MySQL Create User

The MySQL user is a record in the **USER** table of the MySQL server that contains the login information, account privileges, and the host information for MySQL account. It is essential to create a user in MySQL for accessing and managing the databases.

If you want to use the Create User, it is required to have a **global** privilege of Create User statement or the **INSERT** privilege for the MySQL system schema. When you create a user that already exists, it gives an error. But if you use, **IF NOT EXISTS** clause, the statement gives a warning for each named user that already exists instead of an error message.

Why Did Users require in MySQL server?

When the MySQL server installation completes, it has a **ROOT** user account only to access and manage the databases. But, sometimes, you want to give the database access to others without granting them full control. In that case, you will create a non-root user and grant them specific privileges to access and modify the database.

```
CREATE USER [IF NOT EXISTS] account_name IDENTIFIED BY 'password';
```

MySQL CREATE USER Example

The following are the step required to create a new user in the MySQL server database.

Step 1: Open the MySQL server by using the **mysql client tool**.

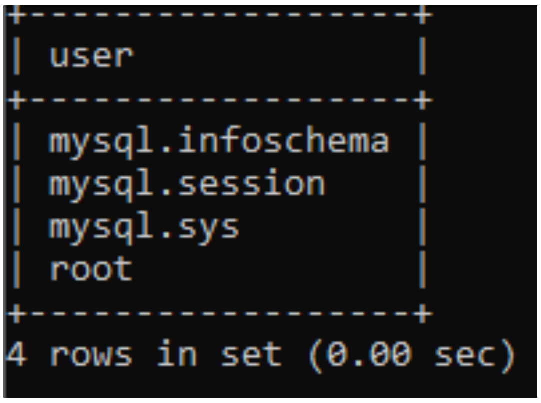
Step 2: Enter the password for the account and press Enter.

```
Enter Password: ********
```

Step 3: Execute the following command to show all users in the current MySQL server.

```
mysql> select user from mysql.user;
```

We will get the output as below:

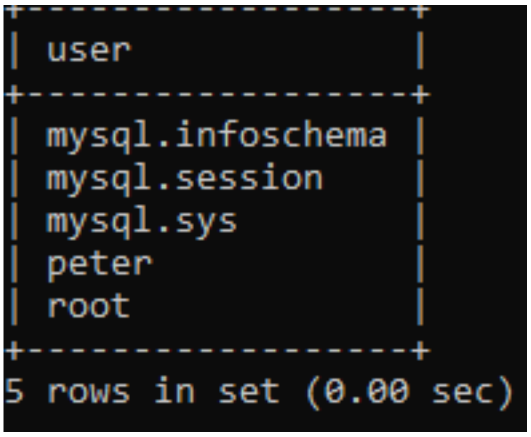


```
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
+------------------+
4 rows in set (0.00 sec)
```

Step 4: Create a new user with the following command.

mysql> **create** user peter@localhost identified **by** 'jtp12345';

Now, run the command to show all users again.



```
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| peter            |
| root             |
+------------------+
5 rows in set (0.00 sec)
```

In the above output, we can see that the user **peter** has been created successfully.

Step 5: Now, we will use the IF NOT EXISTS clause with the CREATE USER statement.

mysql> **CREATE** USER IF NOT EXISTS adam@localhost IDENTIFIED **BY** 'jtp12345
6';

Grant Privileges to the MySQL New User

MySQL server provides multiple types of privileges to a new user account. Some of the most commonly used privileges are given below:

1. **ALL PRIVILEGES:** It permits all privileges to a new user account.
2. **CREATE:** It enables the user account to create databases and tables.
3. **DROP:** It enables the user account to drop databases and tables.
4. **DELETE:** It enables the user account to delete rows from a specific table.
5. **INSERT:** It enables the user account to insert rows into a specific table.

6. **SELECT:** It enables the user account to read a database.
7. **UPDATE:** It enables the user account to update table rows.

If you want to give all privileges to a newly created user, execute the following command.

If you want to give all privileges to a newly created user, execute the following command.

If you want to give all privileges to a newly created user, execute the following command.

```
mysql> GRANT ALL PRIVILEGES ON * . * TO peter@localhost;
```

If you want to give specific privileges to a newly created user, execute the following command.

```
mysql> GRANT CREATE, SELECT, INSERT ON * . * TO peter@localhost;
```

Sometimes, you want to **flush** all the privileges of a user account for changes occurs immediately, type the following command.

```
FLUSH PRIVILEGES;
```

If you want to see the existing privileges for the user, execute the following command.

```
mysql> SHOW GRANTS for training@localhost;
```

MySQL Drop User

The MySQL Drop User statement allows us to **remove** one or more user accounts and their **privileges** from the database server. If the account does not exist in the database server, it gives an error.

