Activity No. 2	
SQL Data Manipulation Language Commands	
Course Code: CPE011	**Program:** BSCPE21-S3
Course Title: Database Management System	**Date Performed: 08/25/2022**
Section: S3	**Date Submitted: 08/26/2022**
Name: Aaron Martin Parallag Caro Christian Efa	**Instructor:** Dr. Jonathan Taylar

1. Objective(s):

This activity aims to introduce the Structured Query Language (SQL) using the Data Manipulation Language (DML) commands in a MySQL Database

2. Intended Learning Outcomes (ILOs):

The students should be able to:
2.1 Use the command line interface to perform SQL Data Manipulation Commands.
2.2 Insert data to a MySQL Database
2.3 Update the contents of a database table
2.4 Delete the contents of database table

3. Discussion:

Recall that in a relational database, we organize our data in terms of columns and rows or fields and tuples. In the previous activity, we have learned to create a database and its schema by create tables in it through the SQL Data Definition Language (DDL) commands of MySQL.

In order to insert or modify any data in the table as well as to read/retrieve values that you inserted to the database, you will need to use the SQL Data Manipulation Language also called DML. This activity will cover the four fundamental SQL DML commands SELECT, INSERT, UPDATE, and DELETE which is essential in any database connected system.

Some of the SQL DML Commands will require additional keywords or clauses and inputs in order for you to get your desired output.

Inserting Data to a MySQL table

To insert data into MySQL table, you would need to use SQL INSERT INTO command. Here is generic SQL syntax of INSERT INTO command to insert data into MySQL table:

```
INSERT INTO table_name ( field1, field2,...fieldN )
    VALUES
    ( value1, value2,...valueN );
```

Note that the field1, field2 are attributes in the table. The value1 and value2 are the instances of the attributes. This indicates that the fields must follow the same order as values.

An alternative way to insert data without using field names is by using this syntax:

```
INSERT INTO table_name
    VALUES
    ( value1, value2,...valueN );
```

This shows that the data does not necessarily require an indication of the respective attributes. This command inserts data to whichever column is present in the database, as long as the data type matches.

Viewing the data present in a database

The SELECT statement is used to select data from a database. The result is stored in a result table, called the result-set. To select all the data from a particular table, the following syntax can be used:

SELECT * FROM table_name;

The following syntax, on the other hand, selects only the specific columns of the database.

SELECT column_name,column_name FROM table_name;

The SQL UPDATE Statement

The UPDATE statement is used to update existing records in a table. The syntax for this command is as follows:

```
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

SQL DELETE Statement

The DELETE statement is used to delete records in a table. The syntax is as follows:

```
DELETE FROM table_name
WHERE some_column=some_value;
```

We will again connect to the MySQL Database through the Command Prompt and access the mysql.exe program located in the XAMPP folder. If you don't recall how to perform this task, refer to the discussion in activity 1.

4. Materials and Equipment:

Desktop Computer
Windows Operating System
XAMPP Application

5. Procedure

Database Schema

The image below shows the tables in the database that will be used in this activity. You may use your previous database from activity 1 or recreate the database to match the structure below.

```
MariaDB [db1]> describe vehicles;
+-----------------+----------+------+-----+---------+-------+
| Field           | Type     | Null | Key | Default | Extra |
+-----------------+----------+------+-----+---------+-------+
| car_maker       | char(20) | YES  |     | NULL    |       |
| car_model       | char(20) | YES  |     | NULL    |       |
| number_of_doors | int(11)  | YES  |     | NULL    |       |
+-----------------+----------+------+-----+---------+-------+
3 rows in set (0.10 sec)

MariaDB [db1]> describe drivers;
+---------------+------------+------+-----+---------+-------+
| Field         | Type       | Null | Key | Default | Extra |
+---------------+------------+------+-----+---------+-------+
| id            | int(11)    | YES  |     | NULL    |       |
| first_name    | char(50)   | YES  |     | NULL    |       |
| last_name     | char(50)   | YES  |     | NULL    |       |
| age           | int(3)     | YES  |     | NULL    |       |
| address       | char(100)  | YES  |     | NULL    |       |
| years_driving | int(3)     | YES  |     | NULL    |       |
| status        | tinyint(1) | YES  |     | NULL    |       |
+---------------+------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

