SQL | DDL, DQL, DML, DCL and TCL Commands

1. Create a Table:

Command Prompt - mysql -u root -p

```
mysql> create database empdb;
Query OK, 1 row affected (0.03 sec)

mysql> use empdb;
Database changed
mysql> select database();
+------------+
| database() |
+------------+
| empdb      |
+------------+
1 row in set (0.00 sec)

mysql>
```

2. Storing Data in a Table:

Command Prompt - mysql -u root -p

```
mysql> CREATE TABLE Employees (
    ->     EmployeeID INT PRIMARY KEY,
    ->     FirstName VARCHAR(50),
    ->     LastName VARCHAR(50),
    ->     Age INT,
    ->     Department VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

3. Updating Data in a Table:

```
mysql> -- Update the age of employee with EmployeeID 2
mysql> UPDATE Employees
    -> SET Age = 26
    -> WHERE EmployeeID = 2;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

4. Deleting Data from a Table:

```
mysql> -- Delete the employee with EmployeeID 3
mysql> DELETE FROM Employees
    -> WHERE EmployeeID = 3;
Query OK, 1 row affected (0.03 sec)
```

5. Retrieving Specific Attributes:

```
mysql> -- Retrieve EmployeeID and FirstName from the 'Employees' table
mysql> SELECT EmployeeID, FirstName
    -> FROM Employees;
+------------+-----------+
| EmployeeID | FirstName |
+------------+-----------+
|          1 | John      |
|          2 | Jane      |
+------------+-----------+
2 rows in set (0.00 sec)
```

6. Retrieving Selected Rows:

```
mysql> -- Retrieve all columns for employees in the 'IT' department
mysql> SELECT *
    -> FROM Employees
    -> WHERE Department = 'IT';
+------------+-----------+----------+------+------------+
| EmployeeID | FirstName | LastName | Age  | Department |
+------------+-----------+----------+------+------------+
|          2 | Jane      | Smith    |   26 | IT         |
+------------+-----------+----------+------+------------+
```

7. Filtering Data: WHERE Clauses:

```
mysql> -- Retrieve employees older than 28
mysql> SELECT *
    -> FROM Employees
    -> WHERE Age > 28;
+------------+-----------+----------+------+------------+
| EmployeeID | FirstName | LastName | Age  | Department |
+------------+-----------+----------+------+------------+
|          1 | John      | Doe      |   30 | HR         |
+------------+-----------+----------+------+------------+
1 row in set (0.03 sec)
```

8. Column & Table Aliases:

```
mysql> -- Retrieve EmployeeID as ID and FirstName as First_Name
mysql> SELECT EmployeeID AS ID, FirstName AS First_Name
    -> FROM Employees;
+----+------------+
| ID | First_Name |
+----+------------+
|  1 | John       |
|  2 | Jane       |
+----+------------+
2 rows in set (0.00 sec)
```

9. Using LIKE:

Command Prompt - mysql -u root -p

```
mysql> SELECT *
    -> FROM Employees
    -> WHERE LastName LIKE 'D%';
+------------+-----------+----------+------+------------+
| EmployeeID | FirstName | LastName | Age  | Department |
+------------+-----------+----------+------+------------+
|          1 | John      | Doe      |   30 | HR         |
+------------+-----------+----------+------+------------+
1 row in set (0.03 sec)
```

10. Using ALL:

```
mysql> -- Retrieve employees with a salary greater than SOME employees in the Finance department
mysql> SELECT *
    -> FROM Employees
    -> WHERE Salary > SOME (SELECT Salary FROM Salaries WHERE Department = 'Finance');
+------------+-----------+----------+------+------------+----------+
| EmployeeID | FirstName | LastName | Age  | Department | Salary   |
+------------+-----------+----------+------+------------+----------+
|          3 | Bob       | Johnson  |   35 | Finance    | 75000.00 |
+------------+-----------+----------+------+------------+----------+
1 row in set (0.00 sec)
```

