DBMS - Experiment 3

Name: Ayush Jain

Sapid: 60004200132

Div: B1

Branch: Computer Engineering

AIM:

Create and populate database using Data Definition Language (DDL) and DML Commands

Theory:

DDL

Data Definition Language (DDL) statements are used to define the database structure or schema. Data Definition Language understanding with database schemas and describes how the data should consist in the database, therefore language statements like CREATE TABLE or ALTER TABLE belongs to the DDL. DDL is about "metadata".

DDL includes commands such as CREATE, ALTER and DROP statements. DDL is used to CREATE, ALTER OR DROP the database objects (Table, Views, Users).

Data Definition Language (DDL) are used different statements:

1. CREATE - to create objects in the database
    a) CREATE Database – creates the database
    b) CREATE Tables – creates tables in a particular database
2. ALTER - alters the structure of the database

    a) Adding new columns
    b) Dropping a column from the table
    c) Modifying existing column
    d) Renaming existing table
3. DROP - delete objects from the database
4. TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
5. RENAME - rename an object
6. SHOW – shows the available databases and tables
7. DESCRIBE – gets the information about the structure of the table

DML

A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

In any database we have queries regarding the CRUD operations namely Create, Read, Update, Delete to maintain data. In SQL we have Data Definition Language (DDL) which includes the following commands:

1. INSERT – to insert data (single/multiple) into the tables
2. SELECT – to select data from the tables
3. UPDATE – to update the existing data from the tables
4. DELETE – to delete one/more data from the table

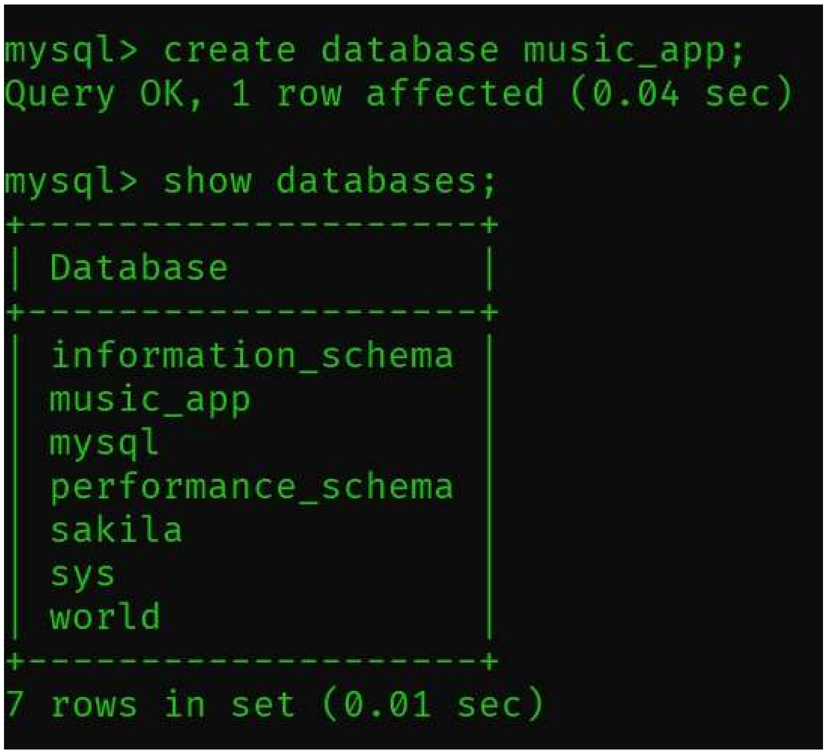
DML Queries

1) CREATE

a) **To create a database**

<u>Syntax:</u>
CREATE DATABASE *dbname*;

<u>Example:</u> CREATE DATABASE *music_app*;

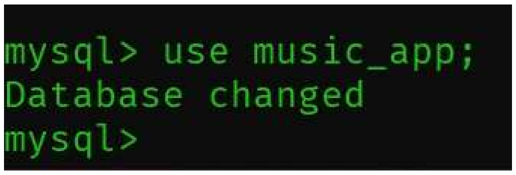


```
mysql> create database music_app;
Query OK, 1 row affected (0.04 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| music_app          |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
7 rows in set (0.01 sec)
```

b) **To select existing database**

<u>Syntax:</u>
Use *dbname*;

<u>Example:</u> USE *music_app*;

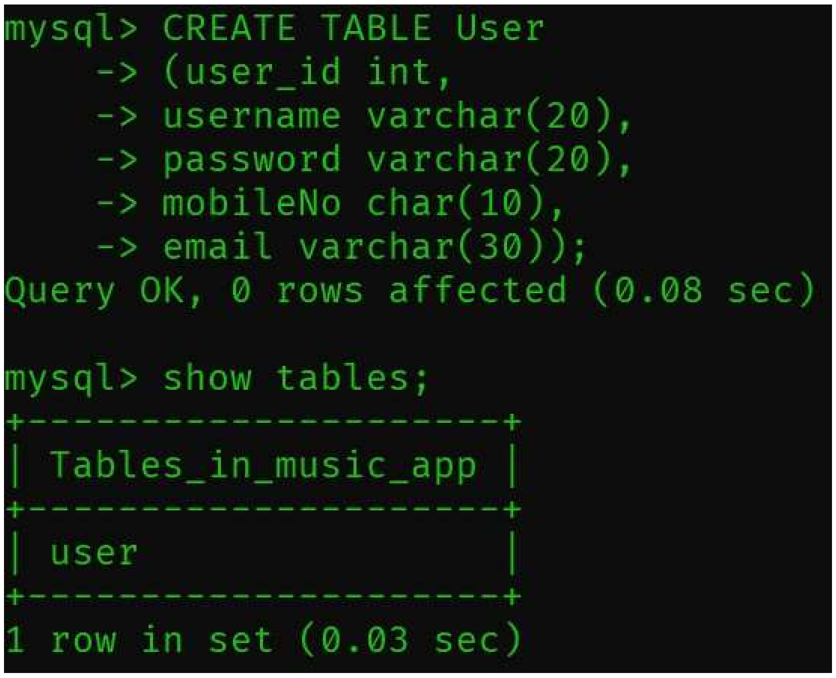


```
mysql> use music_app;
Database changed
mysql>
```

c) **To create a Table user_id**

Syntax:
CREATE TABLE *table_name* (fieldname1 datatype(), fieldname2 datatype() ...);

Example:
CREATE TABLE *User*
(user_id int, username
varchar(20), password
varchar(20),
mobileNo char(10),
email varchar(30));



