

TASKSOBJECTIVES

JOIN QUERIES, EQUIVALENT, AND RECURSIVE

Aim:- To implement and execute join equivalent queries and recursive queries in SQL.

procedure:-

1. Create table department & student 4
2. Insert the values into tables
3. perform join operation.
4. perform equivalent & Recursive Query
5. Display result.

CREATE TABLE DEPARTMENT 4 (

DEPT ID INT PRIMARY KEY ;

DEPT NAME VARCHAR (50) ;

CREATE TABLE STUDENT 4 (

STU - ID INT PRIMARY KEY ,

NAME VARCHAR (50)

AGE INT ,

DEPT ID INT ,

FOREIGN KEY (DEPT ID)

REFERENCES DEPARTMENT4(DEPTU

J.J.

INSERT INTO DEPARTMENT 4 VALUES
 (201, 'computer science'),
 (202, 'electronics'),
 (203, 'mechanical'),

INSERT INTO STUDENT 4 VALUES

(1, 'RAVI', 20, 201),
 (2, 'Sneha', 22, 201),
 (3, 'Amit', 19, 202),
 (4, 'priya', 24, 203);

SELECT FROM DEPARTMENT 4;

| S.NO | DEPT ID | DEPT NAME |
|------|---------|------------------|
| 1 | 201 | Computer Science |
| 2 | 202 | Electronics |
| 3 | 203 | Mechanical |

SELECT FROM STUDENT 4;

| 8.00 | STIP. | 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 | 3 | Kiran | 23 | 201 |

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

| S.NO | NAME | AGE | DEPT NAME |
|------|-------|-----|-------------------|
| 1 | Ravi | 20 | Computer Sciences |
| 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;

| S.NO | NAME | AGE | DEPT NAME |
|------|-------|-----|------------------|
| 1 | Ravi | 20 | Computer Science |
| 2 | Sneha | 22 | Computer Science |
| 3 | Amit | 19 | Electronics |
| 4 | Priya | 24 | Mechanics |
| 5 | Kiran | 23 | Computer Science |

SELECT S.NAME, S.AGE, D.DEPART NAME
FROM STUDENT 4S.

, RIGHT JOIN DEPARTMENT 4D
ON S. DEPT ID = D. DEPT ID;

| S.NO | Name | Age | Dept Name |
|------|-------|-----|------------------|
| 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.DEPART NAME
FROM STUDENT 4S

Full Outer Join Department 4D
ON S. DEPT ID = D. DEPT ID;

| S.NO | NAME | AGE | DEPT NAME |
|------|-------|-----|------------------|
| 1 | RAVI | 20 | Computer Science |
| 2 | Sneha | 22 | Computer Science |
| 3 | Amit | 19 | Electronics |

-- E. EQUIVALENT QUERIES

-- USING JOIN

SELECT S.NAME, S.AGE

JOIN DEPARTMENT D ON S.DEPT_ID = D.DEPT_ID

WHERE D.DEPT_NAME = 'Computer

Science';

| S.NO | NAME | AGE |
|------|-------|-----|
| 1 | RAVI | 20 |
| 2 | Sneha | 22 |
| 3 | Kiran | 23 |

-- RECURSIVE QUERIES

WITH COUNT CTE AS

SELECT ASN

UNION ALL

SELECT N+1

FROM COUNT CTE

WHERE N < S

)

SELECT * FROM COUNT CTE;

| S.NO | N |
|------|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

| VEL TECH - CSE | |
|-------------------------|----|
| EX NO. | 5 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 4 |
| RECORD (5) | — |
| TOTAL (20) | 14 |
| SIGN WITH DATE | ✓ |

8/19

Result:- Thus, implementation of join queries
equivalent and Recursive queries has
successfully executed and verified