MariaDB [db1]>
```

Part I

1. Use the syntax from the discussion to populate each table with 5 entries each.
2. Display all the values in each table.
3. Display only the ID, First Name, and Last name of the drivers.
4. Display the car makers and the car models in the vehicles table.
5. Show the screenshot of each set of results with your observation below each in Section 6 Database Output.

Part II

1. Use the UPDATE command to change a row value (ex. First Name, age) in each table.
2. Display the values in each table.
3. Show the screenshot of the newly updated tables with your observation below each in Section 6 Database Output.

Part III

1. Delete one row from each table.
2. Show the screenshot of the newly updated tables with your observation below each in Section 6 Database Output.

Note: Do not drop the database or any tables.

Part IV

1. Insert two rows to the drivers table with only age and address.
2. Run the command ***SELECT first_name, last_name FROM drivers;***
3. Run the command ***DELETE FROM drivers WHERE first_name=NULL;***
   Note: Observe the rows affected if there were rows deleted.
4. Run the command ***DELETE FROM drivers WHERE first_name is NULL;***
5. Run the command ***SELECT first_name, last_name FROM drivers;***
6. Add a new column in drivers called license_number with data type char(13)
7. Run the command ***DELETE FROM drivers WHERE license_number is NULL;***
8. Update one of the rows in the drivers table with a license_number (ex. "N014-123-M01")

9. Run the command ***SELECT * FROM drivers;***
10. Take a screenshot of the code snippets of resulting tables from SELECT commands with your observation below each in Section 6 Database Output.

6. Database Output:

Copy screenshot(s) of your output with observations after completing the procedures provided in Part 5.

Part 1.

```
MariaDB [driversdb_caro]> describe vehicles;
+-----------------+-----------+------+-----+---------+-------+
| Field           | Type      | Null | Key | Default | Extra |
+-----------------+-----------+------+-----+---------+-------+
| car_maker       | char(100) | YES  |     | NULL    |       |
| car_model       | char(100) | YES  |     | NULL    |       |
| number_of_doors | int(11)   | YES  |     | NULL    |       |
+-----------------+-----------+------+-----+---------+-------+
3 rows in set (0.014 sec)

MariaDB [driversdb_caro]> describe drivers;
+---------------+-----------+------+-----+---------+-------+
| Field         | Type      | Null | Key | Default | Extra |
+---------------+-----------+------+-----+---------+-------+
| id            | int(11)   | YES  |     | NULL    |       |
| first_name    | char(100) | YES  |     | NULL    |       |
| last_name     | char(100) | YES  |     | NULL    |       |
| Age           | int(11)   | YES  |     | NULL    |       |
| Address       | char(100) | YES  |     | NULL    |       |
| Years_driving | int(3)    | YES  |     | NULL    |       |
| status        | char(100) | YES  |     | NULL    |       |
+---------------+-----------+------+-----+---------+-------+
7 rows in set (0.009 sec)
```

Use the syntax from the discussion to populate each table with 5 entries each.

```
MariaDB [driversdb_caro]> INSERT INTO drivers(id,first_name,last_name,Age,Address,Years_driving,status)
    -> VALUES(2,'Alan','Walker',21,'SinCity',8,'Active'),
    -> (3,'jane','Doe',21,'CrimeAlley',2,'Inactive'),
    -> (4,'Max','Anne',21,'BlackWell',3,'Active'),
    -> (5,'Delsin','Roe',24,'BlueCreek',5,'Active');
```

```
MariaDB [driversdb_caro]> INSERT INTO vehicles(car_maker,car_model,number_of_doors)
    -> VALUES('Tesla','ModelS','4'),
    -> ('Subaru','WRX','4'),
    -> ('Toyota','Chasers','4'),
    -> ('Isuzu','D-MAX','4'),
    -> ('Mitsubishi','Lancer','4');
```

Display all the values in each table.

```
MariaDB [driversdb_caro]> SELECT * FROM vehicles;
+------------+-----------+-----------------+
| car_maker  | car_model | number_of_doors |
+------------+-----------+-----------------+
| Tesla      | ModelS    |               4 |
| Subaru     | WRX       |               4 |
| Toyota     | Chasers   |               4 |
| Isuzu      | D-MAX     |               4 |
| Mitsubishi | Lancer    |               4 |
+------------+-----------+-----------------+
5 rows in set (0.000 sec)

MariaDB [driversdb_caro]> SELECT * FROM drivers;
+------+------------+-----------+------+------------+---------------+----------+
| id   | first_name | last_name | Age  | Address    | Years_driving | status   |
+------+------------+-----------+------+------------+---------------+----------+
|    1 | John       | Marcus    |   19 | BloodHaven |             4 | Active   |
|    2 | Alan       | Walker    |   21 | SinCity    |             8 | Active   |
|    3 | jane       | Doe       |   21 | CrimeAlley |             2 | Inactive |
|    4 | Max        | Anne      |   21 | BlackWell  |             3 | Active   |
|    5 | Delsin     | Roe       |   24 | BlueCreek  |             5 | Active   |
+------+------------+-----------+------+------------+---------------+----------+
5 rows in set (0.001 sec)
```

Display only the ID, First Name, and Last name of the drivers.