If you want to use the Drop User statement, it is required to have a **global** privilege of Create User statement or the **DELETE** privilege for the MySQL system schema.

```
DROP USER 'account_name';
```

MySQL Drop USER Example

The following are the step required to delete an existing user from the MySQL server database.

Step 1: Open the MySQL server by using the **mysql client tool**.

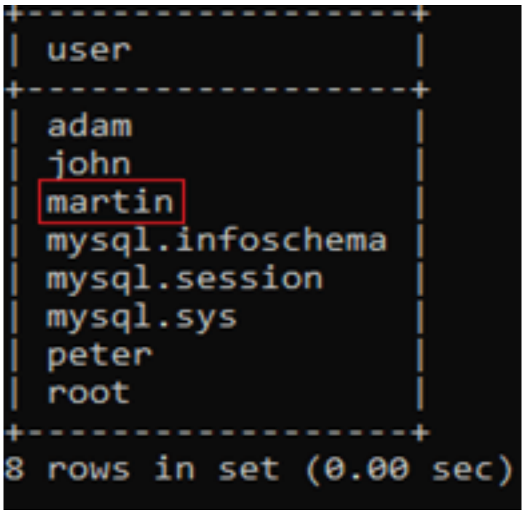
Step 2: Enter the password for the account and press Enter.

Enter **Password**: ********

Step 3: Execute the following command to show all users in the current MySQL server.

mysql> **select** user **from** mysql.user;

We will get the output as below:

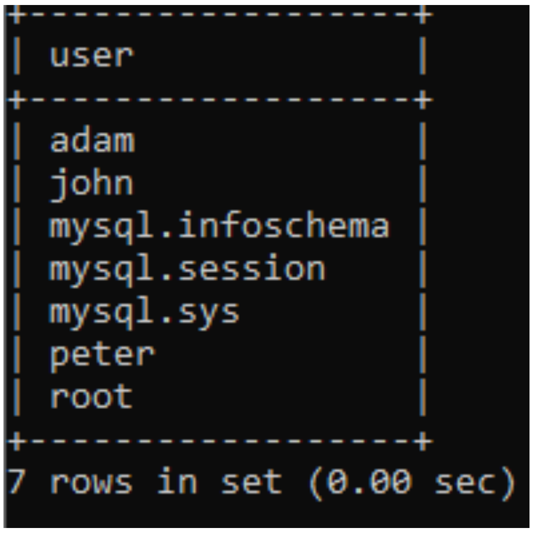


```
+------------------+
| user             |
+------------------+
| adam             |
| john             |
| martin           |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| peter            |
| root             |
+------------------+
8 rows in set (0.00 sec)
```

Step 4: To drop a user account, you need to execute the following statement.

DROP USER martin@localhost;

Here, we are going to remove the username '**martin**' from the MySQL server. After the successful execution of the above command, you need to execute the show user statement again. You will get the following output where username martin is not present.



```
+------------------+
| user             |
+------------------+
| adam             |
| john             |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| peter            |
| root             |
+------------------+
7 rows in set (0.00 sec)
```

Step 5: The DROP USER statement can also be used to remove more than one user accounts at once. We can drop multiple user accounts by separating account_name with **comma** operator. To delete multiple user accounts, execute the following command.

```
DROP USER john@localhost, peter@localhost;
```

Here, we are going to remove **john** and **peter** accounts from the above image. After the successful execution of the above command, you need to execute the show user statement again. You will get the following output where username john and peter is not present.

MySQL Show Users/List All Users

Sometimes you want to manage a database in MySQL. In that case, we need to see the list of all user's accounts in a database. Most times, we assume that there is a **SHOW USERS** command similar to SHOW DATABASES, SHOW TABLES, etc. for displaying the list of all users available in the database server. Unfortunately, MySQL database does not have a SHOW USERS command to display the list of all users in the MySQL server. We can use the following query to see the list of all user in the database server:

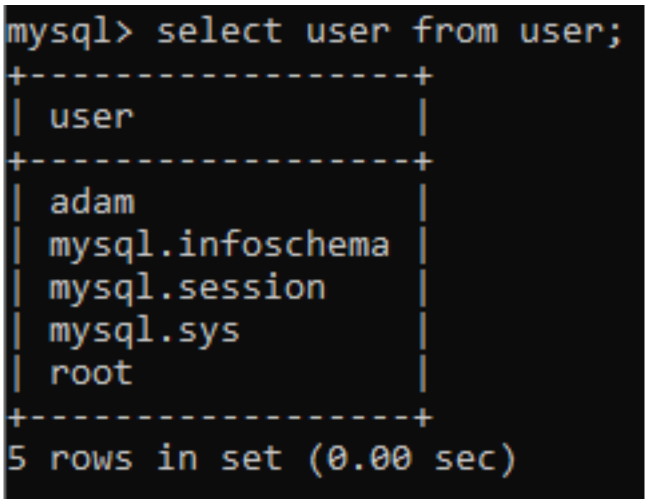
```
mysql> Select user from mysql.user;
```

After the successful execution of the above statement, we will get the user data from the user table of the MySQL database server.

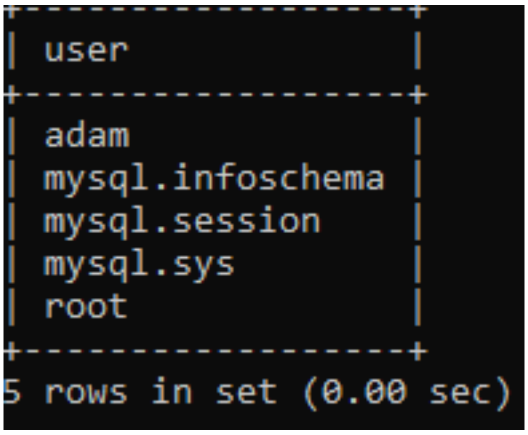
Let us see how we can use this query. First, we have to open the MySQL server by using the **mysql client tool** and log in as an administrator into the server database. Execute the following query:

```
> mysql -u root -p
Enter password: *********
mysql> use mysql;
Database changed
mysql> SELECT user FROM user;
```

We will get the following output where we can see the **five** users in our local database:



```
mysql> select user from user;
+------------------+
| user             |
+------------------+
| adam             |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
+------------------+
5 rows in set (0.00 sec)
```

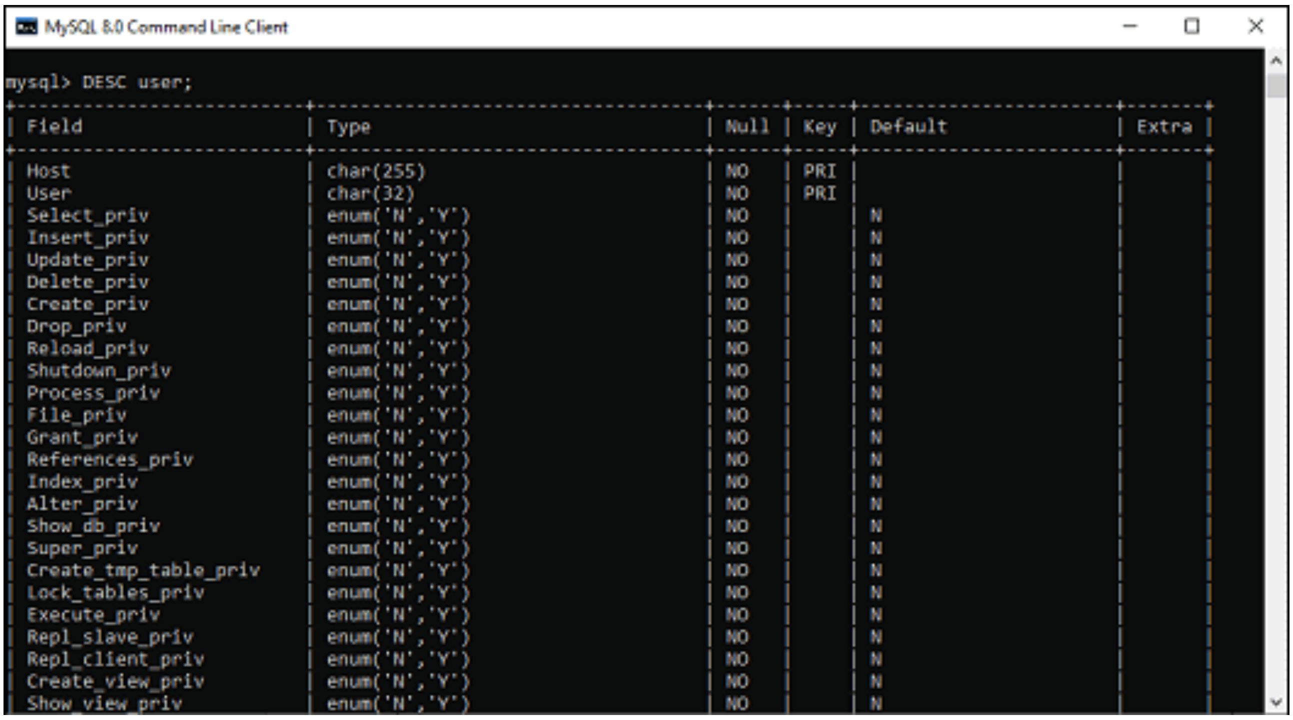


```
+------------------+
| user             |
+------------------+
| adam             |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
+------------------+
5 rows in set (0.00 sec)
```