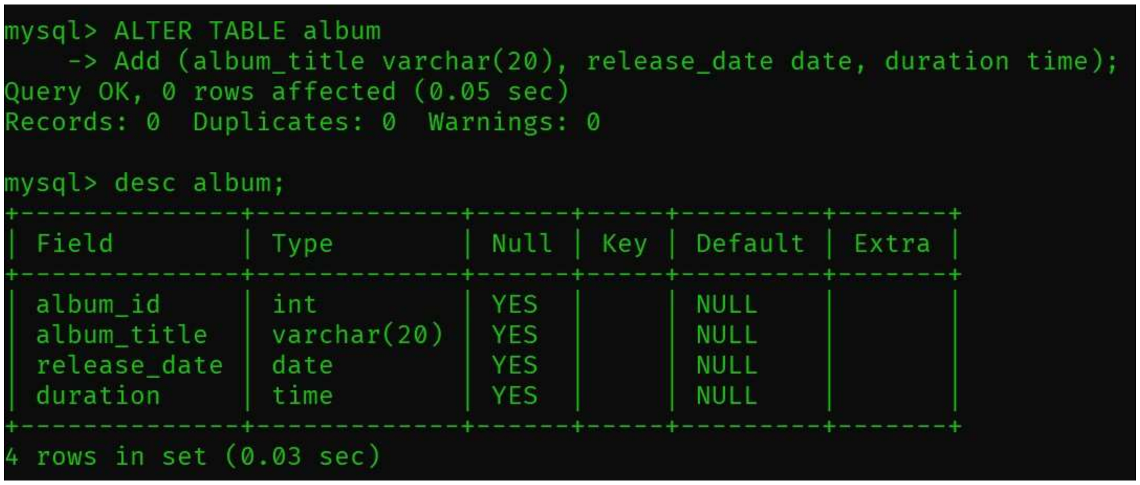
```
mysql> CREATE TABLE User
    -> (user_id int,
    -> username varchar(20),
    -> password varchar(20),
    -> mobileNo char(10),
    -> email varchar(30));
Query OK, 0 rows affected (0.08 sec)

mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| user                |
+---------------------+
1 row in set (0.03 sec)
```

2) ALTER

a) **Adding new columns**

Syntax:
ALTER TABLE *table_name;*
ADD (<*NewColumnName*> <*Data_Type*>(<*size*>), ...n);

Example:
ALTER TABLE *album;*
Add (album_title varchar(20), release_date date, duration time);

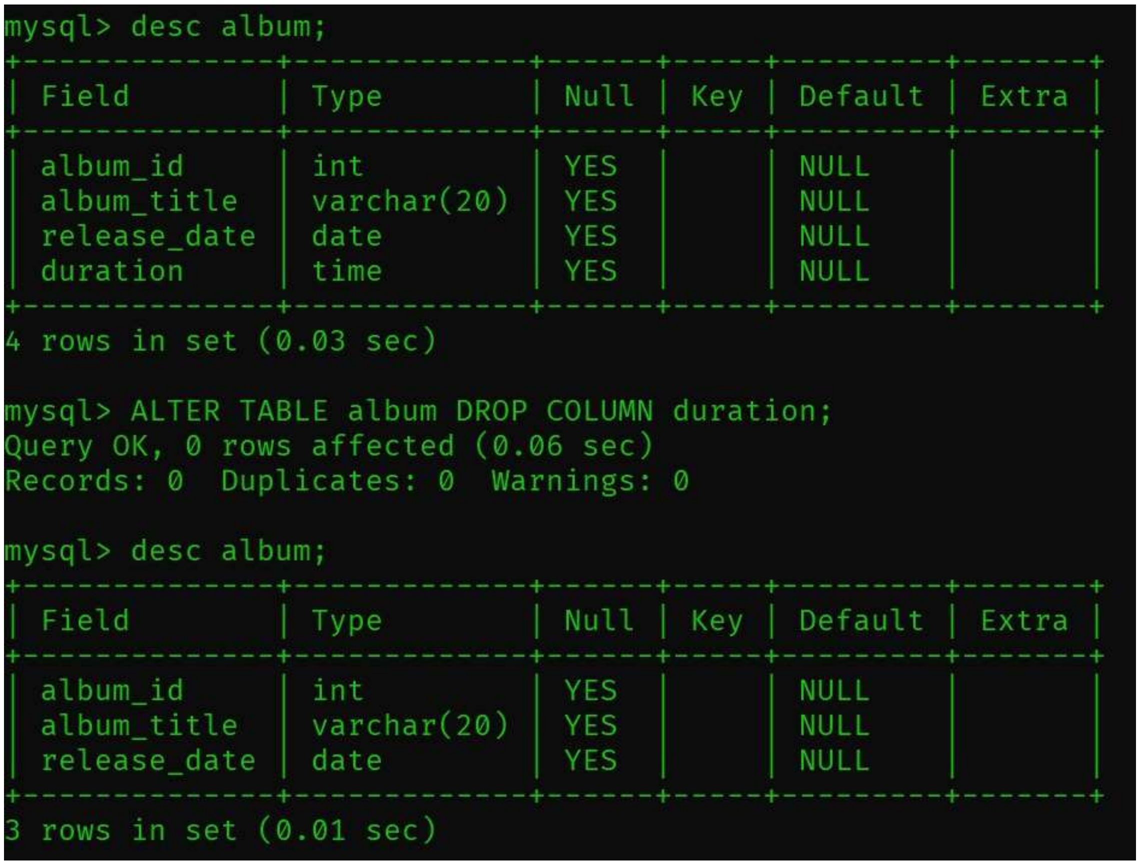


```
mysql> ALTER TABLE album
    -> Add (album_title varchar(20), release_date date, duration time);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(20) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
| duration     | time        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.03 sec)
```

b) **Dropping a column from the table**

Syntax:
ALTER TABLE <*table_name*> DROP COLUMN
<*column_name*>;

Example:
ALTER TABLE *album* DROP COLUMN duration;

