DDL - Data Definition Language (Create Statement)

1. CREATE TABLE Statement

This SQL statement creates the new table in the SQL database.

Syntax of CREATE TABLE Statement:

```
CREATE TABLE table_name
(
column_name1 data_type [column1 constraint(s)],
column_name2 data_type [column2 constraint(s)],
.....
.....,
column_nameN data_type [columnN constraint(s)],
PRIMARY KEY(one or more col)
);
```

DDL - Data Definition Language:Example

Example of CREATE TABLE Statement:

```
CREATE TABLE Employee_details(
    Emp_Id NUMBER(4) NOT NULL,
    First_name VARCHAR(30),
    Last_name VARCHAR(30),
    Salary Money,
    City VARCHAR(30),
    PRIMARY KEY (Emp_Id)
);
```

- This example creates the table Employee_details with five columns or fields in the SQL database. The fields in the table are Emp_Id, First_Name, Last_Name, Salary, and City. The Emp_Id column in the table acts as a primary key, which means that the Emp_Id column cannot contain duplicate values and null values.

DDL - Data Definition Language (Alter statement)

2. ALTER TABLE Statement

This SQL statement adds, deletes, and modifies the columns of the table in the SQL database.

Syntax of ALTER TABLE Statement:

ALTER TABLE table_name ADD column_name datatype[(size)];

The above SQL alter statement adds the column with its datatype in the existing database table.

ALTER TABLE table_name MODIFY column_name column_datatype[(size)];

The above 'SQL alter statement' renames the old column name to the new column name of the existing database table.

-ALTER TABLE table_name DROP COLUMN column_name;

The above SQL alter statement deletes the column of the existing database table.

Example of ALTER TABLE Statement:

– ALTER TABLE Employee_details ADD Designation VARCHAR(18);

This example adds the new field whose name is **Designation** with size **18** in the **Employee_details** table of the SQL database.

DDL - Data Definition Language (Drop Statement)

3. DROP TABLE Statement

This SQL statement deletes or removes the table and the structure, views, permissions, and triggers associated with that table.

Syntax of DROP TABLE Statement:

```
DROP TABLE [ IF EXISTS ]
table_name1, table_name2, ……, table_nameN;
```

The above syntax of the drop statement deletes specified tables completely if they exist in the database.

Example of DROP TABLE Statement:

```
– DROP TABLE Employee_details;
```

This example drops the Employee_details table if it exists in the SQL database. This removes the complete information if available in the table.

DDL - Data Definition Language (Truncate Statement)

4. SQL TRUNCATE TABLE

A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause.

TRUNCATE TABLE Vs DELETE TABLE

Truncate table is faster and uses lesser resources than DELETE TABLE command.

TRUNCATE TABLE Vs DROP TABLE

Drop table command can also be used to delete complete table but it deletes table structure too. TRUNCATE TABLE doesn't delete the structure of the table.

Syntax**:**

TRUNCATE TABLE table_name;

For example: Write following command to truncate the data of employee table

TRUNCATE TABLE Employee;

Note: The rollback process is not possible after truncate table statement. Once you truncate a table you cannot use a flashback table statement to retrieve the content of the table.

DML - Data Manipulation Language (Select Statement)

Let's discuss each statement in short one by one with syntax and one example:

5. SELECT Statement

- This SQL statement reads the data from the SQL database and shows it as the output to the database user.

Syntax of SELECT Statement:

```
SELECT column_name1, column_name2, ...., column_nameN
   [ FROM table_name ]
   [ WHERE condition ]
   [ ORDER BY order_column_name1 [ ASC | DESC ], .... ];
```

SQL - WHERE Clause

- The SQL WHERE clause is used to specify a condition while fetching the data from a single table or by joining with multiple tables. If the given condition is satisfied, then only it returns a specific value from the table.
- WHERE clause is use to filter the records and fetching only the necessary records.
- The WHERE clause is not only used in the SELECT statement, but it is also used in the UPDATE, DELETE statement, etc., which we would examine in the subsequent chapters.

Syntax : The basic syntax of the SELECT statement with the WHERE clause

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition]
```

SQL - AND and OR Conjunctive Operators

- The SQL AND & OR operators are used to combine multiple conditions to narrow data in an SQL statement. These two operators are called as the conjunctive operators.
- The AND Operator: The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.
- Syntax: The basic syntax of the AND operator with a WHERE clause is as follows −

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] AND [condition2]...AND [conditionN];
```

- The OR Operator: The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.

