9/9/25

Task 5 - Working Join Queries equivalent and for Recursive Queries

Aim: To Implement of different types of queries using Employee scenario.

SQL Join clause

Step 1:

Syntax:

```
SELECT column1, column2, column3..
FROM table_name1, table_name2
Where table_name1.column name = table_name2.columnname;
```

Types of joins:

1. simple Join
2. outer and self join

Simple Join:

Query

```
Select * from item, cust where emp_id = cust.id;
```

SELF Join

Query

```
Select * from emp x, emp y where x.salary >= (select avg(salary) from x.emp where x.deptno
```

Outer join:

Extends the result of simple join.

Types of join

Inner join: Return Records

Query

```
FROM Employee1
INNER join Employee2 ON
Employee1. column emp_id
Employee2. Dept_id;
```

LEFT outer join:

Query

```
SELECT Employee_name(s) FROM
Employee1. columnname.
Employee2. column_name.
```

Right outer Join:

Query

```
SELECT column_name FROM table1
Right join table2 on
table1. column_name
table2. column_name;
```

Full outer join

Syntax

```
SELECT column_name FROM table1
FULL outer join table2 ON
table1.column_name
table2.column_name;
```

SQL Join clause

```
CREATE TABLE Employees(
    EmployeeID INT PRIMARY KEY,
    EmployeeName VARCHAR(50),
    DeptID INT,
    ManagerID INT,
    Foreign key (DeptID)
CREATE TABLE projects(
    ProjectID VARCHAR(10), PRIMARY KEY,
    ProjectName VARCHAR(50),
    DeptID INT,
    FOREIGN KEY (Dept;
(CREATE TABLE Assignments
    AssignmentID INT PRIMARY KEY,
    EmployeeID, INT,
    PROJECTID VARLHAR(10);
CREATE Table Hierarchy(
    Employee INT,
);, ManagerID INT
```

```
Insert into Departments
(201, 'Technology')
(202, 'Human Resources'),
(203, Marketing');
Insert into Employees
(1, 'Alice', 201, 3),
(2, 'Bob', 202, 4),
(3, 'Charlie' 201, NULL),
(4, Daved, 202, NULL);
Insert into projects values
('P1', Website Redesign', 201),
('P2', Hiring, 202),
('P3', API Development', 201)
('P4', Social Media campaign', 203);
Insert into Hebrarchy values.
(1,3),
(2,4);
SELECT * FROM Employeehierarchy;
```

VEL TECH	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	—
TOTAL (20)	15
SIGN WITH DATE	

Result: The Implementation of SQL commands for Employee using Joins and recursive queries successfully