

# MySQL database

**Database** is oftenly termed as back-end process, as because it is neither visible to end user nor **End User** interacts directly with the database. They works on front-end process viz., **PHP**, **VB**, **ASP.NET**, etc. and ask the front end to deal with database in back-end.

There are several database server and client available like **Oracle**, **MySQL**, **MySQLi**, [MariaDB](#), [MongoDB](#) etc. The syntax of all of these are more or less the same. Mastering one means gaining control on most of them and learning the queries of a database is very easy and fun. Lets start with simple queries on database. We will be using **MySQL** which comes bundled with most of the **Linux** distributions by default, you could install it manually from repository, if it is not installed by default in your case.

Well a database query is a simple piece of code that is sent to database to get custom and refined result, as required.

## Install MySQL Database

Use “**yum**” or “**apt**” package manager to install **MySQL** Database.

```
# yum install mysql mysql-client mysql-server (on Yum based Systems)
```

```
# apt-get install mysql mysql-client mysql-server (on Apt based Systems)
```

## Start MySQL

Start **MySQL** database service as:

```
# service mysqld start
```

or

```
# service mysql start
```

Well installing a **MySQL** database will take you to the configuration where you are asked to setup **admin** password, etc. Once finished installing and starting the server go to your **MySQL** prompt.

```
# mysql -u root -p
```

Replace **root** with your configured **username** and enter **password** when prompted, if the login credential is correct, you will be at your **MySQL** prompt at the blink of your eyes.

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

```
Your MySQL connection id is 195
```

Server version: 5.5.31-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Other names may be trademarks of their respective owners.

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

```
mysql>
```

Now carrying out queries at this prompt is very educative and fun.

## Create a database tecmint

```
mysql> create database tecmint ;
```

```
Query OK, 1 row affected (0.02 sec)
```

```
mysql>
```

**Note:** It reports that the query was correct, means database is created. You can verify your newly created database as.

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| mysql              |
```

```
| performance_schema |
```

```
| tecmint            |
```

```
| test               |
```

```
+-----+
```

```
9 rows in set (0.00 sec)
```

```
mysql>
```

**Note:** Notice your database in the above output.

## Select Database

Now you need to select the database to work upon it.

```
mysql> use tecmint;
```

```
Database changed
```

```
mysql>
```

## Create Tables in MySQL

Here we will be creating a table say “**minttec**” with three fields as:

```
mysql> CREATE TABLE minttec (
```

```
    -> id Int(3),
```

```
    -> first_name Varchar (15),
```

```
    -> email Varchar(20)
```

```
    -> );
```

```
Query OK, 0 rows affected (0.08 sec)
```

```
mysql>
```

**Note:** The above query says **OK** which means table was created without any error. To verify the table run the below query.

```
mysql> show tables;
```

```
+-----+
```

```
| Tables_in_tecmint |
```

```
+-----+
```

```
| minttec           |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql>
```

Things are going fine till now. Yup! You can view the columns you created in the table “**minttec**” as:

```
mysql> show columns from minttec;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(3)        | YES  |     | NULL    |       |
| first_name | varchar(15)   | YES  |     | NULL    |       |
| email      | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```



```
3 rows in set (0.00 sec)
```

```
mysql>
```

It was nothing less than a magic. Anyway I will tell you about the types of declaration and their meaning.

1.**Int** is Integer

2.**Varchar** is char having variable length as defined. The value after Type is the length of field up-to which it can store data.

OK now we need to add a column say '**last\_name**' after column '**first\_name**'.

```
mysql> ALTER TABLE minttec ADD last_name varchar (20) AFTER  
first_name;
```

```
Query OK, 0 rows affected (0.16 sec)
```

```
Records: 0  Duplicates: 0  Warnings: 0
```

Now, verify it in your table.

```
mysql> show columns from minttec;
```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(3)        | YES  |     | NULL    |       |
| first_name | varchar(15)   | YES  |     | NULL    |       |
| last_name  | varchar(20)   | YES  |     | NULL    |       |
| email      | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

```

4 rows in set (0.00 sec)

```
mysql>
```

## Add Column in MySQL

Now we will add a column to the right say a column '**country**' to the right of **email**.

