Unit 4 Structured Query Language (SQL)

SQL (Structured Query Language) is a standardized programming language used for managing and manipulating relational databases. It allows users to create, modify, query, and manage data stored in a relational database system (RDBMS). SQL is widely used due to its powerful capabilities and ease of use for handling large amounts of data.
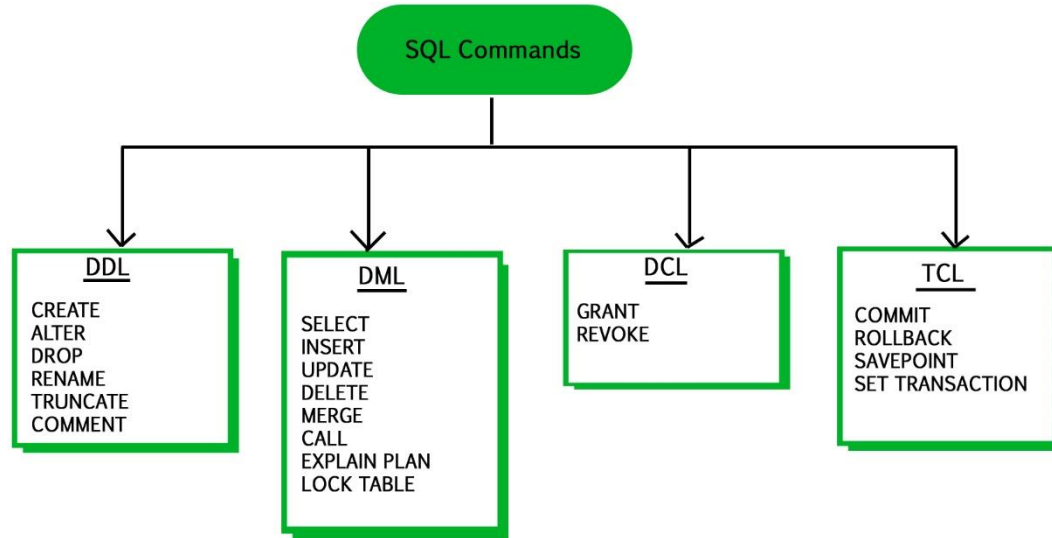
1. **Data Definition Language (DDL):**

   - **CREATE:** Defines new tables, views, indexes, etc.

   ```sql
   sql                                                  Copy code

   CREATE TABLE Employees (
       EmployeeID INT PRIMARY KEY,
       Name VARCHAR(100),
       Position VARCHAR(50),
       Salary DECIMAL(10, 2)
   );
   ```

   - **ALTER:** Modifies existing database objects.

   ```sql
   sql                                                  Copy code

   ALTER TABLE Employees ADD COLUMN HireDate DATE;
   ```

   - **DROP:** Deletes database objects.

   ```sql
   sql                                                  Copy code

   DROP TABLE Employees;
   ```

2. **Data Manipulation Language (DML):**

- **SELECT**: Retrieves data from one or more tables.

```sql
SELECT Name, Position FROM Employees WHERE Salary > 50000;
```

- **INSERT**: Adds new rows to a table.

```sql
INSERT INTO Employees (EmployeeID, Name, Position, Salary)
VALUES (1, 'John Doe', 'Manager', 75000);
```

- **UPDATE**: Modifies existing rows in a table.

```sql
UPDATE Employees SET Salary = 80000 WHERE EmployeeID = 1;
```

- **DELETE**: Removes rows from a table.

```sql
DELETE FROM Employees WHERE EmployeeID = 1;
```

3. **Data Control Language (DCL):**

- **GRANT:** Gives users access privileges.

```sql
GRANT SELECT, INSERT ON Employees TO User1;
```

- **REVOKE:** Removes user access privileges.

```sql
REVOKE SELECT, INSERT ON Employees FROM User1;
```

SQL clauses are used to filter, sort, and organize data retrieved from a database.

1. WHERE Clause

The WHERE clause filters records based on specified conditions. It is used to extract only those records that fulfill a specified condition.

Syntax:

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

Example:

```sql
SELECT Name, Salary
FROM Employees
WHERE Salary > 50000;
```

In this example, only employees with a salary greater than 50,000 are selected.

2. AND / OR Clauses

The AND and OR operators are used to combine multiple conditions in a WHERE clause.

AND: Returns records that satisfy all conditions.

Syntax:

```sql
SELECT column1, column2
FROM table_name
WHERE condition1 AND condition2;
```

Example:

```sql
SELECT Name, Salary
FROM Employees
WHERE Salary > 50000 AND Position = 'Manager';
```

This query selects employees who are managers and have a salary greater than 50,000.

OR: Returns records that satisfy at least one of the conditions.

Syntax:

```sql
SELECT column1, column2
FROM table_name
WHERE condition1 OR condition2;
```

Example:

```sql
SELECT Name, Salary
FROM Employees
WHERE Salary > 50000 OR Position = 'Manager';
```

This query selects employees who either have a salary greater than 50,000 or are managers.

3. WITH Clause

The WITH clause, also known as Common Table Expressions (CTE), is used to define temporary result sets that can be referenced within a SELECT, INSERT, UPDATE, or DELETE statement. CTEs are useful for simplifying complex queries.

Syntax:

```sql
WITH CTE_Name AS (
    SELECT column1, column2
    FROM table_name
    WHERE condition
)
SELECT column1
FROM CTE_Name;
```

Example:

```sql
WITH HighEarners AS (
    SELECT Name, Salary
    FROM Employees
    WHERE Salary > 50000
)
SELECT Name
FROM HighEarners;
```

This query first defines a CTE HighEarners with employees earning more than 50,000 and then selects their names.

4. ORDER BY Clause

The ORDER BY clause sorts the result set of a query by one or more columns. By default, it sorts in ascending order. To sort in descending order, you use the DESC keyword.

Syntax:

```sql
SELECT column1, column2
FROM table_name
ORDER BY column1 [ASC|DESC], column2 [ASC|DESC];
```

Example:

```sql
SELECT Name, Salary
FROM Employees
ORDER BY Salary DESC, Name ASC;
```

This query sorts employees by salary in descending order and, within the same salary, by name in ascending order.