7 MySQL MariaDB Create, Alter, Drop Database Command Examples

Once you've installed and configured MySQL or MariaDB, the first step is to create a database.

Only after creating a database, you can create tables and insert records.

This tutorial explains the following examples that are used to create and manipulate a MySQL database:

1. Create New MySQL Database
2. Create MySQL DB with Specific Character Set (UTF8)
3. Delete Existing MySQL Database
4. Create MySQL DB Only If it Doesn't Exists
5. Drop MySQL DB Only If it Exists
6. Alter Database Characteristics for db.opt
7. Upgrade Data Directory Option for Migration and Encoding

1. Create New MySQL Database

To create a MariaDB database, use the create database command as shown below.

The following will create a database called "thegeekstuff".

```
MariaDB [(none)]> CREATE DATABASE thegeekstuff;
Query OK, 1 row affected (0.00 sec)
```

If you have background on Oracle database, don't confuse the term "database" here.

When we are creating a "database" in the MySQL, we are actually creating a "schema". But in MySQL and MariaDB world, it is really called and referred as "database" instead of "schema".

But for some reason, if you prefer, you can also use the following "create schema" command to create a database. Create schema is nothing but a synonym for Create database.

The following command is exactly same as the above create database.

```
MariaDB [(none)]> CREATE SCHEMA thegeekstuff;
```

Please note that only users who have the CREATE privilege for the database can execute the above commands.

In a typical situation, you'll login to mysql as root and execute the above create database command.

```
# mysql -u root -pMySecretPWD
```

CREATE DATABASE creates a database with the given name. To use this statement, you need the CREATE privilege for the database. CREATE SCHEMA is a synonym for CREATE DATABASE.

2. Create MySQL DB with Specific Character Set (UTF8)

Create database will use whatever default character set from your system while creating a new database.

But, if you know exactly what character set you want, you can specify them during the database creation as shown below.

In the following example, we are creating a mariadb database called “tgs” with “utf8” charater set. Here we’ve also specified the collate along with character set.

```
MariaDB [(none)]> CREATE DATABASE tgs CHARACTER SET = utf8 COLLATE =
utf8_general_ci;
Query OK, 1 row affected (0.00 sec)
```

If you like to view all available character set on your system, use the following show character set command.

```
MariaDB [(none)]> SHOW CHARACTER SET;
+----------+-----------------------------+---------------------+--------+
| Charset  | Description                 | Default collation   | Maxlen |
+----------+-----------------------------+---------------------+--------+
| big5     | Big5 Traditional Chinese    | big5_chinese_ci     |      2 |
| dec8     | DEC West European           | dec8_swedish_ci     |      1 |
| cp850    | DOS West European           | cp850_general_ci    |      1 |
| hp8      | HP West European            | hp8_english_ci      |      1 |
| koi8r    | KOI8-R Relcom Russian       | koi8r_general_ci    |      1 |
| latin1   | cp1252 West European        | latin1_swedish_ci   |      1 |
| latin2   | ISO 8859-2 Central European | latin2_general_ci   |      1 |
..
..
| utf8     | UTF-8 Unicode               | utf8_general_ci     |      3 |
```

When you specify a character set during create database, this information is stored in the db.opt file for that particular database.

For example, for the “tgs” database, this db.opt file will be under /var/lib/mysql/tgs directory as shown below.

```
# cat /var/lib/mysql/tgs/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
```

3. Delete Existing MySQL Database

To remove an existing mysql database from your system, use the drop database command as shown below.

The following will delete “thegeekstuff” database.

```
MariaDB [(none)]> DROP DATABASE thegeekstuff;
```

Few things to keep in mind:

As you can imagine, this is a dangerous commands, as this will drop all the tables along with data from the database and then deletes the database itself.
To execute this command you'll need DROP privilege on the database.

Also, similar to create database and create schema, you can also use drop database and drop schema.

Both of the following commands are exactly the same.

```
MariaDB [(none)]> DROP DATABASE thegeekstuff;
MariaDB [(none)]> DROP SCHEMA thegeekstuff;
```

Once you drop a database, execute the show databases to make sure the database is not listed anymore.

```
MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
```

When you drop the database, it will also delete the database directorhttps://www.amazon.es/plegable-retr%C3%A1ctil-almacenamiento-contenedor-Caravaning/dp/B0727Y7XSP/ref=pd_lutyp_wish_4_15?_encoding=UTF8&pd_rd_i=B0727Y7XSP&pd_rd_r=18b23308-8cdb-4230-adc2-e5c94f524b72&pd_rd_w=qwEJS&pd_rd_wg=niERQ&psc=1&refRID=53C4M5HZSWM2G1MCAQC6y along with all the files including the db.opt from the /var/lib/mysql folder as shown below.

```
# ls -l /var/lib/mysql/thegeekstuff/
ls: cannot access /var/lib/mysql/thegeekstuff/: No such file or directory
```

One thing to keep in mind is that if you have manually created some file under your database directory (i.e /var/lib/mysql/thegeekstuff), then the above drop database command will not remove your custom files or the database directory itself. However, it will remove all other files that was created by the mysql server itself including the table files, etc.