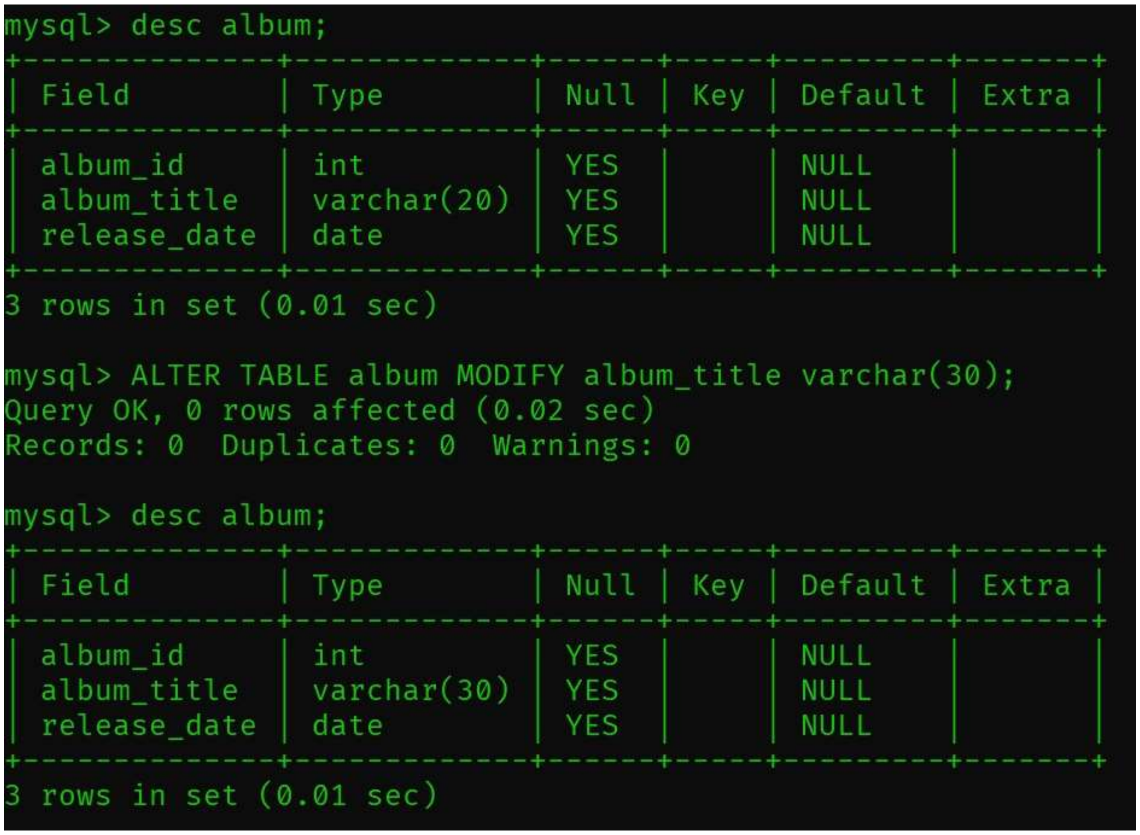


```
mysql> desc album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(20) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
| duration     | time        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.03 sec)

mysql> ALTER TABLE album DROP COLUMN duration;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(20) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

c) **Modifying existing column**

Syntax:
ALTER TABLE <*table_name*> MODIFY <*column_name*> <*NewDataType*>(<*NewSize*>);

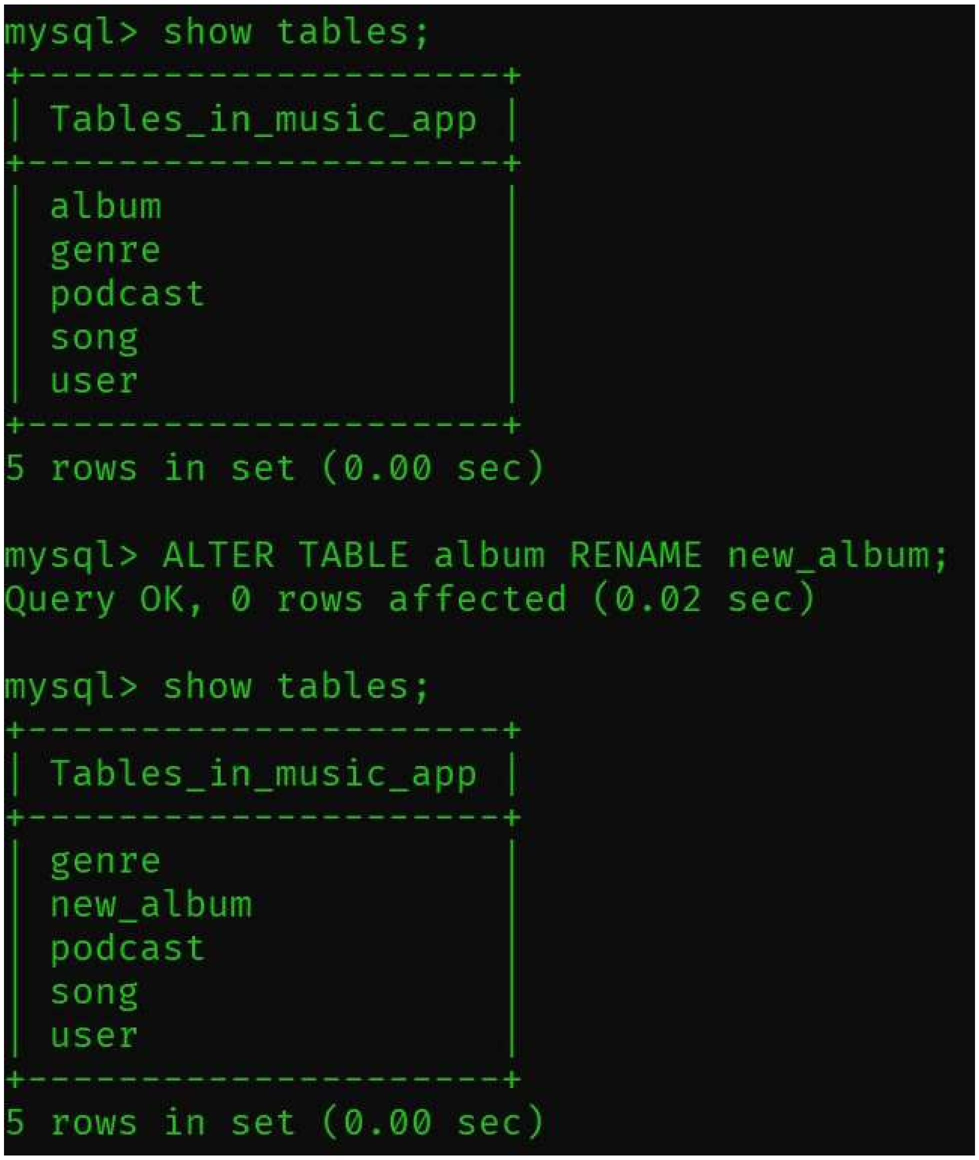
Example:
ALTER TABLE album MODIFY album_title varchar(30);



```
mysql> desc album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(20) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> ALTER TABLE album MODIFY album_title varchar(30);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(30) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