If we want to see more information on the user table, execute the command below:

```
mysql> DESC user;
```

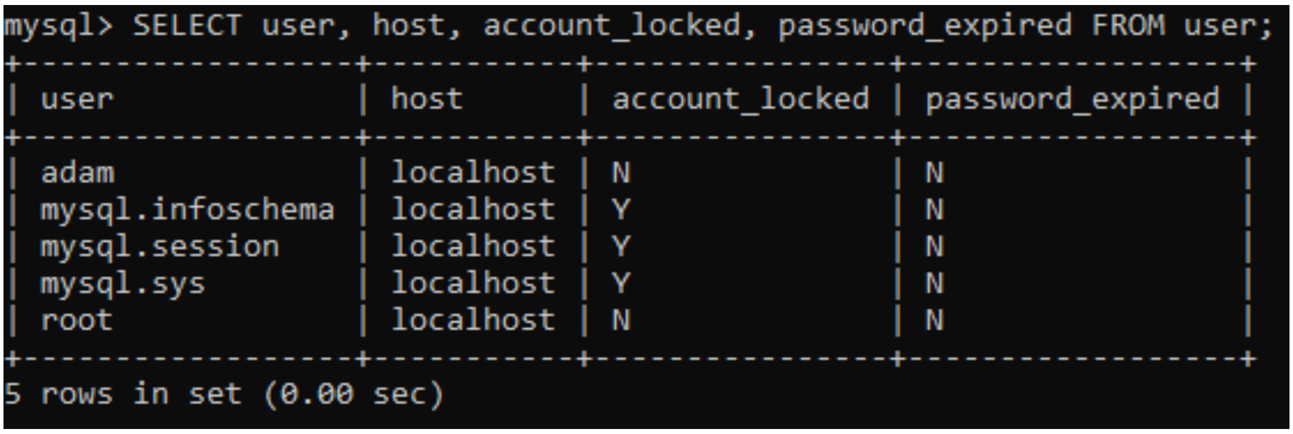
It will give the following output that lists all the available columns of the **mysql.user** database:



```
mysql> DESC user;
+------------------------+-----------------------------------+------+-----+-----------------------+-------+
| Field                  | Type                              | Null | Key | Default               | Extra |
+------------------------+-----------------------------------+------+-----+-----------------------+-------+
| Host                   | char(255)                         | NO   | PRI |                       |       |
| User                   | char(32)                          | NO   | PRI |                       |       |
| Select_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Insert_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Update_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Delete_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Create_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Drop_priv              | enum('N','Y')                     | NO   |     | N                     |       |
| Reload_priv            | enum('N','Y')                     | NO   |     | N                     |       |
| Shutdown_priv          | enum('N','Y')                     | NO   |     | N                     |       |
| Process_priv           | enum('N','Y')                     | NO   |     | N                     |       |
| File_priv              | enum('N','Y')                     | NO   |     | N                     |       |
| Grant_priv             | enum('N','Y')                     | NO   |     | N                     |       |
| References_priv        | enum('N','Y')                     | NO   |     | N                     |       |
| Index_priv             | enum('N','Y')                     | NO   |     | N                     |       |
| Alter_priv             | enum('N','Y')                     | NO   |     | N                     |       |
| Show_db_priv           | enum('N','Y')                     | NO   |     | N                     |       |
| Super_priv             | enum('N','Y')                     | NO   |     | N                     |       |
| Create_tmp_table_priv  | enum('N','Y')                     | NO   |     | N                     |       |
| Lock_tables_priv       | enum('N','Y')                     | NO   |     | N                     |       |
| Execute_priv           | enum('N','Y')                     | NO   |     | N                     |       |
| Repl_slave_priv        | enum('N','Y')                     | NO   |     | N                     |       |
| Repl_client_priv       | enum('N','Y')                     | NO   |     | N                     |       |
| Create_view_priv       | enum('N','Y')                     | NO   |     | N                     |       |
| Show_view_priv         | enum('N','Y')                     | NO   |     | N                     |       |
```

