8/9/25

Task:5 Writing Join quries, Equivalent, AND/OR Recursive queries

Aim:
To implement and execute JOIN queries, equivalent quiries and recursive queries

Procedure:

- Create the database and tables
- Insert sample data
- write SQL queries using different types of JOINs
- write equivalent quiris (different approach to get the same result)
- Implement a recursive query (using WITH RECURSIVE).
- Display results and verify Corredness

Different types of SQL JOINS

- (INNER) JOIN:
  tables SELECT column_name(s) FROM table1 INNER JOIN table2 ON table1.column_name=tale2.column_name;

- LEFT (OUTER) JOIN:
  SELECT column_name(s) FROM table1 LEFT JOIN table2 ON table1.column_name = tale2.column_name;

- RIGHT (OUTER) JOIN:
  SELECT column_names(s) FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.colun_name;

- FULL (OUTER) JOIN:
  FULL OUTER JOIN table 2 ON table1.column_name= table 2.column_name;

1. JOIN Queries (All types)

```
CREATE TABLE departments(
      DEPT ID INT PRIMARY KEY,
      DeptName VARCHAR(50)
  );

CREATE TABLE Patients (
      PatID INT PRIMARY KEY,
      PatName VARCHAR (50),
      DeptID INT,
      FOREIGN KEY (DeptID) REFERENCES Departments
                                            (DeptID)
  );
CREATE TABLE Bills(
      BillID INT PRIMARY KEY,
      PatID INT,
      BillAmount DECIMAL (10,2),
      FOREIGN KEY (PatID) REFERENCES Patients(PatID)
  );
CREATE TABLE Referrals (
      ReferralID INT PRIMARY KEY,
      ReferringPatID INT,
      ReferredPat ID INT,
      FOREIGN KEY (ReferringPatID) REFERENCES
                                    patients (PatID)
  );
```

2 Departments:

Dept ID	DeptName

Patients

PatID	PatName	Dept ID

Table Bills

BillID	PatID	BillAmount

Referrals

Referral ID	ReferringPat ID	ReferredPat ID

2. INSERT SAMPLE DATA

```
INSERT INTO departments VALUES
  (101, 'cardiology'), (102, 'oncology'),
  (103, 'pediatrics');
```

Dept ID	DeptName
101	cardiology
102	oncology
103	pediatrics

```
INSERT INTO patients VALUES
(1, 'Alice', 101),
(2, 'Bob', 102),
(3, 'charlie', 101),
(4, 'David', 103),
(5, 'Emma', 104);
```

Pat ID	Pat Name	Dept ID
1	Alice	101
2	Bob	102
3	charlie	101
4	David	103
5	Emma	104

```
INSERT INTO Bills VALUES
 (1, 1, 5000.00),
 (2, 1, 300.00),
 (3, 2, 7500.00),
 (4, 3, 6000.00);
```

Bill ID	Pat ID	BillAmount
1	1	5000.00
2	1	300.00
3	2	7500.00
4	3	6000.00

```
INSERT INTO Referrals VALUES
(1,1,3),
(2,3,4);
```

ReferralID	ReferringPat ID	RefferedPat ID
1	1	3
2	3	4

3. JOIN Queries (ALL TYPES)

a) INNER JOIN

query:

```
SELECT p.PatName, d.DeptName
FROM Patients p
INNER JOIN Departments d ON p.DeptID=d.DeptID;
```

output:

PatName	DeptName
Alice	Cardiology
Bob	oncology
Charlie	Cardiology
David	Pediatrics

b) LEFT JOIN

query:

```
SELECT p.PatName, d.DeptName
FROM patients p
LEFT JOIN Departments d ON p.DeptID=d.DeptID;
```

output:

PatName	DeptName
Alice	Cardiology
Bob	oncology
Charlie	Cardiology
David	Pediatrics
Emma	NULL

c, Right JOIN:

query:

```
SELECT  p.PatName, d.DeptName
FROM  Patients p
RIGHT  JOIN Departments d ON p.DeptID = d.DeptID
```

output:

PatName	DeptName
Alice	cardiology
Bob	oncology
charlie	cardiology
David	Pediatria

d, FULL OUTER JOIN

query:

```
SELECT     P.PatName , d.DeptName
FROM       Patients P
FULL OUTER JOIN Departments d ON P.DeptID=d.DeptID;
```

output:

PatName	DeptName
Alice	cardiology
Bob	oncology
charlie	Cardiology
David	Pediatria
Emma	NULL

e, CROSS JOIN

query:

```
SELECT P.PatName , d.DeptName
FROM Patients P
CROSS JOIN Departments d;
```

output:

PatName	DeptName
Alice	Cardiology
Alice	onclogy
Alice	Pediatrics
Bob	Cardiology
Bob	oncology
Bob	Pediatria
Charlie	Cardiology
charlie	oncology
charlie	pediatrics
David	Cardiology
David	oncology
David	Pediatrics
Emma	Cardiology
Emma	oncology
Emma	Pediatrics

d) Self JOIN:

Query.

```
SELECT P1.PatName AS Patient1, P2.PatName AS
                                   patients2, P1.DeptID
FROM patients P1
JOIN Patients P2 ON P1.DeptID = P2.DeptID
WHERE P1.PatID < P2.PatID;
```

OUTPUT

Patient1	Patient2	DeptID
Alice	charlie	101

4. Equivalent queries:

JOIN Vs. Subquery

using JOIN:

```
SELECT p.PatName , d.DeptName
FROM Patients p
JOIN Departments d ON p.Dept ID= d.DeptID;
```

using a subquery

```
SELECT PatName,
       (SELECT DeptName FROM Department d WHERE
                      d.DeptID = p.DeptID) AS DeptName
FROM Patients p;
```

output:

PatName	DeptName
Alice	Cardiology
Bob	oncology
charlie	cardiology
David	pediatrics

5. Recursive query

query:

```
WITH RECURSIVE Referralchain AS(
     SELECT ReferringPatID, Referred Pat
     FROM Referrals
     UNION
     SELECT r.ReferringPatID, rc.ReferredPatIn
     FROM Referrals r
     JOIN Referralchain rc ON r.refferredPatID
                          = rc.referringPatID
  )
```

SELECT rc.ReferringPatID AS InitialReferrer,
rc ReferredPatIDAs ReferredPatient,
P1.PatName AS InitialReferrerName, P2.PatName AS ReferredPatientName

FROM Referralchain rc
JOIN Patients P1 ON rc.ReferringPatID = P1.PatID
JOIN Patients P2 ON rc.RefferedPatID = P2.PatID;

Initial reffer	referredpatient	Initial referrer Name	referred Patient Name
1	3	Alice	Charlie
3	4	charlie	David
1	4	Alice	David

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

3/8/25

Result:
Thus the implementation of JOIN Queries, equivalent queries and recursive queries are executed successfully