# University Institute of Engineering

# Department of Computer Science & Engineering

EXPERIMENT : 2

NAME : SARTHAK UID: 23BCS11978

BRANCH : BE-CSE SECTION/GROUP : KRG_2A

SEMESTER : 5TH 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 EmployeeTable (
   EmployeeID INT PRIMARY KEY,
   EmployeeName VARCHAR(25),
   Dept VARCHAR(25),
   ManagerID INT
);

INSERT INTO EmployeeTable (EmployeeID, EmployeeName, Dept, ManagerID) VALUES
(1, 'Sarthak', 'HR', NULL),
(2, 'Reyansh', 'Finance', 1),
(3, 'Yogesh', 'IT', 1),
(4, 'Pravi', 'Finance', 2),
(5, 'Surya', 'IT', 3),
(6, 'MD', 'HR', 1);

SELECT * FROM EmployeeTable;

SELECT
   E.EmployeeName AS [Employee_Name],
   M.EmployeeName AS [Manager_Name],
   E.Dept AS [Employee_Dept],
   M.Dept AS [Manager_Dept]
FROM EmployeeTable E
LEFT JOIN EmployeeTable M
ON E.ManagerID = M.EmployeeID;


---------- HARD ----------

CREATE TABLE YearData (
   RecordID INT,
   YearVal INT,
   NetPresentValue INT
);

CREATE TABLE RequestList (
   RecordID INT,
   YearVal INT
);

INSERT INTO YearData (RecordID, YearVal, NetPresentValue) 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);

SELECT * FROM YearData;

INSERT INTO RequestList (RecordID, YearVal) VALUES
(1, 2019),
(2, 2008),
(3, 2009),
(7, 2018),
(7, 2019),
(7, 2020),
(13, 2019);

SELECT * FROM RequestList;

SELECT
   R.RecordID,
   R.YearVal,
   ISNULL(Y.NetPresentValue, 0) AS NetPresentValue
FROM RequestList R
LEFT JOIN YearData Y
   ON R.RecordID = Y.RecordID
  AND R.YearVal = Y.YearVal;
```

4. Output :

[ MEDIUM ]

| | EmployeeID | EmployeeName | Dept | ManagerID |
|---|---|---|---|---|
| 1 | 1 | Sarthak | HR | NULL |
| 2 | 2 | Reyansh | Finance | 1 |
| 3 | 3 | Yogesh | IT | 1 |
| 4 | 4 | Pravi | Finance | 2 |
| 5 | 5 | Surya | IT | 3 |
| 6 | 6 | MD | HR | 1 |

| | Employee_Name | Manager_Name | Employee_Dept | Manager_Dept |
|---|---|---|---|---|
| 1 | Sarthak | NULL | HR | NULL |
| 2 | Reyansh | Sarthak | Finance | HR |
| 3 | Yogesh | Sarthak | IT | HR |
| 4 | Pravi | Reyansh | Finance | Finance |
| 5 | Surya | Yogesh | IT | IT |
| 6 | MD | Sarthak | HR | HR |

[ HARD ]

| | RecordID | YearVal | NetPresentValue |
|---|---|---|---|
| 1 | 1 | 2018 | 100 |
| 2 | 7 | 2020 | 30 |
| 3 | 13 | 2019 | 40 |
| 4 | 1 | 2019 | 113 |
| 5 | 2 | 2008 | 121 |
| 6 | 3 | 2009 | 12 |
| 7 | 11 | 2020 | 99 |
| 8 | 7 | 2019 | 0 |

| | RecordID | YearVal |
|---|---|---|
| 1 | 1 | 2019 |
| 2 | 2 | 2008 |
| 3 | 3 | 2009 |
| 4 | 7 | 2018 |
| 5 | 7 | 2019 |
| 6 | 7 | 2020 |
| 7 | 13 | 2019 |

| | RecordID | YearVal | NetPresentValue |
|---|---|---|---|
| 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.