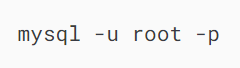
# **Selecting a MySQL Database Using USE Statement**

**Summary**: in this tutorial, you will learn how to select a database in the mysql program and MySQL Workbench by using the USE statement.

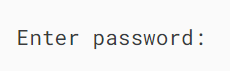
## **Selecting a MySQL database using the mysql client tool**

When you log in to a MySQL database server using the mysql client tool without specifying a database name, MySQL server will set the current database to NULL.

First, log in to MySQL using the root user account:



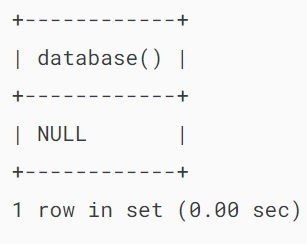
MySQL will prompt you for a password:



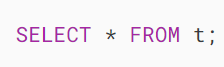
To log in, you need to provide the correct password of the root user account and press Enter. To display the current database, you use the following statement:



It’ll return the following:



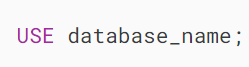
It means the current database is not set. If you issue a statement, MySQL will issue an error. For example:



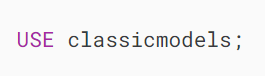
Error:



To select a database to work with, you use the USE statement:



For example, the following statement uses the USE statement to set the current database to classicmodels:



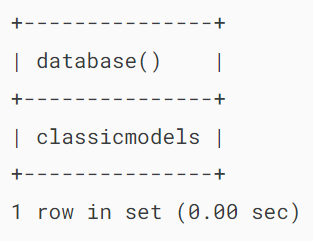
If you see the following message, it means that you have changed the database to classicmodels successfully:



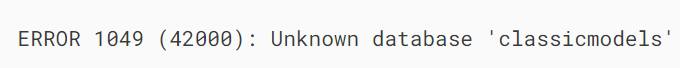
To verify it, you can use the select database() statement:



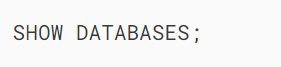
It’ll return something like:



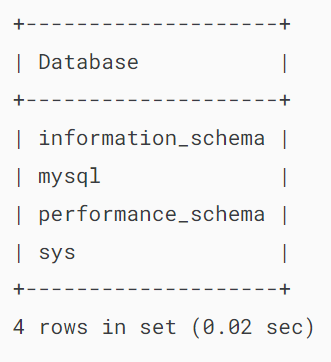
If the classicmodels database doesn’t exist, you’ll get the following error after executing the USE statement:



In this case, you need to find which databases are available on your server by using the show databases statement:

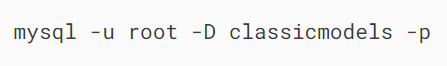


The output may look like the following:



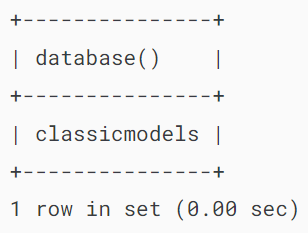
## **Selecting a database when you login**

If you know which database you want to work with before you log in, you can use the -D flag. For example, the following command connects to the classicmodels database with the user account root:



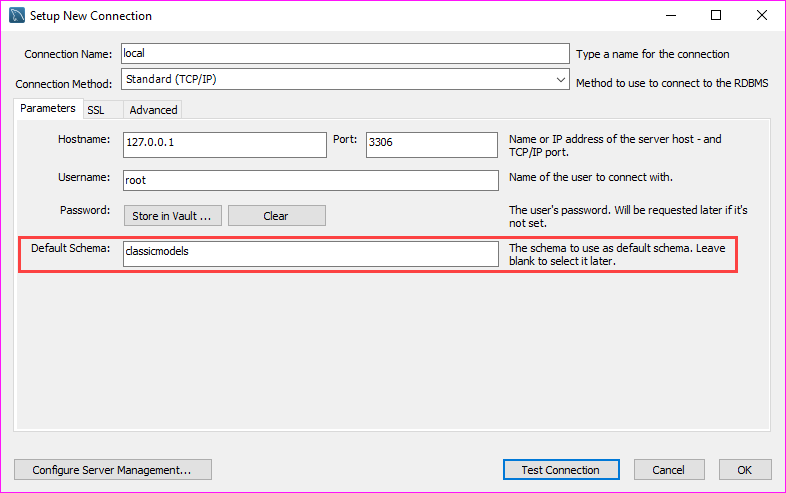
In this command, we specify the database classicmodels after the -D flag.

