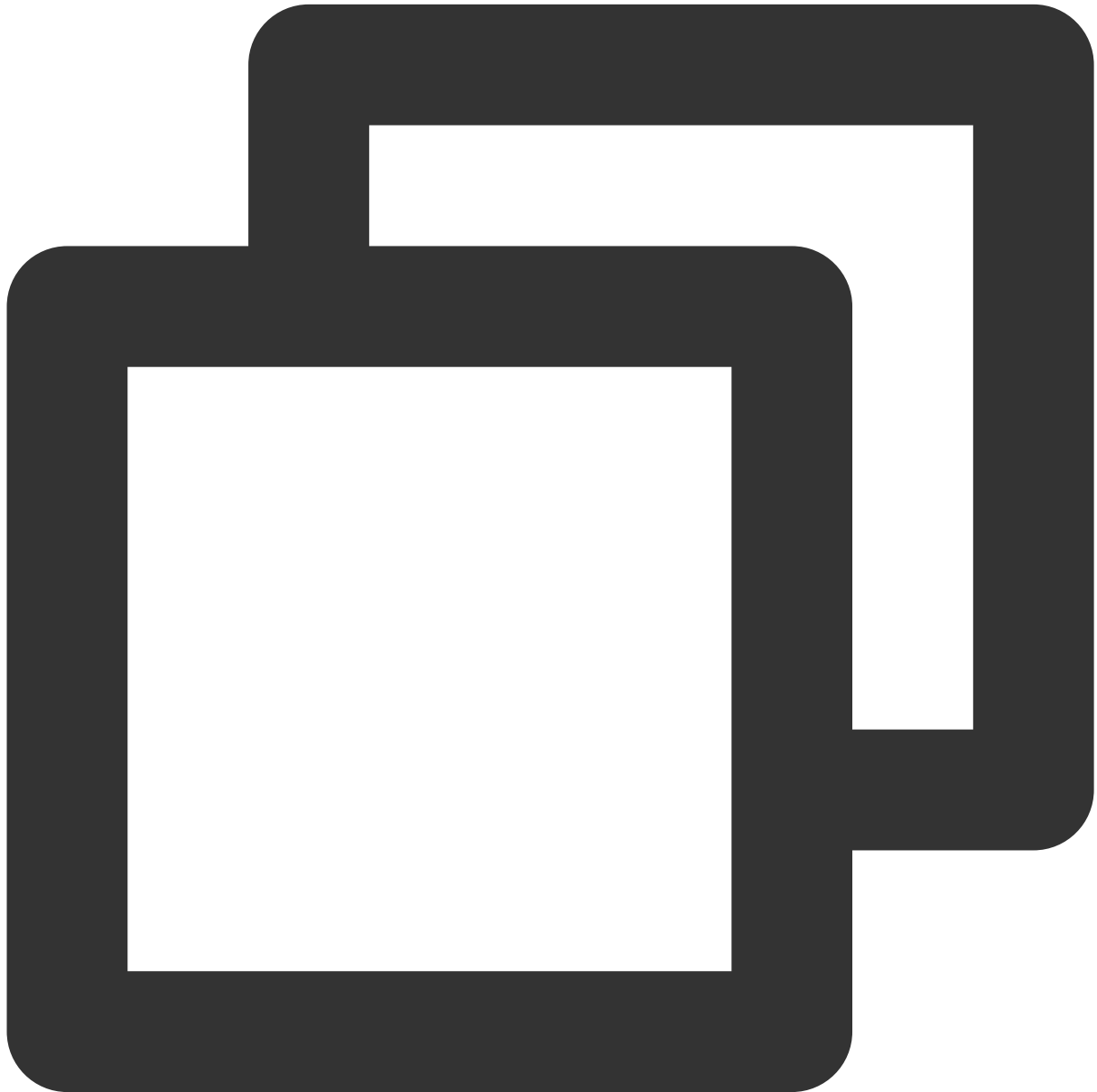
```
mysql> select id from test1 where id=1;
```

Deleting data

Note:

The `delete` field must contain a `WHERE` condition. We recommend that you include the shardkey in the `WHERE` condition.

Below is the sample code for data deletion:



```
mysql> delete from test1 where id=1;  
Query OK, 1 row affected (0.02 sec)
```