d) **Renaming existing table**

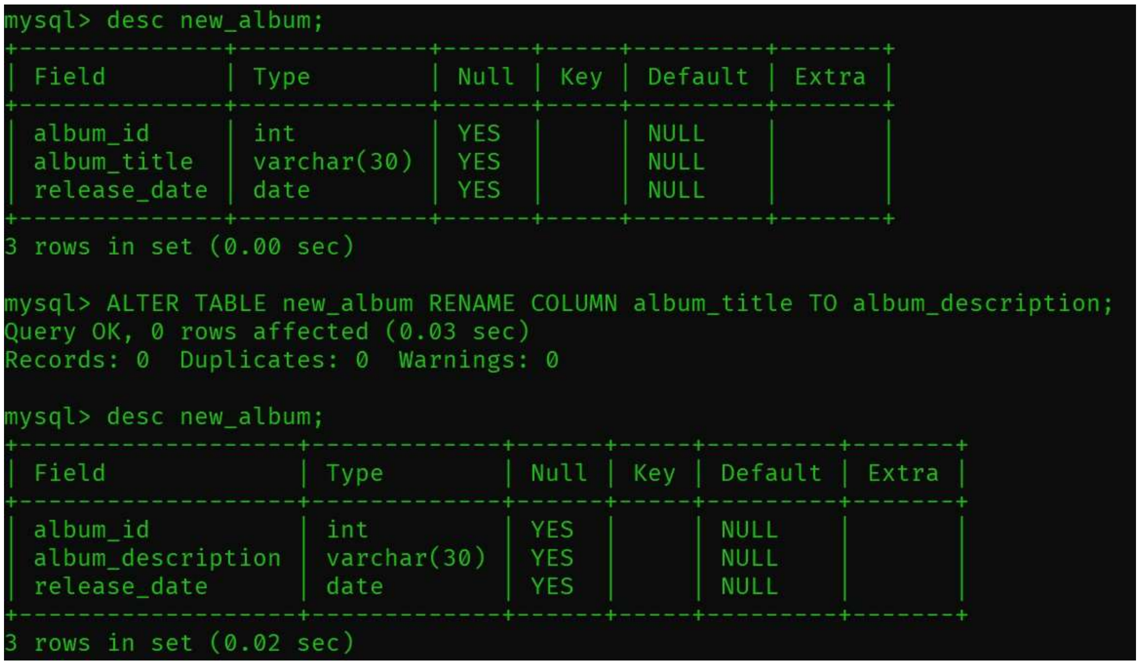
Syntax:
ALTER TABLE <*table_name*> RENAME <*new_table_name*>;
Example:
ALTER TABLE album RENAME new_album;


```
mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| album               |
| genre               |
| podcast             |
| song                |
| user                |
+---------------------+
5 rows in set (0.00 sec)

mysql> ALTER TABLE album RENAME new_album;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| genre               |
| new_album           |
| podcast             |
| song                |
| user                |
+---------------------+
5 rows in set (0.00 sec)
```

e) **Renaming existing column**

Syntax:
ALTER TABLE *table_name* RENAME COLUMN
<*column_name*> TO <*new_column_name*>*;*

Example:
ALTER TABLE new_album RENAME COLUMN album_title TO
album_description;



```
mysql> desc new_album;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| album_id     | int         | YES  |     | NULL    |       |
| album_title  | varchar(30) | YES  |     | NULL    |       |
| release_date | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> ALTER TABLE new_album RENAME COLUMN album_title TO album_description;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc new_album;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| album_id          | int         | YES  |     | NULL    |       |
| album_description | varchar(30) | YES  |     | NULL    |       |
| release_date      | date        | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

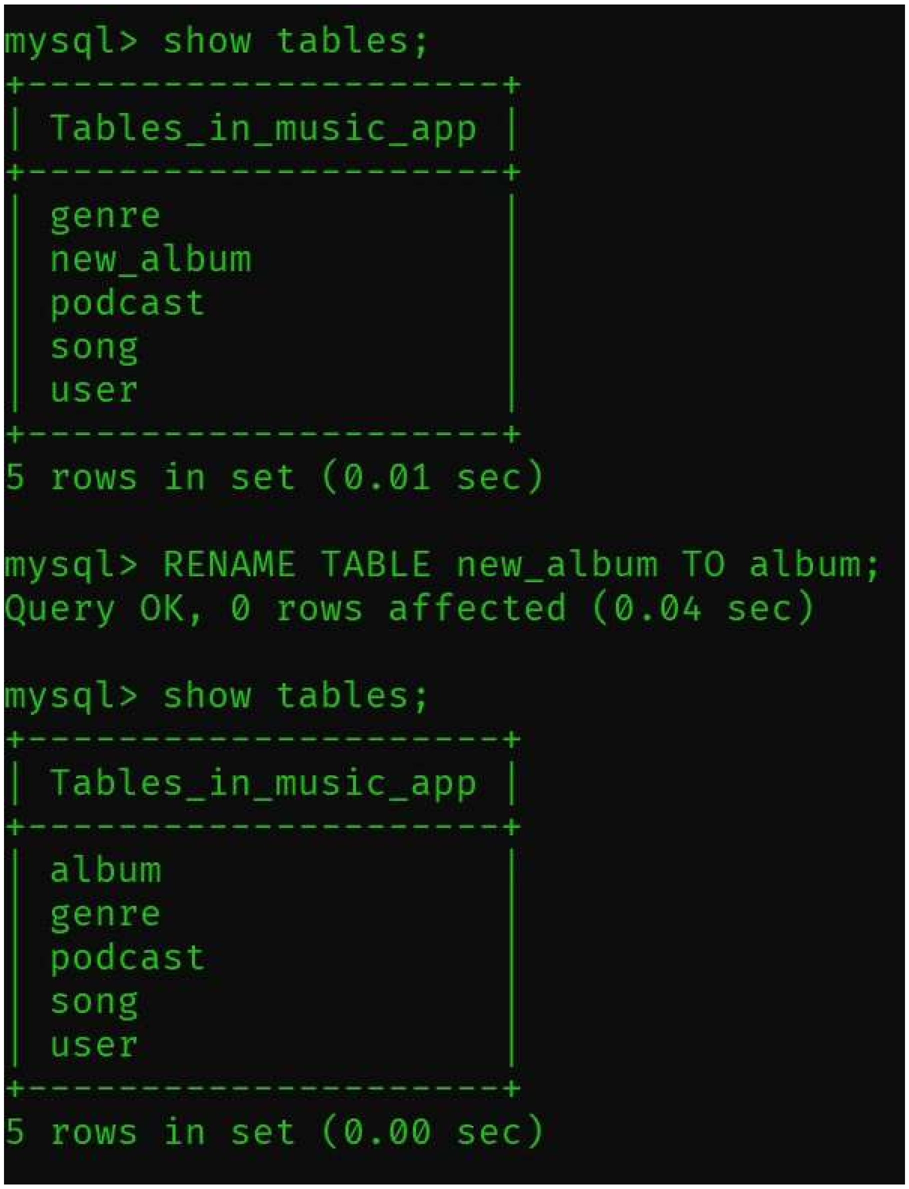
3) RENAME

Syntax:

RENAME TABLE <OldTableName> TO <NewTableName>;

Example:

RENAME TABLE new_album TO album;



```
mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| genre               |
| new_album           |
| podcast             |
| song                |
| user                |
+---------------------+
5 rows in set (0.01 sec)