On a related note, if you have created any TEMPORARY tables, they'll not be removed. But, they'll be removed automatically whenever the particular sessions that created it ends.

4. Create MySQL DB Only If it Doesn't Exists

As shown below, create database command by default will fail when you try to create a database that already exists.

```
MariaDB [(none)]> CREATE DATABASE thegeekstuff;
ERROR 1007 (HY000): Can't create database 'thegeekstuff'; database exists
```

That might be Ok if you are just doing it on command line, as you can imply ignore and move on.

But, if you are doing it inside a script, which checks for any error message from a SQL command and exists the script, then we have a problem.

In that case, use the “if not exists” clause along with create database as shown below.

This will create the database only if it doesn’t exists. But, when the database exists, it will not thrown any error message.

```
MariaDB [(none)]> CREATE DATABASE IF NOT EXISTS thegeekstuff;
Query OK, 1 row affected, 1 warning (0.00 sec)
```

Similar to the create database, you can also do drop database if exists clause as shown below.

```
DROP DATABASE IF EXISTS thegeekstuff;
DROP SCHEMA IF EXISTS thegeekstuff;
```

5. Drop MySQL DB Only If it Exists

When you try to drop a database that doesn’t exists, you’ll get the following error message.

```
MariaDB [none]> DROP DATABASE thegeekstuff;
ERROR 1008 (HY000): Can't drop database 'thegeekstuff'; database doesn't exist
```

This might be Ok when you are doing it interactively from command line. But, if are executing the drop command from an automated script, and you don’t to see the error message, that might stop the script, then use the IF EXISTS clause as shown below.

Same behavior as the above command, but doesn’t return any error message.

```
MariaDB [none]> DROP DATABASE IF EXISTS thegeekstuff;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

6. Alter Database Characteristics for db.opt

Using alter database command, you can change certain characteristics of your database.

Typically, you’ll use this to change the DB options that are set in the db.opt file.

For example, here we have the character set defined as LATIN for thegeekstuff database in the db.opt

```
# cat /var/lib/mysql/thegeekstuff/db.opt
default-character-set=latin1
default-collation=latin1_swedish_ci
..
```

Now, to change this to UTF8, we can use the alter database command as shown below.

```
MariaDB [(none)]> ALTER DATABASE thegeekstuff CHARACTER SET = utf8 COLLATE =
utf8_general_ci;
Query OK, 1 row affected (0.00 sec)
```

Anytime you execute an ALTER database command, you’ll see the value that you updated are reflected in the db.opt file as shown below.

```
# cat /var/lib/mysql/thegeekstuff/db.opt
default-character-set=utf8
```

```
default-collation=utf8_general_ci
```

To execute the alter database command, you’ll see the ALTER privilege on the database.

Just like create and drop, you can also use “schema”. Both of the following are exactly the same, as ALTER SCHEMA is just a synonym to alter database.

```
ALTER DATABASE thegeekstuff CHARACTER SET = utf8 COLLATE = utf8_general_ci;

ALTER SCHEMA thegeekstuff CHARACTER SET = utf8 COLLATE = utf8_general_ci;
```

Also, note that you can omit the database name in the “ALTER” command when you are already inside the database as shown below.

As you see here, I did USE to change the database. So, in my ALTER DATABASE, I didn’t have to specify the database name.

```
MariaDB [(none)]> USE thegeekstuff;

MariaDB [thegeekstuff]> ALTER DATABASE CHARACTER SET = utf8 COLLATE =
utf8_general_ci;
Query OK, 1 row affected (0.00 sec)
```

7. Upgrade Data Directory Option for Migration and Encoding

If you are running an alter version of MySQL database (prior to 5.1), and trying to upgrade to a version of MySQL / MariaDB that is 5.1 or later, then you need to do this.

The following command with the “upgrade data directory name” option will update the name of the database directory with the appropriate encoding implementation that matches the MySQL 5.1 or above version. This will make sure the database name and database directory is mapped properly without any encoding issue in the name.

Again, you’ll use this only when you are upgrading from an older version of MySQL to 5.1 or later version, AND when you have special characters in your database name as shown below which has hypen in the database name (i.e the-geek-stuff).

Also, note that the “#mysql50#” is the keyword that should be used as it is followed by the database name.

```
ALTER DATABASE `#mysql50#the-geek-stuff` UPGRADE DATA DIRECTORY NAME;
```

In this example, the above command will encode the database name properly as “the@002dgeek@002dstuff”, where it will use @002d for the dash special character in the database name.