Experiment 2

Name: Mishika | UID: 23BCS13811
Branch: CSE | Section/Group: 23BCS_KRG-2/A
Semester: 5 | Date of Performance:31/07/25
Subject Name: Advanced Database and Management System | Subject Code: 23CSP-333

1. **Aim:**

   [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 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
   - The 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:**

```sql
-- HEADING: ORGANIZATIONAL HIERARCHY EXflLORER (MEDIUM)

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

```sql
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 A.EmpName AS [Employee Name],
A.Department[Department], B.EmpName[Manager Name],
B.Department [Manager Department]
FROM Employee AS A LEFT OUTER JOIN Employee AS B
ON
A.ManagerID = B.EmpID;



-- HEADING: FINANCIAL FORECAST MATCHING WITH FALLBACK
STRATEGY (HARD)
-- TBL1 = YEAR_TBL: ID, YEAR, NflV
-- TBL2 = QUERY TBL: ID, YEAR
-- IF ANY NflV VALUE IS MISSING IN YEAR TBL MAKE IT O.

-- FIND THE NflV OF EACH QUERY

CREATE TABLE TBL_YEAR(
    ID INT,
    YEAR INT,
    NflV INT
);

CREATE TABLE TBL_QUERY(
    ID INT,
    YEAR INT
);
```

```sql
INSERT INTO TBL_YEAR
VALUES
(1, 2018, 100),
(7, 2020, 30),
(13, 2019, 40),
(1, 2019, 113),
(2, 2008, 121),
(3, 2009, 12),
(11, 2020, 99),
(7, 2019, 9);


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


SELECT B.ID, B.YEAR, ISNULL(A.NflV, 0) AS [NflV]
FROM TBL_YEAR AS A RIGHT JOIN TBL_QUERY AS B
ON A.ID = B.ID AND A.YEAR = B.YEAR;
```

4. **Output:**

[MEDIUM]

	Employee Name	Department	Manager Name	Manager Department
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	NPV
1	1	2019	113
2	2	2008	121
3	3	2009	12
4	7	2018	0
5	7	2019	9
6	7	2020	30
7	13	2019	40

5. Learning Outcomes:

- Understand and implement self-joins to model hierarchy relationships within a single table (e.g, employees reporting to other employees).
- Construct relational queries to fetch meaningful information such as employee-manager 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.