

# University Institute of Engineering Department of Computer Science & Engineering

#### **EXPERIMENT: 2**

NAME: CHARU SINGLA UID: 23BCS12553

BRANCH: BE-CSE SECTION/GROUP: KRG-1-A

SEMESTER: 5<sup>TH</sup> SUBJECT CODE: 23CSP-339

**SUBJECT NAME: ADBMS** 

## 1. Aim Of The Practical:

[ MEDIUM ]

You are a Database Engineer at TalentTree Inc., an enterprise HR analytics platform that stores employee data, including their reporting relationships. The company maintains a centralized Employee relation that holds:

Each employee's ID, name, department, and manager ID (who is also an employee in the same table).

Your task is to generate a report that maps employees to their respective managers, showing:

- The employee's name and department
- Their manager's name and department (if applicable)
- This will help the HR department visualize the internal reporting hierarchy.

## [HARD]

To write SQL queries that retrieve requested Net Present Value (NPV) data by performing conditional joins on multiple columns (ID and YEAR) across two related tables, and to handle missing data using appropriate SQL functions.

2. Tools Used: SQL Server Management Studio

## 3. Code:

## [ MEDIUM ]

CREATE TABLE Employee (EmpID INT PRIMARY KEY, EmpName VARCHAR(50) NOT NULL, Department VARCHAR(50) NOT NULL, ManagerID INT NULL);

```
ALTER TABLE Employee ADD CONSTRAINT FK_Manager FOREIGN KEY
(ManagerID) REFERENCES Employee(EmpID);
INSERT INTO Employee (EmpID, EmpName, Department, ManagerID) VALUES
(1, 'Alice', 'HR', NULL),
(2, 'Bob', 'Finance', 1),
(3, 'Charlie', 'IT', 1),
(4, 'David', 'Finance', 2),
(5, 'Eve', 'IT', 3),
(6, 'Frank', 'HR', 1);
SELECT E.EmpName AS EmployeeName, E.Department AS EmployeeDept,
M.EmpName AS ManagerName, M.Department AS ManagerDept
FROM Employee E
LEFT JOIN Employee M
ON E.ManagerID = M.EmpID;
[HARD]
CREATE TABLE Year_tbl (id INT, year INT, NPV INT);
INSERT INTO Year_tbl (id, year, NPV) VALUES
(1, 2018, 100),
(7, 2020, 30),
(13, 2019, 40),
(1, 2019, 113),
(2, 2008, 121),
(3, 2009, 12),
(11, 2020, 99),
(7, 2019, 0);
```

CREATE TABLE Queries\_tbl (id INT, year INT);

```
INSERT INTO Queries_tbl (id, year) VALUES (1, 2019), (2, 2008), (3, 2009), (7, 2018), (7, 2019), (7, 2020), (13, 2019);
```

SELECT Y.id AS ID, Y.year AS Year, ISNULL(Q.NPV, 0) AS NPV FROM Queries\_tbl AS Y LEFT OUTER JOIN Year\_tbl AS Q ON Y.id = Q.id AND Y.year = Q.year;

# 4. Output:

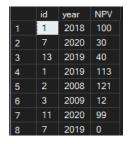
# [ MEDIUM ]

|   | EmplD | EmpName | Department | ManagerID |
|---|-------|---------|------------|-----------|
| 1 | 1     | Alice   | HR         | NULL      |
| 2 | 2     | Bob     | Finance    | 1         |
| 3 | 3     | Charlie | IT         | 1         |
| 4 | 4     | David   | Finance    | 2         |
| 5 | 5     | Eve     | IT         | 3         |
| 6 | 6     | Frank   | HR         | 1         |

|   | EmployeeName | EmployeeDept | ManagerName | ManagerDept |
|---|--------------|--------------|-------------|-------------|
| 1 | Alice        | HR           | NULL        | NULL        |
| 2 | Bob          | Finance      | Alice       | HR          |
| 3 | Charlie      | IT           | Alice       | HR          |
| 4 | David        | Finance      | Bob         | Finance     |
| 5 | Eve          | IT           | Charlie     | IT          |
| 6 | Frank        | HR           | Alice       | HR          |

# [HARD]

|   | id | year |
|---|----|------|
| 1 | 1  | 2019 |
| 2 | 2  | 2008 |
| 3 | 3  | 2009 |
| 4 | 7  | 2018 |
| 5 | 7  | 2019 |
| 6 | 7  | 2020 |
| 7 | 13 | 2019 |



|   | ID | Year | NPV |
|---|----|------|-----|
| 1 | 1  | 2019 | 113 |
| 2 | 2  | 2008 | 121 |
| 3 | 3  | 2009 | 12  |
| 4 | 7  | 2018 | 0   |
| 5 | 7  | 2019 | 0   |
| 6 | 7  | 2020 | 30  |
| 7 | 13 | 2019 | 40  |

## 5. Learning Outcomes:

- Understand and implement self-joins to model hierarchical relationships within a single table (e.g., employees reporting to other employees).
- Construct relational queries to fetch meaningful information such as employeemanager relationships, including handling NULL values using LEFT JOIN.
- Design and populate tables using the CREATE TABLE and INSERT INTO statements for real-world hierarchical and time-series data scenarios.
- Perform multi-table joins to retrieve and match data across different datasets, such as actual vs. requested values (e.g., NPV values for specific years).
- Handle missing data using functions like ISNULL() to substitute default values during join operations.
- Apply conditional joins involving multiple keys (e.g., joining on both ID and YEAR) to ensure accurate data mapping.
- Develop problem-solving approaches using SQL to derive insights from HR records and financial datasets in enterprise applications.