Experiment 2

Student Name: Ishan **UID: 23BCS12656**
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:

```
-- Create the Employee table
CREATE TABLE Employee (
    EmpID INT,
    Ename VARCHAR(100),
    Department VARCHAR(100),
    ManagerID INT
);


-- Insert data into the Employee table with updated names
INSERT INTO Employee (EmpID, Ename, Department, ManagerID) VALUES
    (1, 'Ishan', 'Admin', NULL),
    (2, 'Priyanshu', 'Accounts', 1),
    (3, 'Rohit', 'Tech', 1),
    (4, 'Kaif', 'Accounts', 2),
    (5, 'Lakshay', 'Tech', 3),
    (6, 'Ram', 'Admin', 1);


-- Retrieve employee and their manager details
SELECT
    E1.Ename AS "Employee Name",
    E2.Ename AS "Manager Name",
    E1.Department AS "Employee Department",
    E2.Department AS "Manager Department"
FROM
    Employee AS E1
```

```
LEFT JOIN
  Employee AS E2
  ON E1.ManagerID = E2.EmpID;
```

Output:

Output:

```
+---------------+--------------+----------------------+--------------------+
| Employee Name | Manager Name | Employee Department | Manager Department |
+---------------+--------------+----------------------+--------------------+
| Ishan         | NULL         | Admin                | NULL               |
| Priyanshu     | Ishan        | Accounts             | Admin              |
| Rohit         | Ishan        | Tech                 | Admin              |
| Kaif          | Priyanshu    | Accounts             | Accounts           |
| Lakshay       | Rohit        | Tech                 | Tech               |
| Ram           | Ishan        | Admin                | Admin              |
+---------------+--------------+----------------------+--------------------+
```

B)

```
-- Create table to store yearly NPV data
CREATE TABLE Year_tbl (
    ID INT,
    YEAR INT,
    NPV INT
);

-- Create table to store query data
CREATE TABLE Queries (
    ID INT,
    YEAR INT
);

-- Insert values into Year_tbl
INSERT INTO Year_tbl (ID, YEAR, NPV) VALUES
    (2, 2017, 95),
    (4, 2020, 55),
    (6, 2018, 120),
    (2, 2018, 88),
    (3, 2016, 110),
    (5, 2015, 15),
    (4, 2019, 42),
    (6, 2017, 60);
```

```
-- Insert values into Queries
INSERT INTO Queries (ID, YEAR) VALUES
    (2, 2018),
    (3, 2016),
    (4, 2020),
    (4, 2019),
    (6, 2017),
    (6, 2019),   -- No match in Year_tbl
    (1, 2021);   -- No match in Year_tbl

-- Retrieve NPV for each query; if no match, return 0
SELECT
    Q.ID,
    Q.YEAR,
    IFNULL(Y.NPV, 0) AS "NPV"
FROM
    Queries AS Q
LEFT JOIN
    Year_tbl AS Y
    ON Q.ID = Y.ID AND Q.YEAR = Y.YEAR;
```

Output:

Output:

ID	YEAR	NPV
2	2018	88
3	2016	110
4	2020	55
4	2019	42
6	2017	60
6	2019	0
1	2021	0

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.