1. **DQL** *(Data Query Language)* : Used to retrieve data from databases. (SELECT)
2. **DDL** *(Data Definition Language)* : Used to create, alter, and delete database objects like tables, indexes, etc. (CREATE, DROP, ALTER, RENAME, TRUNCATE)
3. **DML** *(Data Manipulation Language)*: Used to modify the database. (INSERT, UPDATE, DELETE)
4. **DCL** *(Data Control Language)*: Used to grant & revoke permissions. (GRANT, REVOKE)
5. **TCL** *(Transaction Control Language)*: Used to manage transactions. (COMMIT, ROLLBACK, START TRANSACTIONS, SAVEPOINT)

---

1. Data Definition Language (DDL)

Data Definition Language (DDL) is a subset of SQL (Structured Query Language) responsible for defining and managing the structure of databases and their objects.

DDL commands enable you to create, modify, and delete database objects like tables, indexes, constraints, and more.

Key DDL Commands are:

- **CREATE TABLE:**
    - Used to create a new table in the database.
    - Specifies the table name, column names, data types, constraints, and more.
    - Example:
      CREATE TABLE employees (id INT PRIMARY KEY, name VARCHAR(50), salary DECIMAL(10, 2));

- **ALTER TABLE:**
    - Used to modify the structure of an existing table.
    - You can add, modify, or drop columns, constraints, and more.
    - Example: ALTER TABLE employees ADD COLUMN email VARCHAR(100);

- **DROP TABLE:**
    - Used to delete an existing table along with its data and structure.
    - Example: DROP TABLE employees;

1. Inner Join
2. Outer Join
3. Cross Join
4. Self Join

1) Inner Join

An inner join combines data from two or more tables based on a specified condition, known as the join condition.
The result of an inner join includes only the rows where the join condition is met in all participating tables.
It essentially filters out non-matching rows and returns only the rows that have matching values in both tables.

Syntax:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
```

Here:

- columns refers to the specific columns you want to retrieve from the tables.
- table1 and table2 are the names of the tables you are joining.
- column is the common column used to match rows between the tables.
- The ON clause specifies the join condition, where you define how the tables are related.

Example: Consider two tables: Customers and Orders.

Customers Table:

CustomerID	CustomerName
1	Alice
2	Bob
3	Carol

Orders Table:

OrderID	CustomerID	Product

NULL	Monitor
NULL	Keyboard

In this full outer join example, all rows from both tables are included in the result. Both non-matching rows from the Customers and Orders tables are represented with NULL values.

3) Cross Join

A cross join, also known as a Cartesian product, is a type of join operation in a Database Management System (DBMS) that combines every row from one table with every row from another table.

Unlike other join types, a cross join does not require a specific condition to match rows between the tables. Instead, it generates a result set that contains all possible combinations of rows from both tables.

Cross joins can lead to a large result set, especially when the participating tables have many rows.

Syntax:

```
SELECT columns
FROM table1
CROSS JOIN table2;
```

In this syntax:

- columns refers to the specific columns you want to retrieve from the cross-joined tables.
- table1 and table2 are the names of the tables you want to combine using a cross join.

Example: Consider two tables: Students and Courses.

Students Table:

StudentID	StudentName
1	Alice

Self joins are commonly used to find relationships, hierarchies, or patterns within a single table.

In a self join, you treat the table as if it were two separate tables, referring to them with different aliases.

Syntax:

The syntax for performing a self join in SQL is as follows:

```
SELECT columns
FROM table1 AS alias1
JOIN table1 AS alias2 ON alias1.column = alias2.column;
```

In this syntax:

- columns refers to the specific columns you want to retrieve from the self-joined table.
- table1 is the name of the table you're joining with itself.
- alias1 and alias2 are aliases you assign to the table instances for differentiation.
- column is the column you use as the join condition to link rows from the same table.

Example: Consider an Employees table that contains information about employees and their managers.

Employees Table:

EmployeeID	EmployeeName	ManagerID
1	Alice	3
2	Bob	3
3	Carol	NULL
4	David	1

Self Join Query:

```
SELECT e1.EmployeeName AS Employee, e2.EmployeeName AS Manager
FROM Employees AS e1
JOIN Employees AS e2 ON e1.ManagerID = e2.EmployeeID;
```

Result:

Employee	Manager