DML Commands in SQL

DML is an abbreviation of **Data Manipulation Language**.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

SELECT DML Command

SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

Syntax of SELECT DML command

1. **SELECT** column_Name_1, column_Name_2, ….., column_Name_N **FROM** Name_ of_table;

Here, **column_Name_1, column_Name_2, ….., column_Name_N** are the names of those columns whose data we want to retrieve from the table.

If we want to retrieve the data from all the columns of the table, we have to use the following SELECT command:

1. **SELECT * FROM** table_name;

Examples of SELECT Command

Example 1: This example shows all the values of every column from the table.

1. **SELECT * FROM** Student;

This SQL statement displays the following values of the student table:

Student_ID	Student_Name	Student_Marks
BCA1001	Abhay	85
BCA1002	Anuj	75
BCA1003	Bheem	60
BCA1004	Ram	79
BCA1005	Sumit	80

Example 2: This example shows all the values of a specific column from the table.

1. **SELECT** Emp_Id, Emp_Salary **FROM** Employee;

This SELECT statement displays all the values of **Emp_Salary** and **Emp_Id** column of **Employee** table:

Emp_Id	Emp_Salary
201	25000
202	45000
203	30000
204	29000
205	40000

Example 3: This example describes how to use the WHERE clause with the SELECT DML command.

Let's take the following Student table:

Student_ID	Student_Name	Student_Marks
BCA1001	Abhay	80
BCA1002	Ankit	75
BCA1003	Bheem	80
BCA1004	Ram	79
BCA1005	Sumit	80

If you want to access all the records of those students whose marks is 80 from the above table, then you have to write the following DML command in SQL:

1. **SELECT * FROM** Student **WHERE** Stu_Marks = 80;

The above SQL query shows the following table in result:

Student_ID	Student_Name	Student_Marks
BCA1001	Abhay	80
BCA1003	Bheem	80
BCA1005	Sumit	80

INSERT DML Command

INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.

Syntax of INSERT Command

1. **INSERT INTO** TABLE_NAME (column_Name1 , column_Name2 , column_Name 3 , ... column_NameN) **VALUES** (value_1, value_2, value_3, ... value_N) ;

Examples of INSERT Command

Example 1: This example describes how to insert the record in the database table.

Let's take the following student table, which consists of only 2 records of the student.

Stu_Id	Stu_Name	Stu_Marks	Stu_Age
101	Ramesh	92	20
201	Jatin	83	19

Suppose, you want to insert a new record into the student table. For this, you have to write the following DML INSERT command:

1. **INSERT INTO** Student (Stu_id, Stu_Name, Stu_Marks, Stu_Age) **VALUES** (104, A nmol, 89, 19);

UPDATE DML Command

UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

Syntax of UPDATE Command

1. **UPDATE** Table_name **SET** [column_name1= value_1, ….., column_nameN = value _N] **WHERE** CONDITION;

Here, 'UPDATE', 'SET', and 'WHERE' are the SQL keywords, and 'Table_name' is the name of the table whose values you want to update.

Examples of the UPDATE command

Example 1: This example describes how to update the value of a single field.

Let's take a Product table consisting of the following records:

Product_Id	Product_Name	Product_Price	Product_Quantity
P101	Chips	20	20

P102	Chocolates	60	40
P103	Maggi	75	5
P201	Biscuits	80	20
P203	Namkeen	40	50

Suppose, you want to update the Product_Price of the product whose Product_Id is P102. To do this, you have to write the following DML UPDATE command:

1. **UPDATE** Product **SET** Product_Price = 80 **WHERE** Product_Id = 'P102' ;

Example 2: This example describes how to update the value of multiple fields of the database table.

Let's take a Student table consisting of the following records:

Stu_Id	Stu_Name	Stu_Marks	Stu_Age
101	Ramesh	92	20
201	Jatin	83	19
202	Anuj	85	19
203	Monty	95	21
102	Saket	65	21
103	Sumit	78	19
104	Ashish	98	20

Suppose, you want to update Stu_Marks and Stu_Age of that student whose Stu_Id is 103 and 202. To do this, you have to write the following DML Update command:

1. **UPDATE** Student **SET** Stu_Marks = 80, Stu_Age = 21 **WHERE** Stu_Id = 103 AND Stu_Id = 202;

DELETE DML Command

DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

Syntax of DELETE Command

1. **DELETE FROM** Table_Name **WHERE** condition;

Examples of DELETE Command

Example 1: This example describes how to delete a single record from the table.

Let's take a Product table consisting of the following records:

Product_Id	Product_Name	Product_Price	Product_Quantity
P101	Chips	20	20
P102	Chocolates	60	40
P103	Maggi	75	5
P201	Biscuits	80	20
P203	Namkeen	40	50

Suppose, you want to delete that product from the Product table whose Product_Id is P203. To do this, you have to write the following DML DELETE command:

1. **DELETE FROM** Product **WHERE** Product_Id = 'P202' ;

Example 2: This example describes how to delete the multiple records or rows from the database table.

Let's take a Student table consisting of the following records:

Stu_Id	Stu_Name	Stu_Marks	Stu_Age
101	Ramesh	92	20
201	Jatin	83	19
202	Anuj	85	19
203	Monty	95	21
102	Saket	65	21
103	Sumit	78	19
104	Ashish	98	20

Suppose, you want to delete the record of those students whose Marks is greater than 70. To do this, you have to write the following DML Update command:

1. **DELETE FROM** Student **WHERE** Stu_Marks > 70 ;

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.

These operators provide a means to make multiple comparisons with different operators in the same SQL statement.

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];
```

Example:

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE SALARY > 2000 AND age < 25;
```

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]
```

Example:

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE SALARY > 2000 OR age < 25;
```

Like Clause:

The SQL **LIKE** clause is used to compare a value to similar values using wildcard operators. There are two wildcards used in conjunction with the LIKE operator.

- The percent sign (%)
- The underscore (_)

The percent sign represents zero, one or multiple characters. The underscore represents a single number or character. These symbols can be used in combinations.

Syntax

The basic syntax of % and _ is as follows −

```
SELECT FROM table_name
WHERE column LIKE 'XXXX%'
```

or

```
SELECT FROM table_name
WHERE column LIKE '%XXXX%'
```

or

```
SELECT FROM table_name
```

WHERE column LIKE 'XXXX_'

or

SELECT FROM table_name
WHERE column LIKE '_XXXX'

or

SELECT FROM table_name
WHERE column LIKE '_XXXX_'

You can combine N number of conditions using AND or OR operators. Here, XXXX could be any numeric or string value.

Example

The following table has a few examples showing the WHERE part having different LIKE clause with '%' and '_' operators −

Sr.No.	Statement & Description
1	**WHERE SALARY LIKE '200%'** Finds any values that start with 200.
2	**WHERE SALARY LIKE '%200%'** Finds any values that have 200 in any position.
3	**WHERE SALARY LIKE '_00%'** Finds any values that have 00 in the second and third positions.
4	**WHERE SALARY LIKE '2_%_%'** Finds any values that start with 2 and are at least 3 characters in length.
5	**WHERE SALARY LIKE '%2'** Finds any values that end with 2.