Experiment 2

Student Name: Hemant Narain Jha   **UID: 23BCS10022**
Branch: B.E.CSE   **Section/Group: 23BCS-KRG-2B**
Semester: 5th   **Date of Performance:28-07-25**
Subject Name: ADBMS   **Subject Code:23CSP-333**

1. Aim:

a) 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.

b) 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:

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

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. Objective:

- To understand how to use JOINS in SQL.

- To understand the basic SQL Queries.

- To generate hierarchical reports from self-referencing tables.

3. DBMS script and output:

a)

```
CREATE TABLE Employee (
    Employee_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Department VARCHAR(100),
    Manager_ID INT,
    FOREIGN KEY (Manager_ID) REFERENCES Employee(Employee_ID) ON
    DELETE SET NULL
);
INSERT INTO Employee (Employee_ID, Name, Department, Manager_ID)
    VALUES
(1, 'Amit Kumar', 'HR', NULL),
(2, 'Priya Sharma', 'Finance', 1),
(3, 'Ravi Patel', 'IT', 1),
(4, 'Neha Gupta', 'Marketing', 2),
(5, 'Vikas Reddy', 'IT', 3);
SELECT
    e.Name AS Employee_Name,
    e.Department AS Employee_Department,
    m.Name AS Manager_Name,
    m.Department AS Manager_Department
```

```
FROM
    Employee e
LEFT JOIN
    Employee m ON e.Manager_ID = m.Employee_ID
ORDER BY
    e.Employee_ID;
```

Output:

```
+----------------+----------------------+----------------+----------------------+
| Employee_Name  | Employee_Department  | Manager_Name   | Manager_Department   |
+----------------+----------------------+----------------+----------------------+
| Amit Kumar     | HR                   | NULL           | NULL                 |
| Priya Sharma   | Finance              | Amit Kumar     | HR                   |
| Ravi Patel     | IT                   | Amit Kumar     | HR                   |
| Neha Gupta     | Marketing            | Priya Sharma   | Finance              |
| Vikas Reddy    | IT                   | Ravi Patel     | IT                   |
+----------------+----------------------+----------------+----------------------+
```

b)

```
CREATE TABLE Year_tbl (
    ID INT,
    YEAR INT,
    NPV DECIMAL(15, 2),
    PRIMARY KEY (ID, YEAR)
);
INSERT INTO Year_tbl (ID, YEAR, NPV) VALUES
(101, 2020, 5000.00),
(102, 2021, 7000.00),
(103, 2020, 4500.00);
CREATE TABLE Queries_tbl (
```

```sql
    ID INT,
    YEAR INT,
    PRIMARY KEY (ID, YEAR)
);
INSERT INTO Queries_tbl (ID, YEAR) VALUES
(101, 2021),
(102, 2021),
(103, 2020),
(101, 2020),
(104, 2020);
SELECT
    q.ID,
    q.YEAR,
    COALESCE(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:

```
ID              YEAR            NPV
----------      ----------      ------------------
       101      2020                       5000.00
       101      2021                           .00
       102      2021                       7000.00
       103      2020                       4500.00
       104      2020                           .00
```

4. Learning outcomes:

- You will be able to write basic SQL queries.

- You will learn to perform JOINS in SQL.

- You will understand how to implement foreign keys.