After entering the password and logging in successfully, you can check the current database:

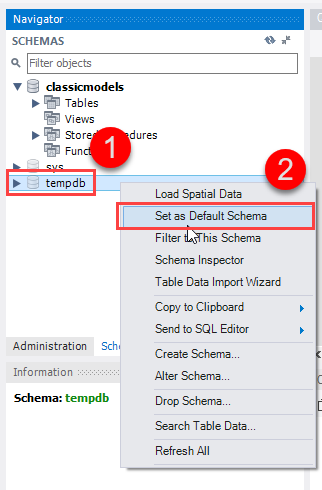


## **Selecting a database in MySQL Workbench**

If you connect to a MySQL Server via the MySQL Workbench application, you can select a database when you create the database connection as shown in the following screenshot:



Once logged in, you can select another database by issuing the USE statement or use the Set As Default Schema feature provided by MySQL Workbench:

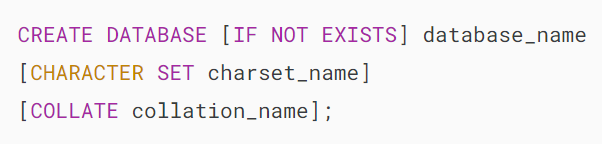


# **MySQL CREATE DATABASE**

**Summary**: in this tutorial, you will learn how to use the MySQL CREATE DATABASE statement to create a new database on a MySQL server.

## **Introduction to the MySQL CREATE DATABASE statement**

To create a new database in MySQL, you use the CREATE DATABASE statement. The following illustrates the basic syntax of the CREATE DATABASE statement:



In this syntax:

* First, specify the name of the database after the CREATE DATABASE keywords. The database name must be unique within a MySQL server instance. If you attempt to create a database with an existing name, MySQL will issue an error.
* Second, use the IF NOT EXISTS option to create a database if it doesn’t exist conditionally.
* Third, specify the [character set](https://www.mysqltutorial.org/mysql-basics/mysql-character-set/) and [collation](https://www.mysqltutorial.org/mysql-basics/mysql-collation/) for the new database. If you skip the CHARACTER SET and COLLATE clauses, MySQL will use the default character set and collation for the new database.

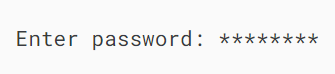
## **Creating a new database using the mysql client tool**

To create a new database via the mysql client tool, you follow these steps:

First, log in to the MySQL server using a user account that has the CREATE DATABASE privilege:

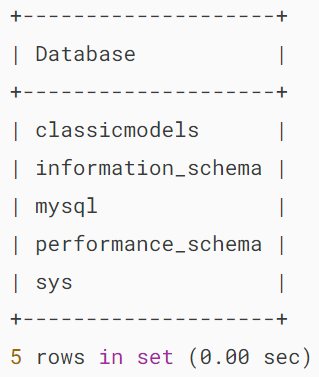


It’ll prompt you to enter a password. To authenticate, you need to type the password for the root user account and press the Enter key:

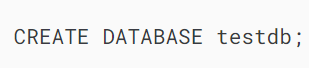


Next, display the databases available on the server using the [SHOW DATABASES](https://www.mysqltutorial.org/mysql-administration/mysql-show-databases/) statement. This step is optional.

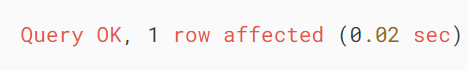




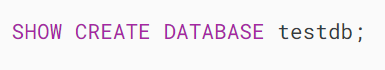
Then, execute the CREATE DATABASE statement to create the testdb database and press Enter:



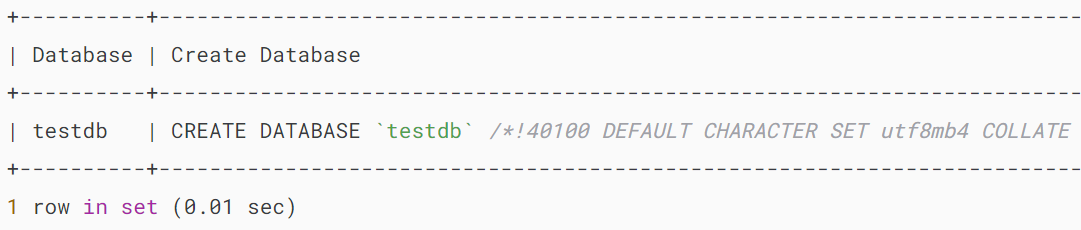
It’ll return the following:



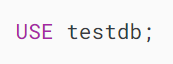
After that, use the SHOW CREATE DATABASE command to review the created database:



MySQL returns the database name and the character set and collation of the database:



Finally, [select the newly created database](https://www.mysqltutorial.org/mysql-select-database/) to work with by using the USE statement:



Output:



Now, you can start [creating tables](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) and other database objects within the  testdb database.