To get the selected information like as hostname, password expiration status, and account locking, execute the query as below:

mysql> **SELECT** user, host, account_locked, password_expired **FROM** user;
After the successful execution, it will give the following output:

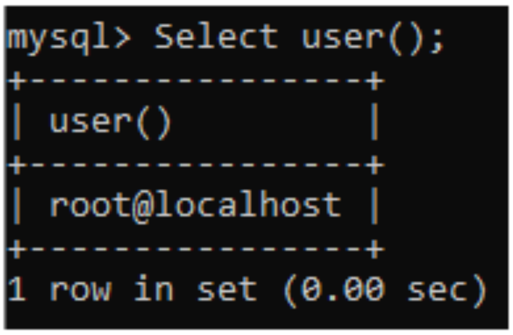


```
mysql> SELECT user, host, account_locked, password_expired FROM user;
+------------------+-----------+----------------+------------------+
| user             | host      | account_locked | password_expired |
+------------------+-----------+----------------+------------------+
| adam             | localhost | N              | N                |
| mysql.infoschema | localhost | Y              | N                |
| mysql.session    | localhost | Y              | N                |
| mysql.sys        | localhost | Y              | N                |
| root             | localhost | N              | N                |
+------------------+-----------+----------------+------------------+
5 rows in set (0.00 sec)
```

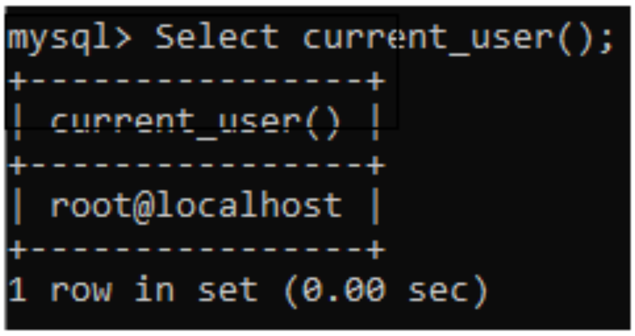
Show Current User

We can get information of the current user by using the **user() or current_user()** function, as shown below:

mysql> **Select** user();
or,
mysql> **Select** current_user();
After executing the above command, we will get the following output:



```
mysql> Select user();
+----------------+
| user()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```



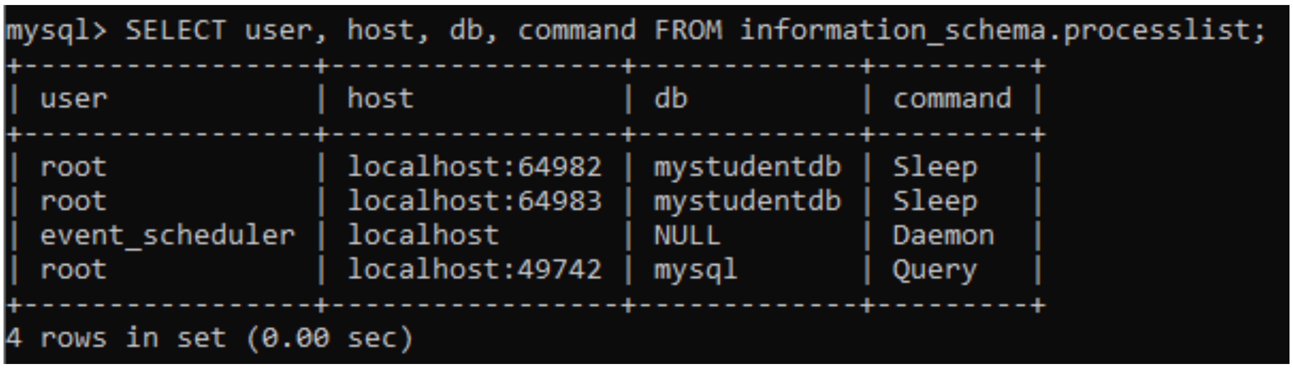
```
mysql> Select current_user();
+----------------+
| current_user() |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