mysql> RENAME TABLE new_album TO album;
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| album               |
| genre               |
| podcast             |
| song                |
| user                |
+---------------------+
5 rows in set (0.00 sec)
```

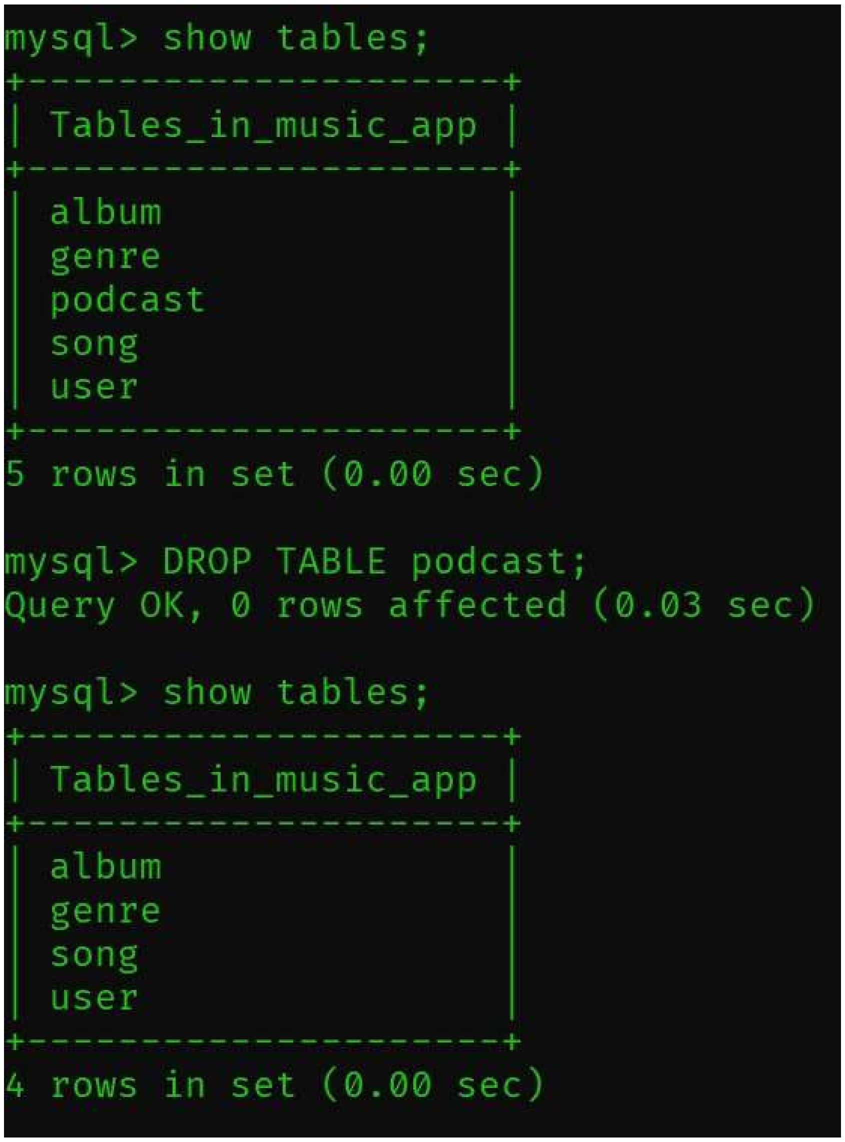
4) DROP

Syntax:

DROP TABLE <table_name>;

Example:

DROP TABLE podcast;


```
mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| album               |
| genre               |
| podcast             |
| song                |
| user                |
+---------------------+
5 rows in set (0.00 sec)

mysql> DROP TABLE podcast;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+---------------------+
| Tables_in_music_app |
+---------------------+
| album               |
| genre               |
| song                |
| user                |
+---------------------+
4 rows in set (0.00 sec)
```

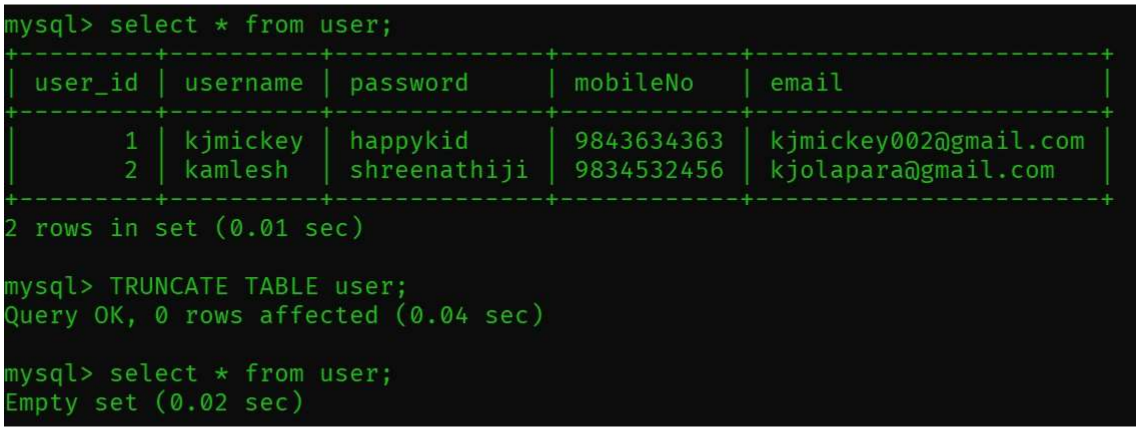
5) TRUNCATE

Syntax:

TRUNCATE TABLE <*table_name*>;

Example:

TRUNCATE TABLE user;



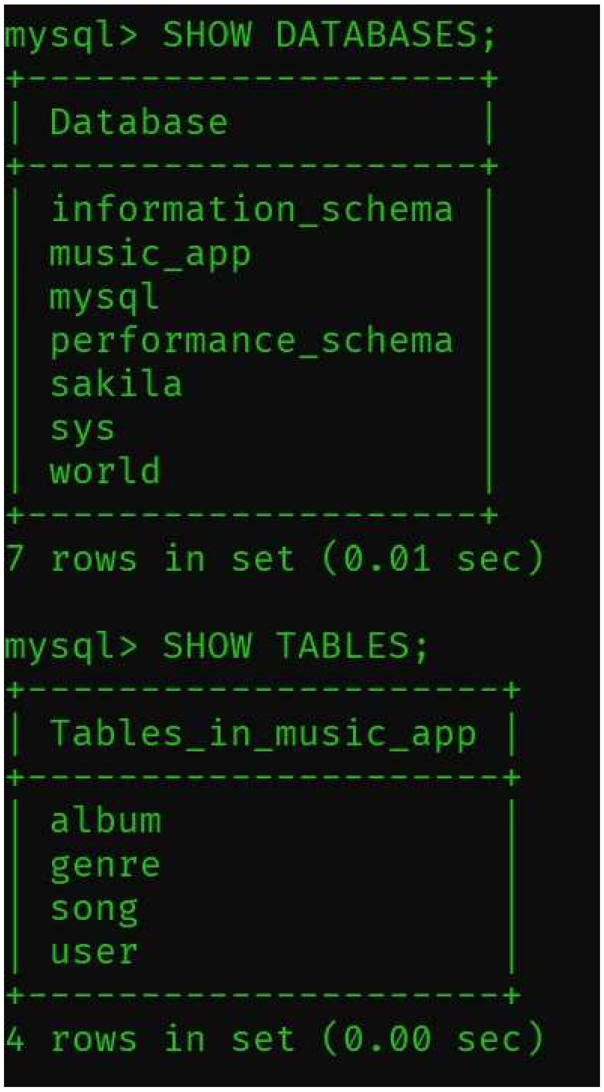
```
mysql> select * from user;
+---------+----------+--------------+------------+-----------------------+
| user_id | username | password     | mobileNo   | email                 |
+---------+----------+--------------+------------+-----------------------+
|       1 | kjmickey | happykid     | 9843634363 | kjmickey002@gmail.com |
|       2 | kamlesh  | shreenathiji | 9834532456 | kjolapara@gmail.com   |
+---------+----------+--------------+------------+-----------------------+
2 rows in set (0.01 sec)

mysql> TRUNCATE TABLE user;
Query OK, 0 rows affected (0.04 sec)

mysql> select * from user;
Empty set (0.02 sec)
```

6) SHOW

<u>Syntax:</u>

SHOW DATABASES;

SHOW TABLES;



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| music_app          |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
7 rows in set (0.01 sec)

mysql> SHOW TABLES;
+---------------------+
| Tables_in_music_app |
+---------------------+
| album               |
| genre               |
| song                |
| user                |
+---------------------+
4 rows in set (0.00 sec)
```