To quit the **mysql** program, type exit command:



Output:

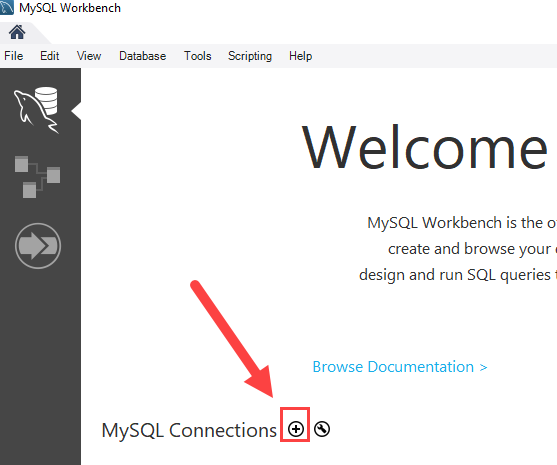


After creating a database, you can [create a user account](https://www.mysqltutorial.org/mysql-administration/mysql-create-user/) and [grant privileges to the database](https://www.mysqltutorial.org/mysql-administration/mysql-grant/).

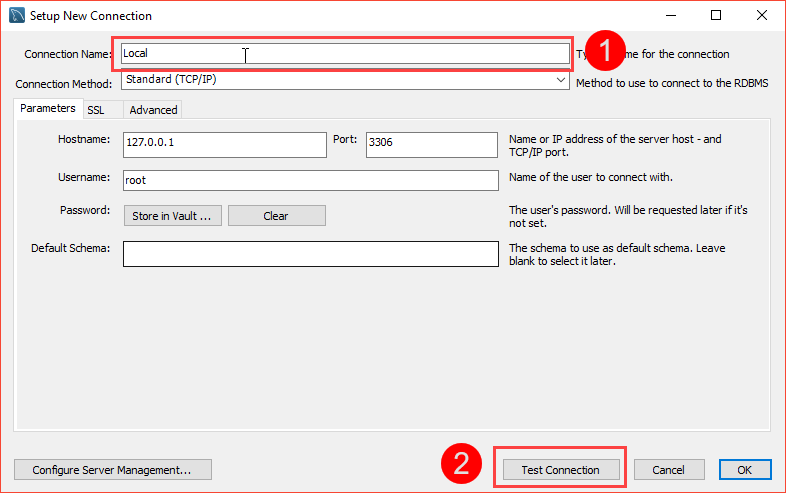
## **Creating a new database using MySQL Workbench**

To create a new database using the MySQL Workbench, you follow these steps:

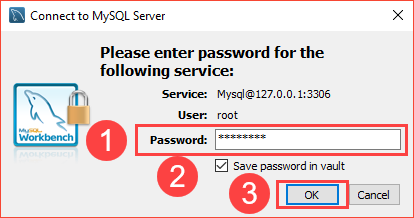
First, launch the MySQL Workbench and click the **setup new connection** button as shown in the following screenshot:



Second, type the name for the connection and click the **Test Connection** button.

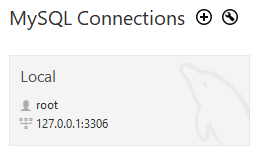


MySQL Workbench displays a dialog asking for the password of the root user:

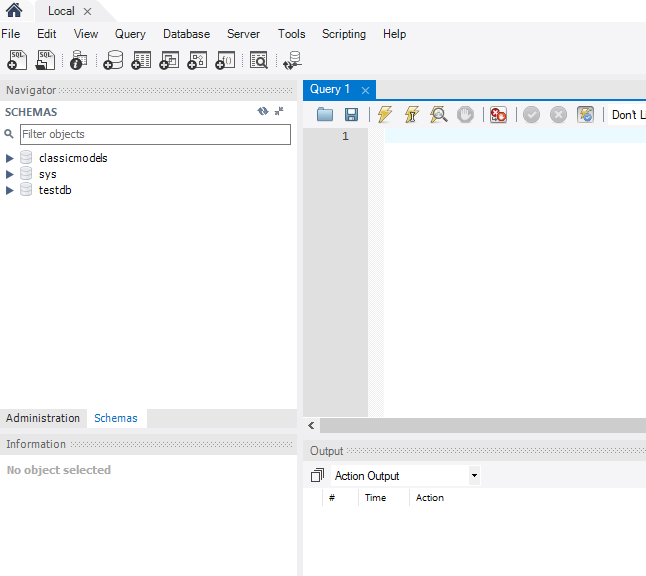


You need to (1) type the password for the root user, (2) check the **Save password in vault**, and (3) click **OK** button.