```
MariaDB [driversdb_caro]> SELECT id,first_name,last_name FROM drivers;
+------+------------+-----------+
| id   | first_name | last_name |
+------+------------+-----------+
|    1 | John       | Marcus    |
|    2 | Alan       | Walker    |
|    3 | jane       | Doe       |
|    4 | Max        | Anne      |
|    5 | Delsin     | Roe       |
+------+------------+-----------+
5 rows in set (0.001 sec)
```

Display the car makers and the car models in the vehicles table.

```
MariaDB [driversdb_caro]> SELECT car_maker,car_model FROM vehicles;
+------------+-----------+
| car_maker  | car_model |
+------------+-----------+
| Tesla      | ModelS    |
| Subaru     | WRX       |
| Toyota     | Chasers   |
| Isuzu      | D-MAX     |
| Mitsubishi | Lancer    |
+------------+-----------+
5 rows in set (0.000 sec)
```

Observation: From what I can see I was able to fill the fields with all kinds of information depending on their data type and I can show which information I can display with some sql commands

Part 2.

Use the UPDATE command to change a row value (ex. First Name, age) in each table.

```
MariaDB [driversdb_caro]> UPDATE drivers
    -> SET first_name = 'Kane',
    -> Age = 20
    -> WHERE id = 1;
Query OK, 1 row affected (0.008 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [driversdb_caro]> UPDATE vehicles
    -> SET car_maker = 'Honda',
    -> car_model = 'Integra'
    -> where car_maker = 'Tesla';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Display the values in each table.

```
MariaDB [driversdb_caro]> SELECT * FROM drivers;
+------+------------+-----------+------+------------+---------------+----------+
| id   | first_name | last_name | Age  | Address    | Years_driving | status   |
+------+------------+-----------+------+------------+---------------+----------+
|    1 | Kane       | Marcus    |   20 | BloodHaven |             4 | Active   |
|    2 | Alan       | Walker    |   21 | SinCity    |             8 | Active   |
|    3 | jane       | Doe       |   21 | CrimeAlley |             2 | Inactive |
|    4 | Max        | Anne      |   21 | BlackWell  |             3 | Active   |
|    5 | Delsin     | Roe       |   24 | BlueCreek  |             5 | Active   |
+------+------------+-----------+------+------------+---------------+----------+
5 rows in set (0.001 sec)
```

```
MariaDB [driversdb_caro]> select * from vehicles;
+------------+-----------+-----------------+
| car_maker  | car_model | number_of_doors |
+------------+-----------+-----------------+
| Honda      | Integra   |               4 |
| Subaru     | WRX       |               4 |
| Toyota     | Chasers   |               4 |
| Isuzu      | D-MAX     |               4 |
| Mitsubishi | Lancer    |               4 |
+------------+-----------+-----------------+
5 rows in set (0.000 sec)
```

Observation: When using an update command I will know it will work when the values in rows match and Changed turns into a value higher than zero.

Part3.

Delete one row from each table.

```
MariaDB [driversdb_caro]> DELETE FROM drivers
    -> WHERE id=1;
Query OK, 1 row affected (0.005 sec)
```

```
MariaDB [driversdb_caro]> DELETE FROM vehicles
    -> WHERE car_maker= 'Subaru';
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [driversdb_caro]> SELECT * FROM drivers;
+------+------------+-----------+------+------------+---------------+----------+
| id   | first_name | last_name | Age  | Address    | Years_driving | status   |
+------+------------+-----------+------+------------+---------------+----------+
|    2 | Alan       | Walker    |   21 | SinCity    |             8 | Active   |
|    3 | jane       | Doe       |   21 | CrimeAlley |             2 | Inactive |
|    4 | Max        | Anne      |   21 | BlackWell  |             3 | Active   |
|    5 | Delsin     | Roe       |   24 | BlueCreek  |             5 | Active   |
+------+------------+-----------+------+------------+---------------+----------+
4 rows in set (0.000 sec)

MariaDB [driversdb_caro]> SELECT * FROM vehicles;
+------------+-----------+-----------------+
| car_maker  | car_model | number_of_doors |
+------------+-----------+-----------------+
| Honda      | Integra   |               4 |
| Toyota     | Chasers   |               4 |
| Isuzu      | D-MAX     |               4 |
| Mitsubishi | Lancer    |               4 |
+------------+-----------+-----------------+
4 rows in set (0.001 sec)
```

Observation: When using a DELETE command I will know it will work when it said query ok and it stated that one row is affected as it is clearly displayed.

Part4.