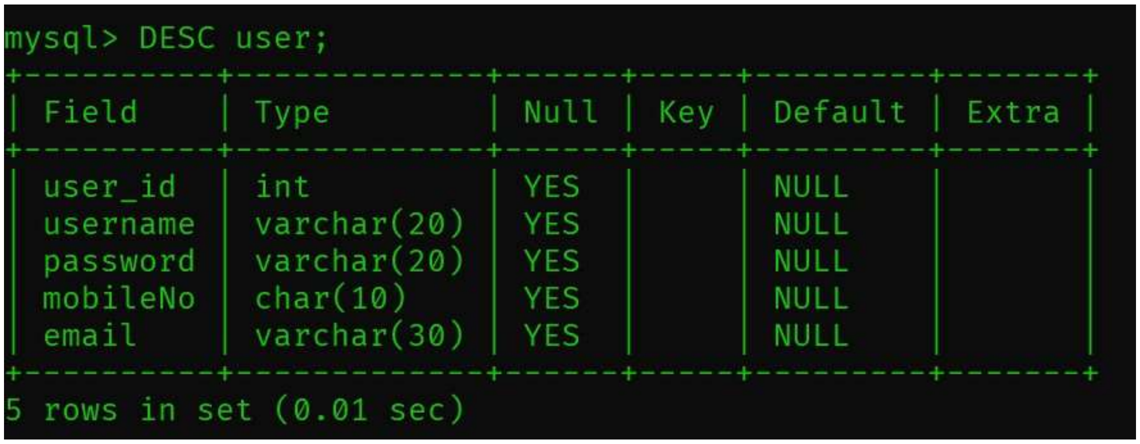
7) DESCRIBE

<u>Syntax:</u>

DESCRIBE <*table_name*>; / DESC <*table_name*>;

<u>Example:</u>

DESC user;



```
mysql> DESC user;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| user_id  | int         | YES  |     | NULL    |       |
| username | varchar(20) | YES  |     | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
| mobileNo | char(10)    | YES  |     | NULL    |       |
| email    | varchar(30) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

DDL Queries

1) INSERT

This command is used insert data in the tables that are created via the DML commands

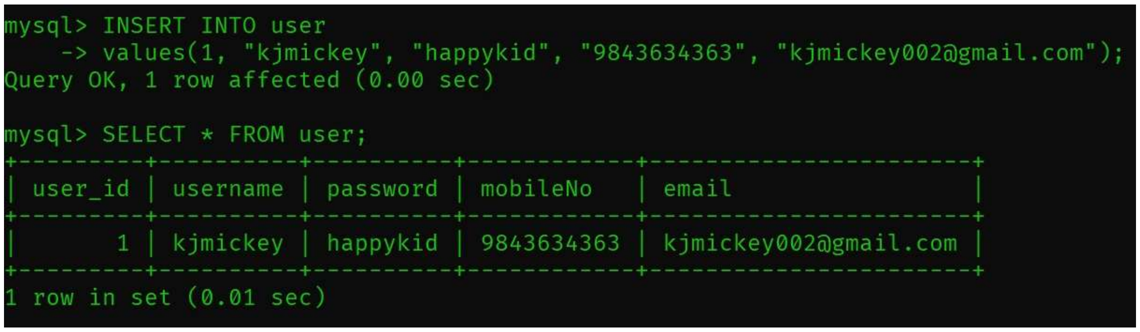
<u>Syntax:</u>

INSERT INTO <*table_name*> (<*column1*>, <*column2*>, ...)

VALUES ("value1", "value2", ...);

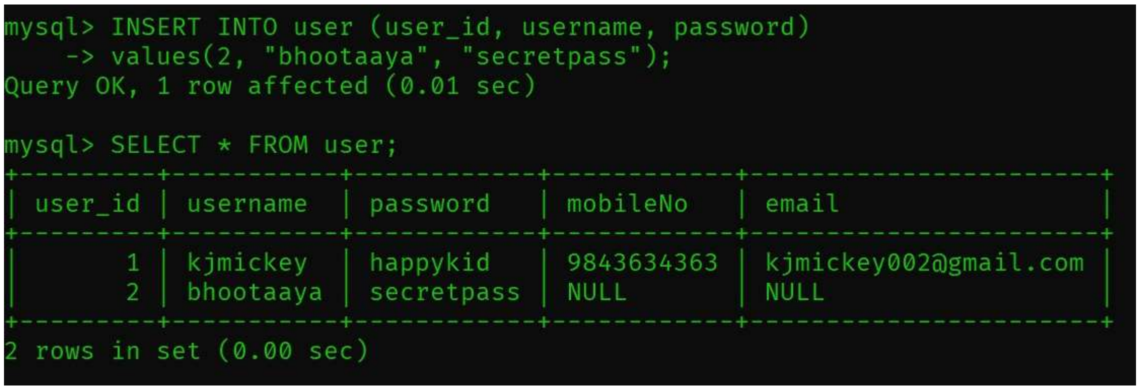
<u>Examples:</u>

a. *Directly inserting* INSERT INTO user values(1, "kjmickey", "happykid", "9843634363", "kjmickey002@gmail.com");



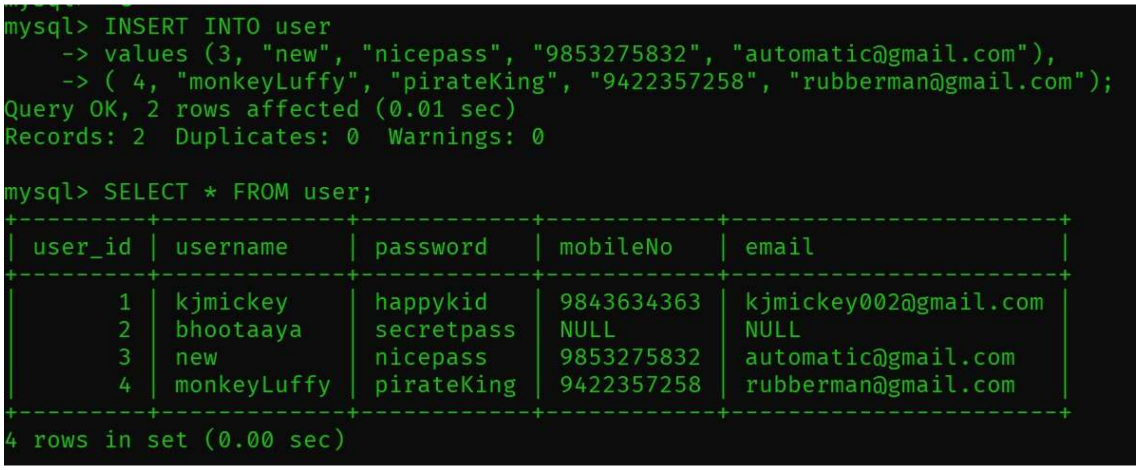
```
mysql> INSERT INTO user
    -> values(1, "kjmickey", "happykid", "9843634363", "kjmickey002@gmail.com");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM user;
+---------+----------+----------+------------+-----------------------+
| user_id | username | password | mobileNo   | email                 |
+---------+----------+----------+------------+-----------------------+
|       1 | kjmickey | happykid | 9843634363 | kjmickey002@gmail.com |
+---------+----------+----------+------------+-----------------------+
1 row in set (0.01 sec)
```

b. *Inserting only some rows*
INSERT INTO user (user_id, username, password) values(2, "bhootaaya", "secretpass");