Show Current Logged User

We can see the currently logged user in the database server by using the following query in the MySQL server:

mysql> **SELECT** user, host, db, command **FROM** information_schema.processlist;

The above command gives the output, as shown below:



```
mysql> SELECT user, host, db, command FROM information_schema.processlist;
+-----------------+-----------------+-------------+---------+
| user            | host            | db          | command |
+-----------------+-----------------+-------------+---------+
| root            | localhost:64982 | mystudentdb | Sleep   |
| root            | localhost:64983 | mystudentdb | Sleep   |
| event_scheduler | localhost       | NULL        | Daemon  |
| root            | localhost:49742 | mysql       | Query   |
+-----------------+-----------------+-------------+---------+
4 rows in set (0.00 sec)
```

In this output, we can see that there are currently **four** users logged in the database, where one is executing a **Query**, and others show in **Sleep or Daemon** status.

Change MySQL User Password

MySQL user is a record that contains the login information, account privileges, and the host information for MySQL account to access and manage the database. The login information includes the user name and password. In some cases, there is a need to change the user password in the MySQL database.

To change the password of any user account, you must have to keep this information in your mind:

- ○ The details of the user account that you want to change.

- An application used by the user whose password you want to change. If you reset the user account password without changing an application connection string, then the application cannot connect with the database server.

MySQL allows us to change the user account password in three different ways, which are given below:

1. UPDATE Statement
2. SET PASSWORD Statement
3. ALTER USER Statement

Let us see how we can change the user account password in MySQL by using the above statement in detail:

Change user account password using the UPDATE statement

This statement is the first way to change the user password for updating the user table of the MySQL database. Here, you have to use the **FLUSH PRIVILEGE** statement after executing an UPDATE statement for reloading privileges from the grant table of the MySQL database.

Suppose, you want to change or update the password for a user **peter** that connects from the localhost with the password **jtp12345**, execute the SQL statements as below:

```
mysql> USE mysql;

mysql> UPDATE user SET password = PASSWORD('jtp12345') WHERE user = 'p
eter' AND host = 'localhost';

mysql> FLUSH PRIVILEGES;
```

If you are using the MySQL version **5.7.6** or higher, the above statement will not work. It is because the MySQL user table contains the **authentication_string** column that stores the password only. Now, the higher versions contain the authentication_string column in the UPDATE statement, like the following statement.

```
mysql> USE mysql;

mysql> UPDATE user SET authentication_string = PASSWORD('jtp12345') WHER
E user = 'peter' AND host = 'localhost';

mysql> FLUSH PRIVILEGES;
```

Change user account password using SET PASSWORD statement

The SET PASSWORD statement is the second way to change the user password in the MySQL database. If you want to change the other account password, you must have the UPDATE privilege. The SET PASSWORD statement uses the user account in the **username@localhost** format.

There is no need to use the FLUSH PRIVILEGES statement for reloading privileges from the grant tables of the MySQL database. We can use the following statement to change the password of user account peter by using the SET PASSWORD statement:

```
mysql> SET PASSWORD FOR 'peter'@'localhost' = PASSWORD('jtp12345');
```

If you are using the MySQL version 5.7.6 or higher, the above statement deprecated and will not work in future releases. Instead, we need to use the following statement:

```
mysql> SET PASSWORD FOR 'peter'@'localhost' = jtp12345;
```

Change user account password using ALTER USER statement

The ALTER USER statement is the third way to change the user password in the MySQL database. MySQL uses ALTER USER statement with the IDENTIFIED BY clause for changing the password of a user account. We need to use the following syntax to change the password of a user **peter** with **jtp123**.

```
mysql> ALTER USER peter@localhost IDENTIFIED BY 'jtp123';
```

Sometimes, you need to reset the MySQL **root** account password. In that case, you can force to stop and restart the MySQL database server without using the grant table validation.

Below are few case studies which you can develop.

Home professional online services	Add new service , portal to review , apply for service, schedule service admin user
Loan Underwriting Process	workflow , loan underwriters process under UW admin user, comparison
Retail , Inventory	purchase , upload products , verify product with
Customer loyalty program	Accessing points for purchase bill , evaluate and provide loyalty points under admin user
Advertisement company portal	advertisement upload , evaluation , price and duration on ad
Online Ticket services	ticket booking workflow , customer care services , ticket lifecycle