```
mysql> ALTER TABLE minttec ADD country varchar (15) AFTER  
email;
```

```
Query OK, 0 rows affected (0.16 sec)
```

```
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql>
```

Verify the above column insertion query.

```
mysql> show columns from minttec;
```

```
+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id          | int(3)         | YES  |     | NULL     |       |
| first_name  | varchar(15)    | YES  |     | NULL     |       |
| last_name   | varchar(20)    | YES  |     | NULL     |       |
| email       | varchar(20)    | YES  |     | NULL     |       |
| country     | varchar(15)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql>
```

## Insert Values in Field

What about inserting values to the field?

```
mysql> INSERT INTO minttec VALUES ('1' , 'Ravi' , 'Saive' ,  
'raivsaive@xyz.com' , 'India' );
```

```
Query OK, 1 row affected (0.02 sec)
```

```
mysql>
```

How about inserting more than **1** value at a time in the above table.

```
mysql> INSERT INTO minttec VALUES ('2' , 'Narad' ,  
'Shrestha' , 'narad@xyz.com' , 'India' ), ('3' , 'user' ,
```

```
'singh' , 'user@xyz.com' , 'Aus' ), ('4' , 'tecmint' ,  
'[dot]com' , 'tecmint@gmail.com' , 'India' );
```

Query OK, 3 rows affected (0.05 sec)

Records: 3 Duplicates: 0 Warnings: 0

Verify the above insertion.

```
mysql> select * from minttec;
```

```
+-----+-----+-----+-----+  
+-----+
```

```
| id   | first_name | last_name | email          |  
country |
```

```
+-----+-----+-----+-----+  
+-----+
```

```
|    1 | Ravi      | Saive     | raivsaive@xyz.com | India  
|
```

```
| 2 | Narad | Shrestha | narad@xyz.com | India  
|
```

```
| 3 | user | singh | user@xyz.com | Aus  
|
```

```
| 4 | tecmint | [dot]com | tecmint@gmail.com | India  
|
```

```
+-----+-----+-----+-----+  
+-----+
```

```
4 rows in set (0.00 sec)
```

```
mysql>
```

## Delete Values in Field

Let's say the third entry in the above output is invalid and we need to delete the third entry.

```
mysql> DELETE FROM minttec WHERE id = 3;
```

```
Query OK, 1 row affected (0.02 sec)
```

Verify the above operation.

```
mysql> select * from minttec;
```

```
+-----+-----+-----+-----+
+-----+
```

```
| id   | first_name | last_name | email          |
country |
```

```
+-----+-----+-----+-----+
+-----+
```



```

|    1 | Ravi          | Saive      | raivsaive@xyz.com | India
|
|
|    2 | Narad         | Shrestha   | narad@xyz.com      | India
|
|
|    4 | tecmint       | [dot]com   | tecmint@gmail.com  | India
|
|
+-----+-----+-----+-----+
+-----+
3 rows in set (0.00 sec)

```

## Update Values in Field

The id (=4) needs to be edited.

```
mysql> UPDATE minttec SET id = 3 WHERE first_name =
'tecmint';
```

```
Query OK, 1 row affected (0.02 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>
```

Verify the above query.

```
mysql> UPDATE minttec SET id = 3 WHERE first_name =  
'tecmint';
```

```
Query OK, 1 row affected (0.02 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>
```

**Note:** The above query, as performed is not a good idea. It will change the id to '4' where ever the first name is '**tecmint**'. It is always a good idea to use more than one column with where clause to get minimal error, as:

```
mysql> UPDATE minttec SET id = 6 WHERE first_name =  
'tecmint'AND last_name = '[dot]com';
```

```
Query OK, 1 row affected (0.03 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>
```

## Delete Column in MySQL

Let we need to drop (delete) a column we think, is of no importance say '**country**' here.