```
mysql> INSERT INTO user (user_id, username, password)
    -> values(2, "bhootaaya", "secretpass");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM user;
+---------+-----------+------------+------------+-----------------------+
| user_id | username  | password   | mobileNo   | email                 |
+---------+-----------+------------+------------+-----------------------+
|       1 | kjmickey  | happykid   | 9843634363 | kjmickey002@gmail.com |
|       2 | bhootaaya | secretpass | NULL       | NULL                  |
+---------+-----------+------------+------------+-----------------------+
2 rows in set (0.00 sec)
```

c. *Inserting multiple rows at once* INSERT INTO user
values (3, "new", "nicepass", "9853275832", "automatic@gmail.com"),
(4, "monkeyLuffy", "pirateKing", "9422357258", "rubberman@gmail.com");



```
mysql> INSERT INTO user
    -> values (3, "new", "nicepass", "9853275832", "automatic@gmail.com"),
    -> ( 4, "monkeyLuffy", "pirateKing", "9422357258", "rubberman@gmail.com");
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM user;
+---------+-------------+------------+------------+-----------------------+
| user_id | username    | password   | mobileNo   | email                 |
+---------+-------------+------------+------------+-----------------------+
|       1 | kjmickey    | happykid   | 9843634363 | kjmickey002@gmail.com |
|       2 | bhootaaya   | secretpass | NULL       | NULL                  |
|       3 | new         | nicepass   | 9853275832 | automatic@gmail.com   |
|       4 | monkeyLuffy | pirateKing | 9422357258 | rubberman@gmail.com   |
+---------+-------------+------------+------------+-----------------------+
4 rows in set (0.00 sec)
```

2) SELECT

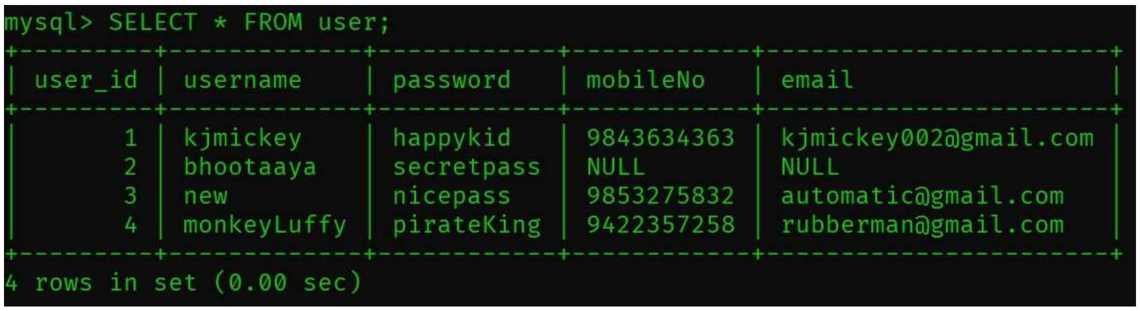
The SELECT statement is used to select data from a database.

Syntax:

SELECT * FROM *table_name*;

Example:

SELECT * FROM user;



```
mysql> SELECT * FROM user;
+---------+-------------+------------+------------+-----------------------+
| user_id | username    | password   | mobileNo   | email                 |
+---------+-------------+------------+------------+-----------------------+
|       1 | kjmickey    | happykid   | 9843634363 | kjmickey002@gmail.com |
|       2 | bhootaaya   | secretpass | NULL       | NULL                  |
|       3 | new         | nicepass   | 9853275832 | automatic@gmail.com   |
|       4 | monkeyLuffy | pirateKing | 9422357258 | rubberman@gmail.com   |
+---------+-------------+------------+------------+-----------------------+
4 rows in set (0.00 sec)
```

3) UPDATE

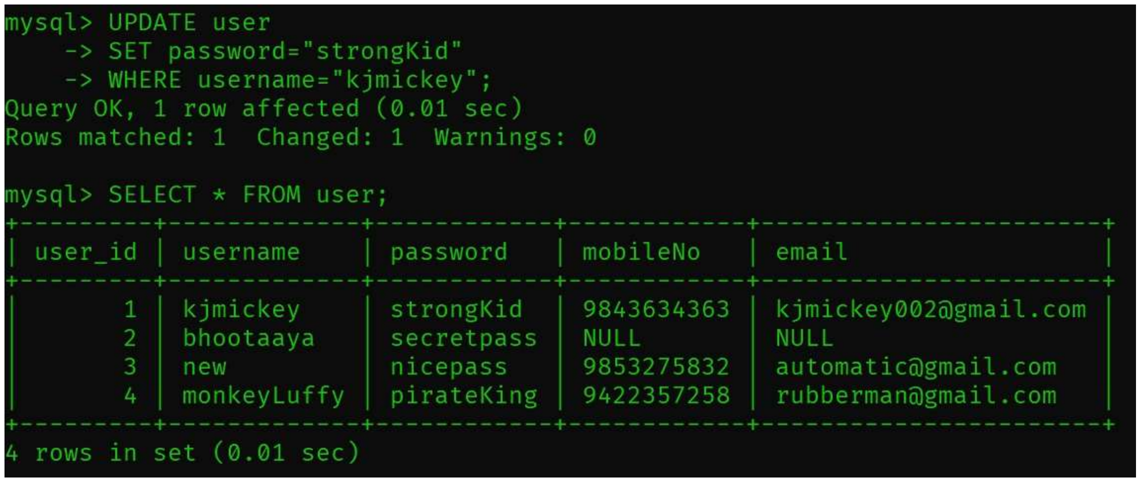
The UPDATE statement is used to update existing records in a table.

Syntax:

UPDATE *table_name*
SET *column1=value1*, *column2=value2,...*
WHERE *some_column=some_value*;
Example:

a. *Update a single field*

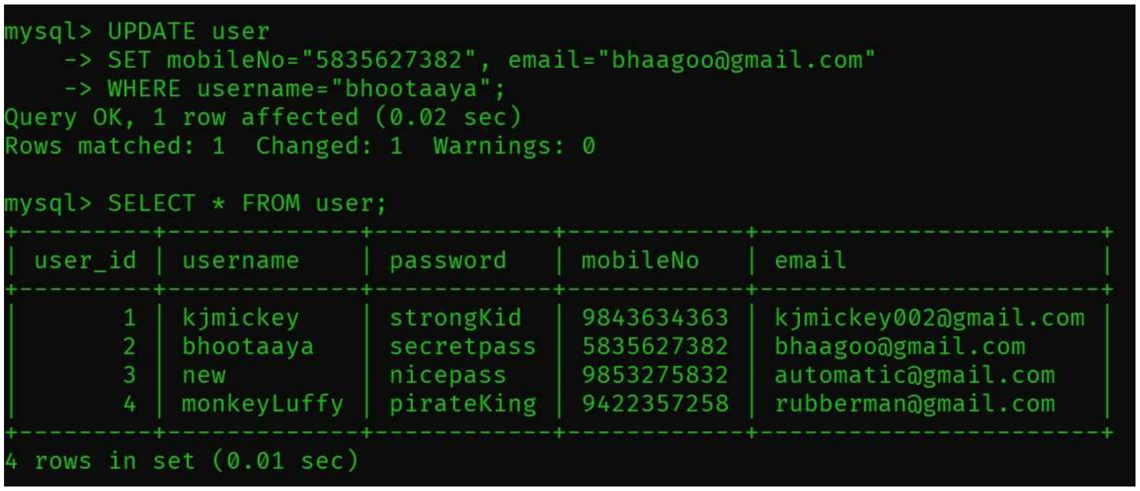
UPDATE user
SET password="strongKid"
WHERE username="kjmickey";



