08/9/25

5. Join Quires, Equivalent and Recursive wires.

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 ques
5) Display result.

```
CREATE TABLE DEPARTMENT
    DEPT ID INT PRIMARY KEY;
    DEPT NAME VARCHAR (50));

CREATE TABLE STUDENT (
    STUID INT PRIMARY KEY
    NAME VARCHAR (50)
    AGE INT,
    DEPT ID, INT,
    FOREIGN KEY (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', 23, 202),
        (4, 'Priya' 24, 203),
        (5, 'Kiran' 23, 201),
```

SELECT * FROM DEPARTMENT

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

```
SELECT S-NAME, S-AGE, DEPTNAME,
FROM STUDENT,
ON S.DEPT ID = D.DEPT JD;
```

	NAME	AGE	DEPT NAME
1	Ravi	20	computer Science
2	Sneha	22	computer Sciene
3	Amit	19	Electronics
4	Priya	24	mechanical
5	Kiran	23	computerScience

--LEFT OUTER JOIN

```
SELECT S.NAME , S.AGE , D.DEPTNAME
FROM STUDENT
LEFT JOIN DEPARTMENT 4D;
```

	NAME	AGE	DEPARTNAME
1	RAVI	20	computer Science
2	Sneha	22	Computer Sciene
3	Amit	19	Electronics
4	Priya	24	mechanical
5	Kiran	23	ComputerScience

```
SELECT S.NAME, S.Age . D. DEPARTMENT
FROM . STUDENT AS
RIGHT JOIN DEPARTMENT ID
ON S. DEPT JD = D. DEPT JD;
```

	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.DEPARTMEN

FROM STUDENT AS S.

FULL OUTER JOIN DEPARTMENT AS D

ON S.DEPT ID : D.DEPT ID',

	NAME	AGE	DEPT NAME
1	Ravi	20	Computer Scien
2.	sneha	12	computer Science
3.	Amit	19	Electronics

SELECT S.NAME >AGE

FROM STUDENT AS.

JOIN DEPART MENT AS ON S. DEPT ID =
D.DEPT ID

WHERE D.DEPT NAME = "computer Scien

	STUDENT NAME	AGE
1	Raul	20
2	Sneha	22
3	Kiran	23

-- Recursive Queries

```
With COUNT (
    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 - CSE	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	
TOTAL (20)	12
SIGN WITH DATE	8/9/25

Result:

Thus, implement of Join, lines equivalent and recursive Queries has successfully executed & verified.