Syntax: The basic syntax of the OR operator with a WHERE clause is as follows −

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] OR [condition2]...OR [conditionN]
```

SQL - SORTING Results

- The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns. Some databases sort the query results in an ascending order by default.
- Syntax: ORDER BY clause which would be used to sort the result in an ascending or descending order is as follows −

```
SELECT column-list
FROM table_name
[WHERE condition]
[ORDER BY column1, column2, .. columnN] [ASC | DESC];
```

- You can use more than one column in the ORDER BY clause. Make sure that whatever column you are using to sort, that column should be in the column-list.

DML - Select Statement: Example

Example of SELECT Statement:

```
SELECT Emp_ID, First_Name, Last_Name, Salary, City
FROM Employee_details
WHERE Salary = 100000
ORDER BY Last_Name  ;
```

- This example shows the Emp_ID, First_Name, Last_Name, Salary, and City of those employees from the Employee_details table whose Salary is 100000. The output shows all the specified details according to the ascending alphabetical order of Last_Name.

DML - Data Manipulation Language(Update Statement

6. UPDATE Statement:

– This SQL statement changes or modifies the stored data in the SQL database.

Syntax of UPDATE Statement:

```
UPDATE table_name
SET column_name1 = new_value_1, column_name2 = new_value_2, ....,
column_nameN = new_value_N
[ WHERE  CONDITION ];
```

DML - Update Statement : Example

Example of UPDATE Statement:

```
UPDATE Employee_details
SET Salary = 100000
WHERE Emp_ID = 10;
```

This example changes the Salary of those employees of the Employee_details table whose Emp_ID is 10 in the table.

DML – Data Manipulation Language (Delete Statement)

7. DELETE Statement

– This SQL statement deletes the stored data from the SQL database.

Syntax of DELETE Statement:

DELETE FROM table_name

[**WHERE** CONDITION];

DML –Delete Statement: Example

Example of DELETE Statement:

```
DELETE FROM Employee_details
WHERE First_Name = 'Sumit' ;
```

This example deletes the record of those employees from the **Employee_details** table whose **First_Name** is **Sumit** in the table.

DML - Data Manipulation Language (Insert statement)

8. INSERT INTO Statement

This SQL statement inserts the data or records in the existing table of the SQL database. This statement can easily insert single and multiple records in a single query statement.

Syntax of insert a single record:

```
INSERT INTO table_name
( column_name1, column_name2, ...., column_nameN
) VALUES (value_1, value_2, ....., value_N );
```

Example of insert a single record:

```
INSERT INTO Employee_details
( Emp_ID, First_name, Last_name, Salary, City
) VALUES (101, Akhil, Sharma, 40000, Bangalore );
```

This example inserts **101** in the first column, **Akhil** in the second column, **Sharma** in the third column, **40000** in the fourth column, and **Bangalore** in the last column of the table **Employee_details.**

DML - Data Manipulation Language (Insert statement)

- **Syntax of inserting a multiple records in a single query:**

  INSERT INTO table_name

  (column_name1, column_name2, …., column_nameN)

  VALUES (value_1, value_2, ……, value_N), (value_1, value_2, ……, value_N),….;

- **Example of inserting multiple records in a single query:**

  INSERT INTO Employee_details

  (Emp_ID, First_name, Last_name, Salary, City)

  VALUES (101, Amit, Gupta, 50000, Mumbai), (102, John, Aggarwal, 45000, Calcutta), (103, Sidhu, Arora, 55000, Mumbai);

- This example inserts the records of three employees in the **Employee_details** table in the single query statement.

DDL - CREATE DATABASE Statement

9. CREATE DATABASE Statement

This SQL statement creates the new database in the database management system.

Syntax of CREATE DATABASE Statement:

CREATE DATABASE database_name;

Example of CREATE DATABASE Statement:

CREATE DATABASE Company;

The above example creates the company database in the system.

DDL - DROP DATABASE Statement

10. DROP DATABASE Statement

This SQL statement deletes the existing database with all the data tables and views from the database management system.

Syntax of DROP DATABASE Statement:

- **DROP DATABASE** database_name;

Example of DROP DATABASE Statement:

- **DROP DATABASE** Company;

The above example deletes the company database from the system.

DCL - Data Control Language (Grant statement)

- DCL commands are used to enforce database security in a multiple user database environment. **Two types of DCL commands are GRANT and REVOKE**. Only Database Administrator's or owner's of the database object can provide/remove privileges on a database object.

11. SQL GRANT Command

- SQL GRANT is a command used to provide access or privileges on the database objects to the users.

- The Syntax for the GRANT command is:

```
GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];
```