

EXPERMINT: 04

- **Title:** MySQL database on Raspberry Pi
- **Aim:** To install MySQL database on Raspberry Pi and perform basic SQL queries.
- **Theory:**

MySQL is a real-time relational database management system that organizes data into one or more data tables. The data stored in the tables may be related to each other, thus, ensuring an organized data structure. This database management system is mostly used in creating web servers or accessing another server and is an ideal choice for both small and large-scale applications. It can handle the most expensive and powerful database packages and is simple to use on any system.

- **Procedure:**

1) How to Setup MySQL Database on Raspberry Pi

The MariaDB server is introduced by MySQL developers as a drop-in replacement for the MySQL server.

Therefore, the MySQL server is no longer available in the Raspberry Pi repository.

1.1) To install and setup MySQL (MariaDB) database, follow the below-mentioned steps:

```
$ sudo apt install mariadb-server -y

pi@raspberrypi:~$ sudo apt install mariadb-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gdebi-core glib-2.0 vte-2.91 python3-debian
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  mariadb-server
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 35.3 kB of archives.
After this operation, 72.7 kB of additional disk space will be used.
Get:1 http://raspbrian.raspberrypi.org/raspbian bullseye/main armhf mariadb-server
  all 1:10.5.15-0+deb11u1 [35.3 kB]
Fetched 35.3 kB in 2s (15.9 kB/s)
Selecting previously unselected package mariadb-server.
(Reading database ... 145954 files and directories currently installed.)
Preparing to unpack .../mariadb-server_1%3a10.5.15-0+deb11u1_all.deb ...
Unpacking mariadb-server (1:10.5.15-0+deb11u1) ...
(Making disappearance of apt-ntop, which has been completely replaced.)
disappear
Setting up mariadb-server (1:10.5.15-0+deb11u1) ...
Progress: [ 50%] [#####.....]
```

1.2) After completing the installation, you must secure your MySQL database from the following command: \$ sudo mysql_secure_installation.

```
pi@raspberrypi:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
```

1.3) Enter your system password to configure MySQL database setup. Then choose the default option “n” to stay with the root account instead of switching to unix_socket.

```
pi@raspberrypi:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.
Switch to unix_socket authentication [Y/n] █
```

1.4) Enter “n” to stick with the default root password.

```
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.
Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.
Change the root password? [Y/n] █
```

1.5) Remove the anonymous users by entering “Y”.

```
You already have your root account protected, so you can safely answer 'n'.
Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] █
```

You can disable the root login remotely or enable it according to your choice. If write “n”.

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] █
```

1.6) Enter “Y” to remove the test database or you can reply with “n” too.

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] █
```

1.7) Reload the privilege tables by entering “Y” to make the changes.

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
Reload privilege tables now? [Y/n] Y
```

This completes the MySQL database configuration.

```
Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
pi@raspberrypi:~$
```

2) Create a Database Using MySQL

2.1) Now, to start creating your first database through MySQL, first, execute the following command to log in to MySQL. `$ sudo mysql`.

```
pi@raspberrypi:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.5.15-MariaDB-0+deb11u1 Raspbian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

2.2) Then create a database using the following syntax:

`CREATE DATABASE <database_name>;`

For example, the name of the database is “sqldatabase”:

`CREATE DATABASE sqldatabase;`

```
MariaDB [(none)]> CREATE DATABASE sqldatabase;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]>
```

2.3) Create a username and password for your database and then grant all privileges to your database

using the following syntax: `GRANT ALL PRIVILEGES ON <database_name>.* TO '<username>'@'localhost' IDENTIFIED BY '<password>';`

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON sqldatabase.* TO 'linuxhint'@'localhost'
IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]>
```

2.4) Replace the <username> and <password> in the above command.

Finalize the changes by flushing the privilege table through the following command:

FLUSH PRIVILEGES;

```
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.003 sec)
```

2.5) Exit to the database through the “exit” command.

```
MariaDB [(none)]> exit  
Bye  
pi@raspberrypi:~ $
```

This completes the MySQL database creation on the Raspberry Pi system.

● Conclusion and Result:

The combination of MySQL database installation on Raspberry Pi and basic SQL queries for data analysis offers a powerful and accessible solution for managing and analyzing data collected by Raspberry Pi-based projects. Whether for educational purposes, hobbyist projects, or professional applications, this setup provides a versatile and scalable platform for exploring the potential of data-driven computing on the Raspberry Pi ecosystem.