```
mysql> UPDATE user
    -> SET password="strongKid"
    -> WHERE username="kjmickey";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM user;
+---------+-------------+------------+------------+------------------------+
| user_id | username    | password   | mobileNo   | email                  |
+---------+-------------+------------+------------+------------------------+
|       1 | kjmickey    | strongKid  | 9843634363 | kjmickey002@gmail.com  |
|       2 | bhootaaya   | secretpass | NULL       | NULL                   |
|       3 | new         | nicepass   | 9853275832 | automatic@gmail.com    |
|       4 | monkeyLuffy | pirateKing | 9422357258 | rubberman@gmail.com    |
+---------+-------------+------------+------------+------------------------+
4 rows in set (0.01 sec)
```

b. *Update multiple fields at once*

UPDATE user
SET mobileNo="5835627382", email="bhaagoo@gmail.com"
WHERE username="bhootaaya";



```
mysql> UPDATE user
    -> SET mobileNo="5835627382", email="bhaagoo@gmail.com"
    -> WHERE username="bhootaaya";
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM user;
+---------+-------------+------------+------------+------------------------+
| user_id | username    | password   | mobileNo   | email                  |
+---------+-------------+------------+------------+------------------------+
|       1 | kjmickey    | strongKid  | 9843634363 | kjmickey002@gmail.com  |
|       2 | bhootaaya   | secretpass | 5835627382 | bhaagoo@gmail.com      |
|       3 | new         | nicepass   | 9853275832 | automatic@gmail.com    |
|       4 | monkeyLuffy | pirateKing | 9422357258 | rubberman@gmail.com    |
+---------+-------------+------------+------------+------------------------+
4 rows in set (0.01 sec)
```

4) DELETE

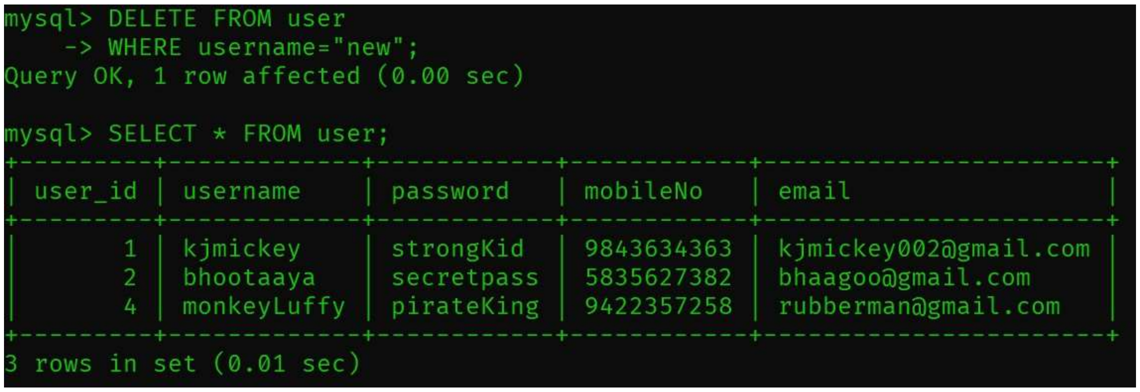
This command removes one or more records from a table according to specified conditions.

Syntax:

DELETE FROM <*table_name*>
WHERE *some_column*=*some_value*;

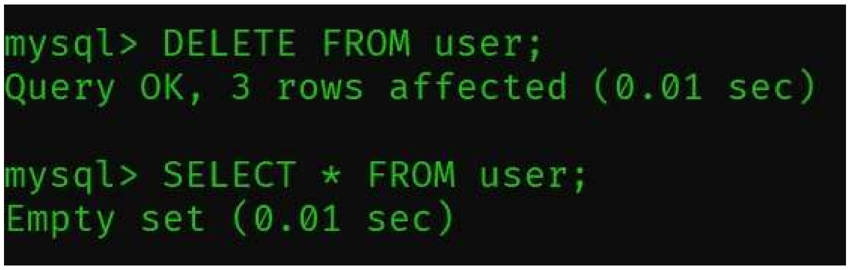
Example:

a. *Deletes a specific row with the condition*
   DELETE FROM user
   WHERE username="new";



```
mysql> DELETE FROM user
    -> WHERE username="new";
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM user;
+---------+-------------+------------+------------+-----------------------+
| user_id | username    | password   | mobileNo   | email                 |
+---------+-------------+------------+------------+-----------------------+
|       1 | kjmickey    | strongKid  | 9843634363 | kjmickey002@gmail.com |
|       2 | bhootaaya   | secretpass | 5835627382 | bhaagoo@gmail.com     |
|       4 | monkeyLuffy | pirateKing | 9422357258 | rubberman@gmail.com   |
+---------+-------------+------------+------------+-----------------------+
3 rows in set (0.01 sec)
```

b. *Deletes all the rows*
   DELETE FROM user;



```
mysql> DELETE FROM user;
Query OK, 3 rows affected (0.01 sec)

mysql> SELECT * FROM user;
Empty set (0.01 sec)
```

Conclusion:

Data definition Language (DDL) is used to create the schema or the pattern of the database for the project and used Data Manipulation Language (DML) to fill the data and change it according to the need.