11. Using SOME:

Command Prompt - mysql -u root -p

```
mysql> -- Retrieve employees with a salary greater than ALL employees in the IT department
mysql> SELECT *
    -> FROM Employees
    -> WHERE Salary > ALL (SELECT Salary FROM Salaries WHERE Department = 'IT');
+------------+-----------+----------+------+------------+----------+
| EmployeeID | FirstName | LastName | Age  | Department | Salary   |
+------------+-----------+----------+------+------------+----------+
|          1 | John      | Doe      |   30 | HR         | 50000.00 |
|          3 | Bob       | Johnson  |   35 | Finance    | 75000.00 |
+------------+-----------+----------+------+------------+----------+
2 rows in set (0.00 sec)
```

12. Using ANY:

```
mysql> -- Retrieve employees with a salary greater than ANY employee in the HR department
mysql> SELECT *
    -> FROM Employees
    -> WHERE Salary > ANY (SELECT Salary FROM Salaries WHERE Department = 'HR');
Empty set (0.00 sec)
```

13. Using EXISTS:

```
mysql> -- Retrieve employees who have a salary record in the 'Salaries' table
mysql> SELECT *
    -> FROM Employees
    -> WHERE EXISTS (SELECT 1 FROM Salaries WHERE Salaries.EmployeeID = Employees.EmployeeID);
+------------+-----------+----------+------+------------+----------+
| EmployeeID | FirstName | LastName | Age  | Department | Salary   |
+------------+-----------+----------+------+------------+----------+
|          1 | John      | Doe      |   30 | HR         | 50000.00 |
|          2 | Jane      | Smith    |   25 | IT         | 60000.00 |
|          3 | Bob       | Johnson  |   35 | Finance    | 75000.00 |
+------------+-----------+----------+------+------------+----------+
3 rows in set (0.03 sec)
```

14. UNION (Combine rows from two tables, removing duplicates):

```
mysql> -- Combine distinct rows from 'Employees' and 'NewEmployees'
mysql> SELECT * FROM Employees
    -> UNION
    -> SELECT * FROM NewEmployees;
+------------+-----------+----------+------+------------+----------+
| EmployeeID | FirstName | LastName | Age  | Department | Salary   |
+------------+-----------+----------+------+------------+----------+
|          1 | John      | Doe      |   30 | HR         | 50000.00 |
|          2 | Jane      | Smith    |   25 | IT         | 60000.00 |
|          3 | Bob       | Johnson  |   35 | Finance    | 75000.00 |
|          4 | Alice     | Johnson  |   28 | IT         | 58000.00 |
|          5 | Charlie   | Brown    |   32 | Finance    | 70000.00 |
|          6 | Eva       | Williams |   27 | HR         | 52000.00 |
+------------+-----------+----------+------+------------+----------+
6 rows in set (0.05 sec)
```

15. INTERSECT (Retrieve common rows between two tables):

```
mysql> -- Retrieve common rows between 'Employees' and 'NewEmployees'
mysql> SELECT * FROM Employees
    -> INTERSECT
    -> SELECT * FROM NewEmployees;
Empty set (0.00 sec)
```

16. EXCEPT (Retrieve rows from the first table that are not in the second table):

```
mysql> -- Retrieve rows from 'Employees' that are not in 'NewEmployees'
mysql> SELECT * FROM Employees
    -> EXCEPT
    -> SELECT * FROM NewEmployees;
+------------+-----------+----------+------+------------+----------+
| EmployeeID | FirstName | LastName | Age  | Department | Salary   |
+------------+-----------+----------+------+------------+----------+
|          1 | John      | Doe      |   30 | HR         | 50000.00 |
|          2 | Jane      | Smith    |   25 | IT         | 60000.00 |
|          3 | Bob       | Johnson  |   35 | Finance    | 75000.00 |
+------------+-----------+----------+------+------------+----------+
3 rows in set (0.00 sec)
```

17. Using COMMIT

```
mysql> -- Start a transaction
mysql> BEGIN;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> -- Update data in 'Employees'
mysql> UPDATE Employees SET Salary = Salary * 1.1 WHERE Department = 'HR';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```