Chapter 4

Data Manipulation Language (DML)

"Flight Reservation Systems decide whether or not you exist. If your information isn't in their database, then you simply don't get to go anywhere." - Arthur Miller

4.1. Data Manipulation Language (DML)

DML serves to manipulate data in the database that has been created. The commands/instructions for DML in SQL are:

1. INSERT: insert or add new data into the table
2. UPDATE: update the old data to the new data.
3. DELETE: delete data in the table.

The difference between DDL and DML are:

DDL	DML
It stands for Data Definition Language.	It stands for Data Manipulation Language.
It is used to create database schema and can be used to define some constraints as well.	It is used to add, retrieve or update the data.
It basically defines the column (Attributes) of the table.	It adds or updates the row of the table. These rows are called tuples.
It doesn't have any further classification.	It is further classified into Procedural and Non-Procedural DML.
Basic commands present in DDL are CREATE, DROP, ALTER	BASIC commands present in DML are UPDATE, INSERT, DELETE
DDL does not use the WHERE clause in its statement.	While DML uses a WHERE clause in its statement.

4.2. INSERT command

INSERT command followed by INTO in SQL is used to insert a new row in a table. There are two ways of using the INSERT INTO command for inserting the rows:

1. Only values: The first method is to specify only the value of data to be inserted without the column names, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table.

Syntax:

INSERT INTO table_name VALUES (value1, value2, value3,…);

Example:

We have a student table with NPM, Name and Address as the attributes.

```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| NPM     | char(10)    | NO   | PRI | NULL    |       |
| Name    | varchar(20) | YES  |     | NULL    |       |
| Address | varchar(20) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

Since the record of the table is empty, we want to add a new record to the table by using the INSERT INTO command:

```
MariaDB [DML]> INSERT INTO Student VALUES ('2021010101', 'BUDI', 'Karawang');
Query OK, 1 row affected (0.070 sec)
```

We can use SELECT command to show the records from table(s), with the syntax:

SELECT column1,column2 FROM table_name; or **SELECT * FROM table_name;**

For example we want to see the record that we inserted into the student table:

```
MariaDB [DML]> SELECT * FROM student;
+------------+------+----------+
| NPM        | Name | Address  |
+------------+------+----------+
| 2021010101 | BUDI | Karawang |
+------------+------+----------+
1 row in set (0.001 sec)
```

See the example below and discuss the difference from the earlier command of inserting records.

```
MariaDB [DML]> INSERT INTO student VALUES
    -> ('2021010102', 'ANITA', 'Jakarta'),
    -> ('2021010103', 'HABIBIE', 'Bekasi'),
    -> ('2021010104', 'ADIL','Jakarta'),
    -> ('2021010105', 'EKA', 'Karawang');
Query OK, 4 rows affected (0.130 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+----------+
| NPM        | Name    | Address  |
+------------+---------+----------+
| 2021010101 | BUDI    | Karawang |
| 2021010102 | ANITA   | Jakarta  |
| 2021010103 | HABIBIE | Bekasi   |
| 2021010104 | ADIL    | Jakarta  |
| 2021010105 | EKA     | Karawang |
+------------+---------+----------+
5 rows in set (0.001 sec)
```

2. Column names and values both: In the second method, we will specify both the columns which we want to fill and their corresponding values.

   Syntax:

   INSERT INTO table_name (column1, column2, column3,..) VALUES (value1, value2, value3,..);

```
MariaDB [DML]> INSERT INTO student (NPM, Name, Address)
    -> VALUES ('2021010106', 'SUSANTI', 'Bogor');
Query OK, 1 row affected (0.052 sec)
```

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+----------+
| NPM        | Name    | Address  |
+------------+---------+----------+
| 2021010101 | BUDI    | Karawang |
| 2021010102 | ANITA   | Jakarta  |
| 2021010103 | HABIBIE | Bekasi   |
| 2021010104 | ADIL    | Jakarta  |
| 2021010105 | EKA     | Karawang |
| 2021010106 | SUSANTI | Bogor    |
+------------+---------+----------+
6 rows in set (0.001 sec)
```

4.3. UPDATE command

The UPDATE statement in SQL is used to update the data of an existing table in the database. We can update single columns as well as multiple columns using the UPDATE statement as per our requirement.

Syntax:

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

Note: Be careful when updating records in a table! Notice the WHERE clause in the UPDATE statement. The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated!

Example:

Using the student table

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+----------+
| NPM        | Name    | Address  |
+------------+---------+----------+
| 2021010101 | BUDI    | Karawang |
| 2021010102 | ANITA   | Jakarta  |
| 2021010103 | HABIBIE | Bekasi   |
| 2021010104 | ADIL    | Jakarta  |
| 2021010105 | EKA     | Karawang |
| 2021010106 | SUSANTI | Bogor    |
+------------+---------+----------+
6 rows in set (0.001 sec)
```

We need to update ANITA's address into Tangerang since she moved from Jakarta to Tangerang.

```
MariaDB [DML]> UPDATE student
    -> SET Address = 'Tangerang'
    -> WHERE NPM ='2021010102';
Query OK, 1 row affected (0.141 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+-----------+
| NPM        | Name    | Address   |
+------------+---------+-----------+
| 2021010101 | BUDI    | Karawang  |
| 2021010102 | ANITA   | Tangerang |
| 2021010103 | HABIBIE | Bekasi    |
| 2021010104 | ADIL    | Jakarta   |
| 2021010105 | EKA     | Karawang  |
| 2021010106 | SUSANTI | Bogor     |
+------------+---------+-----------+
6 rows in set (0.001 sec)
```

4.4. DELETE command

The DELETE command is used to delete existing records in a table.

Syntax:

```
DELETE FROM table_name WHERE condition;
```

Note: Be careful when deleting records in a table! Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

Example:

The records of student table, and we want to delete 1 of the record.

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+-----------+
| NPM        | Name    | Address   |
+------------+---------+-----------+
| 2021010101 | BUDI    | Karawang  |
| 2021010102 | ANITA   | Tangerang |
| 2021010103 | HABIBIE | Bekasi    |
| 2021010104 | ADIL    | Jakarta   |
| 2021010105 | EKA     | Karawang  |
| 2021010106 | SUSANTI | Bogor     |
+------------+---------+-----------+
6 rows in set (0.001 sec)
```

The following SQL statement deletes EKA's record from the table

```
MariaDB [DML]> DELETE FROM student WHERE NPM ='2021010105';
Query OK, 1 row affected (0.130 sec)
```

Now the student table will look like this:

```
MariaDB [DML]> SELECT * FROM student;
+------------+---------+-----------+
| NPM        | Name    | Address   |
+------------+---------+-----------+
| 2021010101 | BUDI    | Karawang  |
| 2021010102 | ANITA   | Tangerang |
| 2021010103 | HABIBIE | Bekasi    |
| 2021010104 | ADIL    | Jakarta   |
| 2021010106 | SUSANTI | Bogor     |
+------------+---------+-----------+
5 rows in set (0.001 sec)
```

TASK 4:

1. Write an article (cover, table of contents, discussion)
2. Use Times New Roman font size 12, margin 2,5cm, letter
3. Do documentation (screenshot) on each query and its results (show tables, desc table, select *)
4. Design the database as follows:
   a. Create a "sales" database
   b. Consists of at least 3 tables that are related to each other (Primary Key & Foreign Key)
   c. Each table consists of: There are DML commands (INSERT (10), UPDATE (5), DELETE (2))