MySQL Statements Basic

Note by Tawhid Monowar

1. **CREATE DATABASE** (Creates a new database)

```
CREATE DATABASE database_name;
```

2. **CREATE TABLE** (Creates a new table in a database)

```
CREATE TABLE table_name (
        column1 datatype,
        column2 datatype,
        column3 datatype,
);
```

3. **INSERT INTO** (Inserts data into a table)

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

/*adding values for all the columns of the table*/
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

4. **SELECT** (Retrieves data from a table)

```
SELECT column1, column2, ...
FROM table_name;

/* select all the fields available in the table */
SELECT * FROM table_name;
```

5. **SELECT DISTINCT** (SELECT DISTINCT statement is used to return only distinct values)

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

6. **WHERE** (WHERE clause is used to filter records)

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

7. **UPDATE** (UPDATE statement is used to modify the existing records in a table.)

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

8. **DELETE** (DELETE statement is used to delete existing records in a table)

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

9. **AND, OR, NOT, IN, LIKE, BETWEEN** (Operators)

```
/*AND Syntax*/
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;

/*OR Syntax*/
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;

/*NOT Syntax*/
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;

/*IN Syntax*/
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);

/*LIKE Syntax*/
/*Wildcards (%) (_)*/
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;

/*BETWEEN Syntax*/
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

10. **SUM, MIN, MAX & AVG** (Aggregate Functions)

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

11. **ORDER BY** (Used to sort the result-set in ascending or descending order)

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

12. **GROUP BY** (Groups data based on a column)

```
SELECT column1, COUNT(column2) FROM table_name GROUP BY column1;
```

13. **LIMIT, OFFSET** (LIMIT clause is used to specify the number of records to return)

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number OFFSET starting position;
```

14. **Constraints** (constraints are used to specify rules for data in a table)

```
CREATE TABLE table_name (
   col_name1 datatype constraint
   .....
);
```

NOT NULL - Ensures that a column cannot have a NULL value.
UNIQUE - Ensures that all values in a column are different.
PRIMARY KEY - A combination of NOT NULL and UNIQUE. Uniquely identifies each row.
FOREIGN KEY - Prevents actions that would destroy links between tables.
CHECK - Ensures that the values in a column satisfies a specific condition.
DEFAULT - Sets a default value for a column if no value is specified.
CREATE INDEX - Used to create and retrieve data from the database very quickly.