Experiment 2

Student Name: Rachit Kumar
UID: 23BCS11597
Branch: BE CSE
Section: 23BCS_KRG-2/A
Subject Name: Advanced Database. and Management System
Subject Code: 23CSP-333
Date :07/08/2025
Semester: 5th

1. **Aim:**

Q1: 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.

Q2: You are a Data Engineer at FinSight Corp, a company that models Net Present Value (NPV) projections for
investment decisions. Your system maintains two key datasets:
1. Year_tbl: Actual recorded NPV's of various financial instruments over different years:
ID: Unique Financial instrument identifier.
YEAR: Year of record
NPV: Net Present Value in that year
2. Queries_tbl: A list of instrument-year pairs for which stakeholders are requesting NPV values:
ID: Financial instrument identifier
YEAR: Year of interest.
Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted
form.
However, not all ID-YEAR combinations in the Queries table are present in the Year_tbl. If an NPV is missing
for a requested combination, assume it to be 0 to maintain a consistent financial report.

2. Tools Used: SQL Server (One compiler)

3. Code:

Q1:

```
CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    Ename VARCHAR(50),
    Department VARCHAR(50),
    ManagerID INT NULL
);

INSERT INTO Employee (EmpID, Ename, 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.Ename AS EmployeeName,
    E.Department AS EmployeeDept,
    M.Ename AS ManagerName,
    M.Department AS ManagerDept
FROM Employee E
LEFT JOIN Employee M
    ON E.ManagerID = M.EmpID;
```

Q2:

```
CREATE TABLE Year_tbl (
    ID INT,
    YEAR INT,
    NPV INT
);

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

SELECT
    Q.ID,
    Q.YEAR,
    ISNULL(Y.NPV, 0) AS NPV
FROM Queries_tbl Q
LEFT JOIN Year_tbl Y
    ON Q.ID = Y.ID AND Q.YEAR = Y.YEAR
ORDER BY Q.ID, Q.YEAR;
```

Output:

Output:

EmployeeName	EmployeeDept	ManagerName
Alice	HR	NULL
Bob	Finance	Alice
Charlie	IT	Alice
David	Finance	Bob
Eve	IT	Charlie
Frank	HR	Alice

Output:

ID	YEAR	NPV
1	2019	113
2	2008	121
3	2009	12
7	2018	0
7	2019	0
7	2020	30
13	2019	40

4. Learning Outcomes

- Design relational tables with self-referencing **foreign keys** (e.g., Employee → Manager).
- Understand **self-joins** to map hierarchical relationships (employee–manager).
- Apply **INNER JOIN** and **LEFT JOIN** operations to retrieve related data across tables.

• Implement **fallback strategies** using joins and `ISNULL` / `COALESCE` for missing values (e.g., NPV defaults to 0).

• Write queries to **combine and align financial data** across Year and Queries tables.

• Develop skills in **ordering and grouping query outputs** for clear reporting.