08/09/25 Task 5

Join Queries , equivalent And Recursive Queries

Aim: To implement and execute joins , equivalent queries and Recursive Queries in SQL.

Procedure:

1. Create table DEPARTMENT & STUDENT
2. Insert the values into tables.
3. Perform join operation
4. Perform equivalent & Recursive Query.
5. Display result.

```
CREATE .TABLE DEPARTMENT(
               DEPT ID INT PRIMARY KEY,
               DEPTNAME VARCHAR (50));

CREATE TABLE STUDENT(
          STU_ID INT PRIMARYKEY,
          NAME VARCHAR (50),
          AGE INT,
          DEPT ID INT,
          FOREIGN KEY (DEPT ID)
          REFERENCES DEPARTMENT ((DEPTID).
          );
INSERT INTO DEPARTMENT VALUES.
(201, 'computer science'),
(202, 'Electronics'),
(203, 'Mechanical');
```

```
INSERT INTO STUDENT VALUES.
(1, 'Ravi', 20, 201),
(2, 'Sneha', 22, 201),
(3, 'Amit', 19, 202),
(4, 'Priya', 24, 203),
(5, 'Kiran', 23, 201);
```

```
SELECT * FROM DEPARTMENT;
```

	DEPTID	DEPTNAME
1	201	Computer Science
2	202	Electronics
3	203	Mechanical

```
SELECT * FROM STUDENT;
```

	STUID	NAME	AGE	DEPTID
1	1	Ravi	20	201
2	2	Sneha	22	201
3	3	Amit	19	202
4	4	Priya	24	203
5	5	Kiran	23	201

```
SELECT S.NAME, S.AGE, D.DEPTNAME
FROM STUDENT
```

```
SELECT S.NAME, S. AGE, D.DEPTNAME
FROM STUDENT 4 S
INNER JOIN ·DEPARTMENT 4 D
ON S.DEPTID = D. DEPT ID;
-- INNER JOIN
```

	NAME	AGE	DEPTNAME
1.	Ravi	20	Computer Science
2.	Sneha	22	Computer Science
3.	Amit	19	Electronics
4.	Priya	24	Mechanics
5.	Kiran	23	Computer Science,

```
-- LEFT OUTER JOIN.
SELECT S.NAME, S.AGE, D.DEPTNAME
FROM STUDENT 4 S.
LEFT JOIN DEPARTMENT 4 D.
ON S.DEPT ID= D.DEPTID;
```

	Name	Age	Dept name
1.	Ravi	20	computer science
2.	Sneha	22	computer science
3.	Amit	19	Electronics
4.	Priya	24	Mechanics
5.	Kiran	21	computer science.

```
SELECT S.NAME, S.AGE, D.DEPARTMENTNAME
FROM STUDENT4S
RIGHT JOIN DEPARTMENT4D
ON S.DEPTID = D.DEPTID;
```

	Name	Age	DEPTNAME
1.	Ravi	20	Computer science
2.	Sneha	22	Computer Science
3.	Kiran	23	Computer Science
4.	Amit	19	Electronics
5.	Priya	24	Mechanical

```
SELECT TOP 3 S.NAME, S.AGE, D.DEPTNAME
FROM STUDENT4S
FULL OUTER JOIN DEPARTMENT4D
ON. S.DEPTID = D.DEPTID;
```

	Name	Age	DEPTname
1	Ravi	20	computer Science
2.	Sneha	22	Computer Science
3.	Amit	19	electronics.

```
-- EQUIVALENT QUERIES.
-- USING JOIN
SELECT S.NAME, S.AGE
FROM STUDENT4S
JOIN DEPARTMENT4D ON S.DEPTID = D.DEPTID
WHERE D.DEPTNAME = "computer science";
```

	Name	Age
1.	Ravi	20
2.	Sneha	22
3.	Kiran	23

```
-- RECURSING QUERIES

WITH COUNTCTE AS(
        SELECT 1 AS N
        UNION ALL
        SELECT N+1
        FROM COUNTCTE
        WHERE N<5
    )
 SELECT * FROM COUNTCTE;
```

```
    N
1   1
2   2
3   3
4   4
5   5
```

EX No.	
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	-
SIGN WITH DATE	15

8/9/25

RESULT: Thus, implementation of Join Quires, Equivale and Recursive Quires has succesfully executed and verified.