18/8/25

TASK : 3.1 DML Commands using operators and functions. Queries.

Aim :

To implement of DML commands 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 at the basic DML command.

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 (data1, data2 .... dataN);

Example:

sql:

INSERT INTO, patients VALUES (111;
'Arun', 'cardiology', 'male');

Table after INSERT:

patient ID	patient name	Dept	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 field1 = data
WHERE condition;

example:

sql:

UPDATE patients SET patientName = 'kumar'
WHERE patient ID = 111;

Table after UPDATE:

patient ID	patient name	Dept	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:

DELETE FROM Appointments;

Appointments table after DELETE:

Appointment ID	patient ID

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 Doctor ID = 202;

Doctors table after DELETE:

Doctor ID	Doctor Name	Dept	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 keeps the table structure.

Syntax:

TRUNCATE TABLE < table_name >;

Example:

sql:

TRUNCATE TABLE patients;

patients table after TRUNCATE:

patient ID	patientName	Dept	Gender.

sample Queries and output:

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

Query:

SQL:

```
SELECT patientname, Dept, Gender.
FROM patients.
WHERE patientname LIKE %.'n' AND
patient ID BETWEEN 111 AND 115.
```

patient name	Dept	Gender.
Arun	cardiology	male
Karan	orthopedics	male
Rohan	Dermatology	male.

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

query:

Sql:

SELECT * FROM Doctors WHERE Fees BETWEEN 700 AND 800.

Doctor ID	Doctor Name	Deptname	Fees
202	Dr. priya	Pediatria	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 >= '2023-2-7.

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 patientID FROM patients
```

patient ID

111
112
113
114
115.

6. Combine patient IDs from patients and Appointments (UNION)

Query:

SQL:

```
SELECT patientID FROM patients.
UNION
SELECT patient ID from Appointment.
```

output:

patient ID
111
112
113
114
116.

7. Group patients based on gender and dept.

Query:

sql:

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

Dept	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 Doctorname, Department, COUNT(*) AS
count
FROM Doctors
GROUP BY Doctorname, Department
ORDER BY Doctor Name;
```

Doctorname	Department	count.
Dr. Ahmed	newology	1
Dr. Neha	Dermatology	1
Dr. priya	pediatrian	1
Dr. Rajesh	orthopedic	1
Dr. Sharma	Cardiology	1

VEL TECH - CSE	3.1
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	—
RECORD (5)	15
TOTAL (20)	
SIGN WITH DATE	25/8/25

Result:

The implementation of DML commands using clauses, joins and functions in Queries executed successfully.