

| | | | | |
|---|----------|--|----|--------|
| 5 | 08/09/25 | Join Queries, Equivalent and Recursive Queries | 14 | 8/5/25 |
|---|----------|--|----|--------|

8/1/25

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,  
    DEPT NAME VARCHAR (50));
```

```
CREATE TABLE STUDENT (  
    STUD ID INT PRIMARY KEY,  
    NAME VARCHAR (50),  
    AGE INT,  
    DEPT ID INT,  
    FOREIGN KEY (DEPT ID)  
    REFERENCES DEPARTMENT (DEPT ID)  
);
```

```
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 4;

| | DEPT ID | DEPT NAME |
|---|---------|------------------|
| 1 | 201 | Computer Science |
| 2 | 202 | Electronics |
| 3 | 203 | Mechanical |

SELECT * FROM STUDENT 4;

| | STU-ID | NAME | AGE | DEPT ID |
|---|--------|-------|-----|---------|
| 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 SNAME, S.AGE, D.DEPT NAME
FROM STUDENT 4 S
INNER JOIN DEPARTMENT 4 D
ON S.DEPT ID = D.DEPT ID;
-- INNER JOIN

| | NAME | AGE | DEPT NAME |
|---|-------|-----|------------------|
| 1 | Ravi | 20 | Computer Science |
| 2 | Sneha | 22 | Computer Science |
| 3 | Amit | 19 | Electronics |
| 4 | Priya | 24 | Mechanical |
| 5 | Kiran | 23 | Computer Science |

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

| | NAME | AGE | DEPT NAME |
|---|-------|-----|------------------|
| 1 | Ravi | 20 | Computer Science |
| 2 | Sneha | 22 | Computer Science |
| 3 | Amit | 19 | Electronics |
| 4 | Priya | 24 | Mechanical |
| 5 | Kiran | 23 | Computer Science |


```

SELECT NAME S.NAME, S.AGE, D.DEPTNAME
FROM STUDENT4 S
RIGHT JOIN DEPARTMENT4 D
ON S.DEPT ID = D.DEPT ID;

```

| | 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 TOP3 S.NAME, S.AGE, D.DEPTNAME
FROM STUDENT4 S
FULL OUTER JOIN DEPARTMENT4 D
ON S.DEPT ID = D.DEPT ID;

```

| | NAME | AGE | DEPT NAME |
|---|-------|-----|------------------|
| 1 | Ravi | 20 | Computer Science |
| 2 | Sneha | 22 | Computer Science |
| 3 | Amit | 19 | Electronics |

-- EQUIVALENT QUERIES

-- USING JOIN

```

SELECT S.NAME, S.AGE
FROM STUDENT4 S
JOIN DEPARTMENT4 D ON S.DEPT ID = D.DEPT ID
WHERE D.DEPT NAME = 'Computer Science';

```

| | NAME | AGE |
|---|-------|-----|
| 1 | Ravi | 20 |
| 2 | Sneha | 22 |
| 3 | Kiran | 23 |

-- RECURSIVE QUERIES

WITH COUNT CTE ASC

```

SELECT ASN
UNION ALL
SELECT N+1

```

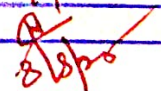
FROM COUNT CTE

WHERE N < 5

)

SELECT * FROM COUNT CTE;

| | N |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

| VEL TECH | |
|-------------------------|---|
| EX No. | 5 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 4 |
| RECORD (5) | 1 |
| TOTAL (20) | 14 |
| SIGN WITH DATE |  |

Result: Thus, implementation of Join, queries, Equivalent and Recursive Queries has been successfully executed and verified.