Insert two rows to the drivers table with only age and address.

```
MariaDB [driversdb_caro]> INSERT INTO drivers(Age,Address)
    -> VALUES(30,'Louise Ville');
Query OK, 1 row affected (0.003 sec)
```

Run the command ***SELECT first_name, last_name FROM drivers;***

```
MariaDB [driversdb_caro]> SELECT first_name,last_name FROM drivers;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Alan       | Walker    |
| jane       | Doe       |
| Max        | Anne      |
| Delsin     | Roe       |
| NULL       | NULL      |
+------------+-----------+
5 rows in set (0.000 sec)
```

Run the command ***DELETE FROM drivers WHERE first_name=NULL;***

Note: Observe the rows affected if there were rows deleted.

```
MariaDB [driversdb_caro]> DELETE FROM drivers WHERE first_name=NULL;
Query OK, 0 rows affected (0.001 sec)
```

Run the command ***DELETE FROM drivers WHERE first_name is NULL;***

```
MariaDB [driversdb_caro]> DELETE FROM drivers WHERE first_name is NULL;
Query OK, 1 row affected (0.005 sec)
```

Run the command ***SELECT first_name, last_name FROM drivers;***

```
MariaDB [driversdb_caro]> SELECT first_name, last_name FROM drivers;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Alan       | Walker    |
| jane       | Doe       |
| Max        | Anne      |
| Delsin     | Roe       |
+------------+-----------+
4 rows in set (0.001 sec)
```

Add a new column in drivers called **license_number** with data type **char(13)**

```
MariaDB [driversdb_caro]> ALTER TABLE drivers ADD license_number char(13);
Query OK, 0 rows affected (0.021 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Run the command ***DELETE FROM drivers WHERE license_number is NULL;***

```
MariaDB [driversdb_caro]> DELETE FROM drivers WHERE license_number is NULL;
Query OK, 4 rows affected (0.003 sec)
```

Update one of the rows in the drivers table with a license_number (ex. "N014-123-M01")

```
MariaDB [driversdb_caro]> UPDATE drivers
    -> SET license_numer = 'N014-123-M01'
    -> WHERE id = NULL;
ERROR 1054 (42S22): Unknown column 'license_numer' in 'field list'
MariaDB [driversdb_caro]> UPDATE drivers
    -> SET license_numer = 'N014-123-M01'
    -> WHERE id is NULL;
ERROR 1054 (42S22): Unknown column 'license_numer' in 'field list'
MariaDB [driversdb_caro]> SELECT * FROM drivers;
Empty set (0.000 sec)
```

Observation: I was able to input values in specified field lists then delete rows with a specified values so when the license_number column was added and when we have to delete all with null values in the licence_ number I knew that we would get an empty set.

7. Supplementary Activity

Questions

1. **What happens if you insert an exact duplicate of any record that is already in your current database table?**
   Ex. > INSERT INTO drivers VALUES(1,"James","Nicolas",20,"5 Peace St. QC",3,true);
   > INSERT INTO drivers VALUES(1,"James","Nichelistine",20,"5 Peace St. QC",3,true)
   You can try the result. Is the output allowable in a real environment? Why do you think the output was what you saw?
   If duplicate records are not removed, then data processing will fail.

2. **Given that you ran your SQL Command in the example in Question 1.**

id	first_name	last_name	age	address	years_driving	status
1	Mary	Nichelistine	20	5 Peace St. QC	3	1
2	Roger	Michael	45	8 Orange St. Marikina	10	0
1	James	Nichelistine	20	5 Peace St. QC	3	1
1	James	Nichelistine	20	5 Peace St. QC	3	1

   What happens if you try to delete one of the rows with the redundant id? (Ex. DELETE FROM drivers WHERE id=1;) What do you think should the solution be against this kind of problem?
   if we remove information about the one of the row with redundant ID, all of the branch information disappears.

3. **In Part IV of the Procedure section, what do you think is the difference between using an equals sign and the is keyword when checking for NULL?**
   According to the documentation, the only distinction between the two operators is how they treat null. The evaluation of NULL on either side of = returns false. The obvious change that we notice is that when equal Sign is used the rows was not affected but when the key word [is] is used the row was affected.

   Additional Reference: https://dev.mysql.com/doc/refman/8.0/en/working-with-null.html

4. **Following from the part IV tasks where you've added a new column called license_number. What would happen if you inserted "N014-123-M012XD3"? Why do you think that was the output?**
   The data will be added to a new row with NULL values for the other fields.

5. **Why is it necessary to introduce a WHERE clause when updating and deleting data?**

The WHERE clause is used to apply conditions and filter out results while retrieving or manipulating any data from the database.

8. Conclusion

To alter or manipulate the data records that are present in the database tables, DML commands are employed. Data insertion (INSERT), data updating (UPDATE), data deletion (DELETE), and data querying are some of the fundamental DML processes (SELECT). Datas that have the same value can affect each other and that once a set is empty using the Update command will not update it as shown in the observation.

9. Assessment Rubric