Date: 01/09/2025

Task: 3.1 DML commands using clauses, operators and functions in queries.

Aim: To implement of DML command using clauses, Operators and functions in queries

Data manuplation Language (DML):

The Data manipulation Language (DML) is using to retrive, insert and modify database information These commands will be used by all database users during the routine operation of the database. Let's take a brief look of the basic look at the basic DML commands:

1. INSERT

2. UPDATE

3. DELETE

INSERT INTO

This is used to add records into a relation

Syntax:

```
INSERT INTO <table_name> (field 1, field 2.. fieldN)
Values (data 1, data 2.. dataN),
```

Example:

Sql:

```
INSERT INTO Patients VALUES (111, 'Arun', 'Cardiology',
                             'Male');
```

Table after INSERT :

Patient ID	PatientName	Department	Gender
111	Arun	cardiology	Male

UPDATE - SET - WHERE :

This is used to update the content of a record in a relation.

Syntax:

Sql:

```
UPDATE table_name SET field 1 = data WHERE
 condition;
```

Example:

Sql :

```
UPDATE patients SET patient Name = 'Kumar'
 WHERE Patient ID = 111;
```

Table after UPDATE

Patient ID	Patient Name	Department	Gender
111	Kumar	Cardiology	Male

DELETE FROM:

This is used to delete all records of a relation but it retains the structure.

Syntax:

Sql:

```
DELETE FROM table_name;
```

Example:

Sql:

```
DELETE FROM Appointments;
```

Appointments table after DELETE:

Appointment ID	Patient ID	Doctor ID	Appointment	Purchase

DELETE - FROM - WHERE:

This is used to delete specific records from a relation.

Syntax:

Sql:

```
DELETE FROM table_name WHERE condition;
```

Example:

Sql:

```
DELETE FROM Doctors WHERE DoctorID = 202;
```

Doctors Table after DELETE:

Doctor ID	DoctorName	Department	Fees
201	Dr. Sharma	Cardiology	1000
203	Dr. Ahmed	Neurology	900
204	Dr. Rajesh	Orthopedic	500
205	Dr. Neha	Dermatology	800

TRUNCATE:

This removes all data permanently but keeps the table structure.

Syntax:

Sql:

```
TRUNCATE TABLE <table_name>;
```

Example:

Sql:

```
TRUNCATE TABLE Patients;
```

Patients Table after TRUNCATE

Patient ID	PatientName	Department	Gender

Sample Queries and Output:

1. Retrieve patient names ending with letter 'n' and patient no between 111 and 115.

Query:

Sql:

```
SELECT PatientName, Department, Gender
FROM Patients
WHERE PatientName LIKE '%n' AND Patient ID
BETWEEN 111 AND 115;
```

Patient Name	Department	Gender
Arun	Cardiology	Male
Karan	orthopedics	Male
Rohan	Dermatology	Male

2. List doctors where consultation fees between 700 and 800

Query:

Sql:

```
SELECT * FROM Doctors WHERE Fees BETWEEN 700 AND 800
```

Doctor ID	Doctor Name	Department	Fees
202	Dr. Priya	Padiatria	700
205	Dr. Neha	Dermatology	800

3. Find the record with minimum appointment duration.

Query:

Sql:

```
SELECT MIN (Duration) FROM Appointments;
```

MIN (Duration)
20

4. Find appointments with date ≥ '2023-02-07'

Query:

Sql:

```
SELECT * FROM Appointments WHERE
AppointmentDate >= '2023-02-07'
```

Appointment ID	Patient ID	Doctor ID	Appointment	Duration
302	112	203	2023-02-07	45
303	113	204	2023-02-04	20
304	114	202	2023-02-10	60
305	115	205	2023-02-12	25

5. List distinct Patient IDs

Query:

Sql:

```
SELECT DISTINCT Patient ID FROM Patients
```

Patient ID
111
112
113
114
115

6 combine Patient IDs from patients and Appointments (UNION)

query:

Sql:

```
SELECT patient ID FROM Patients
UNION
SELECT Patient ID FROM Appointment;
```

Output:

Patient ID
111
112
113
114
116

7. Group patients based on gender and department

Query:

Sql:

```
SELECT Department, Gender, COUNT(*) AS
Total Patients.
 FROM Patients
 GROUP BY Department, Gender;
```

Department	Gender	Total Patients
Cardiology	Male	1
Neurology	Female	1
orthopedics	Male	1
Pediatrics	Female	1
Dermatology	Male	1

8. Find doctors and their department details using GROUP BY and ORDER BY

Query:

Sql:

```
SELECT Doctor name, Department, count (*) AS
Count
FROM Doctors
GROUP BY Doctorname, Department
ORDER BY Doctor name;
```

Doctor name	Department	Count
Dr. Ahmed	Neurology	1
Dr. Neha	Dermatology	1
Dr. Priya	Pediatrics	1
Dr. Rajesh	Orthopedics	1
Dr. Sharma	Cardiology	1

VEL TECH	
EX No.	
PERFORMANCE (5)	3
RESULT AND ANALYSIS (5)	6
VIVA VOCE (5)	5
RECORD (5)	3
TOTAL (20)	-
SIGN WITH DATE	13

Result: The Implementation of DML Commands using clauses, generatio and functions in Queries executed successfully.

Task 3.2 Aggregate functions (multi Row operations)

Aim: To study and implement aggregate functions (COUNT (), SUM(), AVG(), MIN(), MAX ()) on a sample patient database.

Procedure:

1. Create a table named students
2. Insert sample records.
3. write queries using aggregate functions.
4. Observe and record the Output.

Commands with Explanation:

Example Table: Patients.

Patient ID	Patient Name	Department	Bill Amount.
101	Arun	Cardiology	2000
102	Sneha	Neurology	3500
103	Karan	Orthopedics	1500
104	Meena	Pediatrics	4000
105	Rohan	Dermatology	

1) Count the total number of Patient:

Sql:-

```
SELECT COUNT (*) AS Total - Patients
FROM Patients ;
```

Output:

Total _ Patients
5

2) Find the highest bill amount

Sql:

```
SELECT MAX (Bill Amount) As Highest_Bill
FROM Patients;
```

Output:

Highest_Bill
4000

3) Find the average bill amount of patients

Sql:

```
SELECT AVG(Bill Amount) AS Average_Bill
FROM Patients;
```

Output:

Average_Bill
2700

4) Find the minimum bill amount among patients in Neurology department.

Sql:

```
SELECT MIN(Bill Amount) AS Min_Neuro_Bill
```

Output:

Min_Neuro_Bill
3500

5) Find the total bill amount by each department

Sql:

```
SELECT Department, SUM (Bill Amount) AS
Total_Bill
FROM Patients
GROUP BY Department;
```

output :

Department	Total-Bill
Cardiology	2000
Neurology	3500
Orthopedics	1500
Pedeatria	4000
Dermato logy	

6) Find the average bill per department, ordered by average descending.

Sql:-

```
SELECT Department, AVG (Bill Amount) AS AVG Bill
FROM Patients
GROUP BY Department
ORDER BY Avg-Bill DESC;
```

output:

Department	Avg-Bill
Pedeatrics	4000
Neurology	3800
Dermatology	2500
Cardiology	2000
Orthopedia	1500

VEL TECH	
EX No	3.2
PERFORMANCE (5)	5.
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	-
TOTAL (20)	13
SIGN WITH DATE	1/9/25

Result:- The Implementation of Aggregate functions are executed successfully.