25/8/25

TASK:3.1 DML commands using clauses operators and functions in Queries

Aim:

To implement of DML commands using clauses, operators and functions in queries

Data manipulation 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

1. INSERT
2. UPDATE
3. DELETE

INSERT INTO:

This is used to add records into a relation

Syntax:

```
INSERT INTO <table_name> (field1, field2 ... fieldN)
VALUES (data 1, data 2, ... data N);
```

Example:

SQL:

```
INSERT INTO patients VALUES (111, 'ARUN',
'Cardiology', 'Male');
```

Patient ID	Patient Name	Department	Gender
111	Arunsen	Cardiology	male

UPDATE-SET-WHERE:

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

Syntax:

UPDATE table_name SET field1 = data WHERE condition;

Example:

sql:

UPDATE Patients SET PatientName = 'Kumar' WHERE PatientID = 111;

Table after UPDATE:

PatientID	PatientName	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 DELTE:

AppointmentID	PatientID	DoctorID	AppointmentName	Duration

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	Doctor Name	Department	Fee
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	Patient Name	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 PatientID
BETWEEN 111 AND 115;
```

PatientName	Department	Gender
Arun	Cardiology	Male
Karan	Orthopedics	Male
Rohan	Dermatology	Male

2. List doctors where consultation fees between 700 and 800.

query:

sol:

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

Doctor ID	Doctor Name	Department Name	Fees
202	Dr. priya	Pediatrics	700
205	Dr. Neha	Dermatology	800

3. Find the record with minimum appointment duration

Query:

sol:

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

Min(Duration)
20

4. Find appointment with date >= '2023-02-07'

query:

sol:

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

Appointment ID	Patient ID	Doctor ID	Appointment Date	Duration
302	112	203	2023-02-07	45
303	113	204	2023-0209	20
304	114	202	2023-02-10	60
305	115	205	2023-02-12	25

5. List distinct Patient IDs

Query:

Sol:

```
SELECT DISTINCT Patient ID FROM patients;
```

Patient ID
111
112
113
114
115

6. Combine Patient IDs from Patients and Appointment (UNION)

Query:

Sol:

```
SELECT PatientID FROM Patients
UNION
SELECT Patient ID FROM Appointments;
```

Patient ID
111
112
113
114
115

7. Group Patients based on gender and department

Query:

Sol:

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

Department	Grender	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:

```
SELECT DoctorName , Department , COUNT(*) AS count
FROM Doctor
GROUP BY DoctorName, Department
ORDER BY DoctorName;
```

DoctorName	Department	count
Dr. Ahmed	Neurology	1
Dr. Neha	Dermatology	1
Dr. Priya	Pediatrics	1
Dr. Rajesh	Orthopedics	1
Dr. Sharma	Cardiology	1

VEL TECH - CSE	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15
	25/3/25

Result:

Thus the implementation of DML Commands using clauses, operators and functions in queries executed successfully.

25/8/25

TASK: 3.2 Aggregate function (Multi Row operations)

Aim:

To implement Aggregat functions

Procedure:

1. Create a table named Students
2. Insert Sample record
3. Write queries using aggregate functions
4. observe and record the output.

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	

COMMADS WITH EXPLATION:

1) Count the total number of patient

query:

```
SELECT COUNT(*) AS Total_Patients
FROM Patients;
```

output:

Total_Patients
5

2) Find the highest bill amount

query:

```
SELECT MAX(BILLAMOUNT) AS Highest_Bill
FROM Patients;
```

Output:

Highest_Bill
4000

3) Find the average bill amount of patients

query:

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

output:

Average_Bill
2700

4) Find the minimum bill amount among patients in Nuerology department

query:

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

output:

Min_Neuro_Bill
3500

5. Find the total bill amount by each department

query:

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

Output:

Department	Total_Bill
Cardiology	2000
Neurology	3500
Orthopedics	1500
Preadatrics	4000
Permatology	

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

query:

```
SELECT Department, AVG(Bill Amount) AS AVG_Bill
FROM Patients
GROUP_BY Department
ORDER_BY AVG_Bill DESC;
```

output:

Department	Avg_Bill
Pediatrics	4000
Neurology	3500
Dermatology	2500
Cardiology	2000
Orthopedics	1500

VEL TECH - CSE	
EX NO.	3.2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	15
SIGN WITH DATE	M 28/8/25

Result:
The implimentation of Aggregate functions are executed successfully