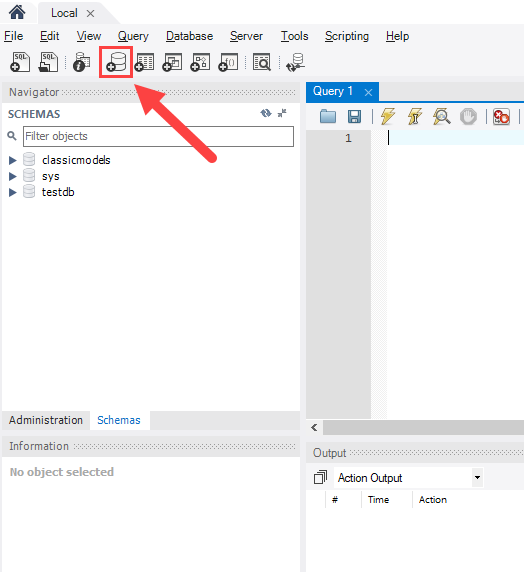
Third, double-click the connection name **Local** to connect to the MySQL Server.



MySQL Workbench opens the following window which consists of four parts: Navigator, Query, Information, and Output.

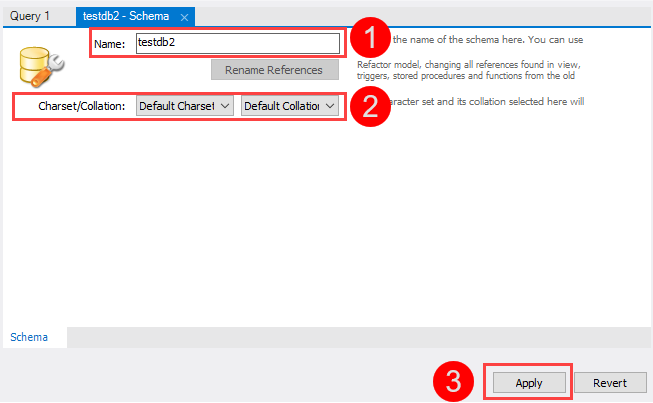


Fourth, click the **create a new schema in the connected server** button from the toolbar:

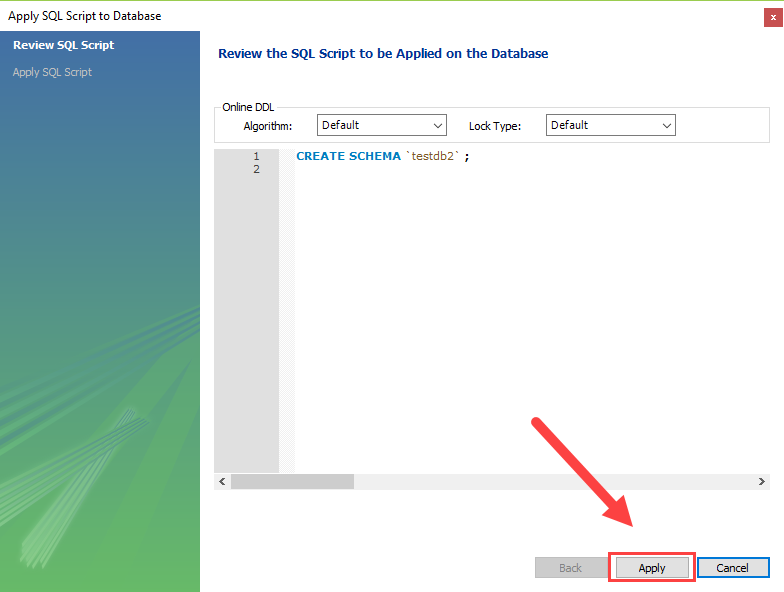


In MySQL, the schema is the synonym for the database. Creating a new schema also means creating a new database.