```
mysql> ALTER TABLE minttec drop country;
```

```
Query OK, 3 rows affected (0.15 sec)
```

```
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql>
```

Verify the table.

```
mysql> select * from minttec;
```

```
+-----+-----+-----+-----+
```

```
| id   | first_name | last_name | email                |
```

```
+-----+-----+-----+-----+
```

```
|    1 | Ravi      | Saive     | raivsaive@xyz.com |
```

```
|    2 | Narad     | Shrestha  | narad@xyz.com      |
```

```
|    6 | tecmint   | [dot]com  | tecmint@gmail.com |
```

```
+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

```
mysql>
```

## Rename Table in MySQL

Don't you think our table name "**minttec**" is not very much relevant. How about changing it to **tecmin\_table**.

```
mysql> RENAME TABLE minttec TO tecmin_table;
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
mysql>
```

## List all Tables

See all the tables under the current database.

```
mysql> show tables;
```

```
+-----+
```

```
| Tables_in_tecmint |
```

```
+-----+
```

```
| tecmint_table      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql>
```

The table has been renamed. Now take a backup of the above **MySQL** database, in a single line of command without any sophisticated tool. Run the below code at your terminal and not on mysql prompt.

```
# mysqldump -u root -p tecmint > tecmint.sql
```

check the dumped file on your desktop which would have contents something like

```
-- MySQL dump 10.13  Distrib 5.5.31, for debian-linux-gnu (i686) --
```

```
-- Server version 5.5.31-0+wheezy1 --
```

```
Dump completed on 2013-09-02 12:55:37
```

It is always a good idea to maintain [Backup of MySQL databases](#). Restoring the backed up **MySQL** Data is again a simple line of code you need to run at your terminal prompt and not at your mysql prompt.

But, wait first we will delete the database to verify if our restore is perfect.

Delete a Database

```
mysql> drop database tecmint;
```

```
Query OK, 1 row affected (0.02 sec)
```

Check for database 'tecmint' on your database server.

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| my_database |
```

```
| mysql |
```

```
| performance_schema |
```

```
| phpmyadmin |
```



```
|  siso          |  
  
|  test          |  
  
+-----+  
  
7 rows in set (0.00 sec)  
  
mysql>
```

Great! The database is lost, but we need not to worry, we are having the backup.

## Restore a Database

To restore lost database, run the following command.

```
# mysql -u root -p tecmint < tecmint.sql  
  
Enter password:  
  
ERROR 1049 (42000): Unknown database 'tecmint'
```

**OOPS!** An error, hey we have not create the database **tecmint**. So go to your mysql prompt and create a database '**tecmint**'.

```
mysql> create database tecmint;  
  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```

Now time to run restore command at your shell prompt (strictly).

```
# mysql -u root -p tecmint < tecmint.sql  
  
Enter password:
```

Verify your database.

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| mysql |
```

```
| performance_schema |
```

```
| tecmint |
```

```
| test |
```

```
+-----+
```

```
8 rows in set (0.00 sec)
```

Verify the contents of database.

```
mysql> show tables from tecmint;
```

```
+-----+
```

```
| Tables_in_tecmint |
```

```
+-----+
```

```
| tecmint_table      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql>
```

Verify the contents of your restored table.

```
mysql> select * from tecmint_table;
```

```
+-----+-----+-----+-----+
```

```
| id    | first_name | last_name | email                |
```

```
+-----+-----+-----+-----+
```

```
|      1 | Ravi       | Saive     | raivsaive@xyz.com |
```

```
|      2 | Narad      | Shrestha  | narad@xyz.com      |
```

```
|      6 | tecmint    | [dot]com  | tecmint@gmail.com |
```

```
+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

This is not the end definitely, we will cover the concept of **primary key, foreign key, multiple tables** and **running queries** using simple **PHP** script in the next part of the article.

Don't forget to **tell us**, how you felt while going through the article. Your comments are highly appreciated. Stay **Healthy** and **Tuned**, remain connected to **Tecmint**.

=====