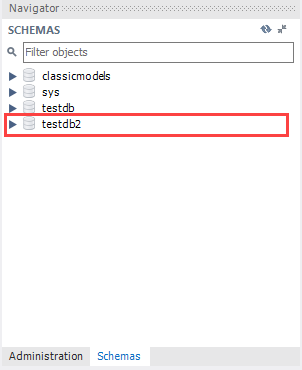
Fifth, the following window is open. You need to (1) enter the schema name, (2) change the character set and collation if necessary, and click the **Apply** button:



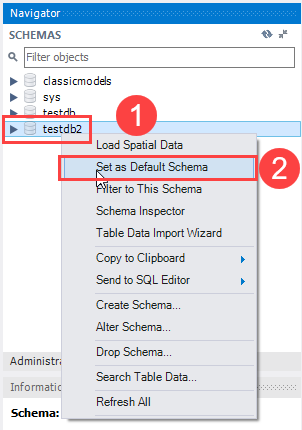
Sixth, MySQL Workbench opens the following window that displays the SQL script that will be executed. Note that the CREATE SCHEMA statement command has the same effect as the CREATE DATABASE statement.



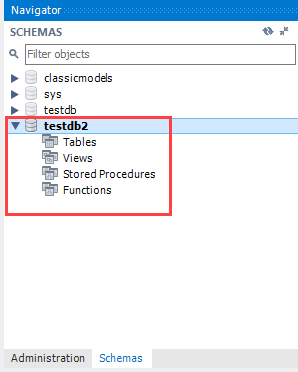
If everything is fine, you will see the new database created and shown in the **schemas** tab of the **Navigator** section.



Seventh, to select the testdb2 database, (1) right-click the database name and (2) choose **Set as Default Schema** menu item:



The testdb2 node is open as shown in the following screenshot.



Now, you can work with testdb2 from the MySQL Workbench.

## **Summary**

* Use the CREATE DATABASE statement to create a new database.
* In MySQL, schemas are synonyms of databases.

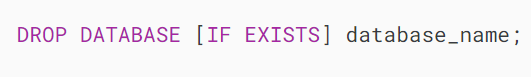
# **MySQL DROP DATABASE**

**Summary**: in this tutorial, you will learn how to use the MySQL DROP DATABASE statement to delete an existing database in the server.

## **Introduction to the MySQL DROP DATABASE statement**

The DROP DATABASE statement drops all tables in the database and deletes the database permanently. Therefore, you need to be very careful when using this statement.

The following shows the syntax of the DROP DATABASE statement:



In this statement, you specify the name of the database which you want to delete after the DROP DATABASE keywords.

If you drop a database that does not exist, MySQL will issue an error.

To prevent an error from occurring if you delete a non-existing database, you can use the IF EXISTS option. In this case, MySQL will terminate the statement without issuing any error.

The DROP DATABASE statement returns the number of tables it deleted.

In MySQL, the schema is the synonym for the database. Therefore, you can use them interchangeably:



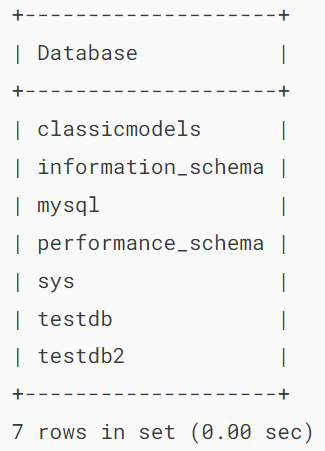
In the next section, we will use the testdb and testdb2 created in the [CREATE DATABASE](https://www.mysqltutorial.org/mysql-basics/mysql-create-database/) tutorial. If you do not have these databases available, you can follow the [previous tutorial](https://www.mysqltutorial.org/mysql-basics/mysql-create-database/) to create them.

## **MySQL DROP DATABASE using mysql program example**

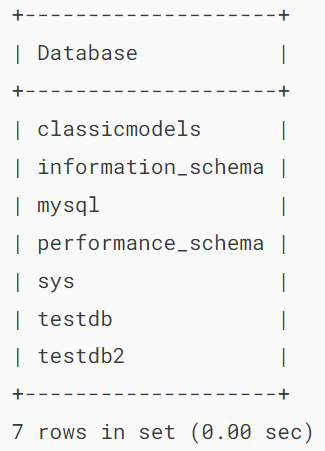
Display all the databases using the [SHOW DATABASES](https://www.mysqltutorial.org/mysql-administration/mysql-show-databases/) statement:



Output:



Third, drop the testdb database by issuing the DROP DATABASE statement:



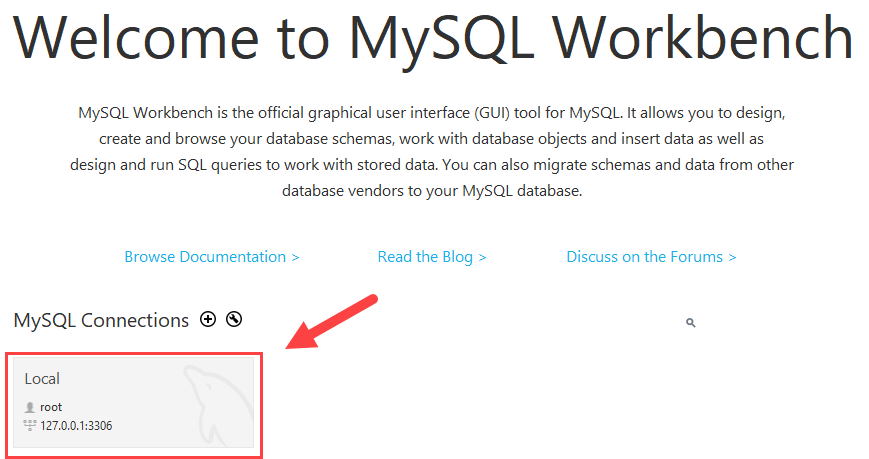
Output:



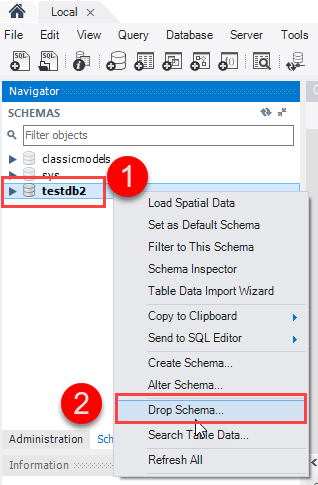
MySQL returned zero affected rows indicating that the testdb database has no tables.

## **DROP DATABASE using MySQL Workbench**

First, launch the MySQL workbench and log in to the MySQL Server.

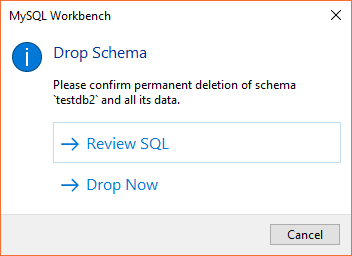


Second, right-click the database that you want to remove, for example, testdb2 and choose the Drop Schema... option.

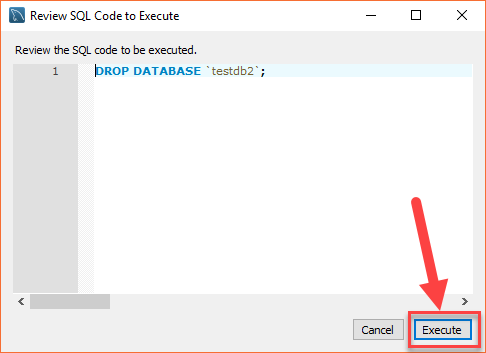


Third, MySQL Workbench displays a dialog to confirm the deletion.

If you choose **Review SQL**, you’ll see the SQL statement that will be executed. If you choose **Drop Now**, it’ll delete the database immediately.

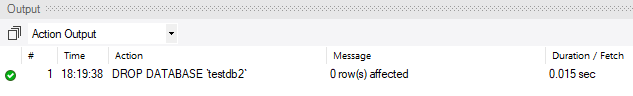


To be safe, let’s choose Review SQL:

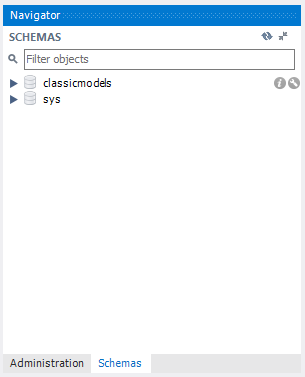


Fourth, once you are sure that the SQL statement is going to drop the right database, you can click the **Execute** button to execute the statement.

MySQL returns the following output indicating that the database is dropped successfully. Because the testdb2 is an empty database, the number of affected rows is zero.



If you view the **schemas** pane, you will see that the **testdb2** is not on the list anymore.



## **Summary**

* Use the MySQL DROP DATABASE statement to delete a database.