DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2

Student Name: Raghav Yadav
Branch: CSE
Semester: 5
Subject Name: Advanced Database and Management System

UID: 23BCS11935
Section/Group: 23BCS_KRG-2/A
Date of Performance:31/07/25
Subject Code: 23CSP-333

1. Aim:

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:

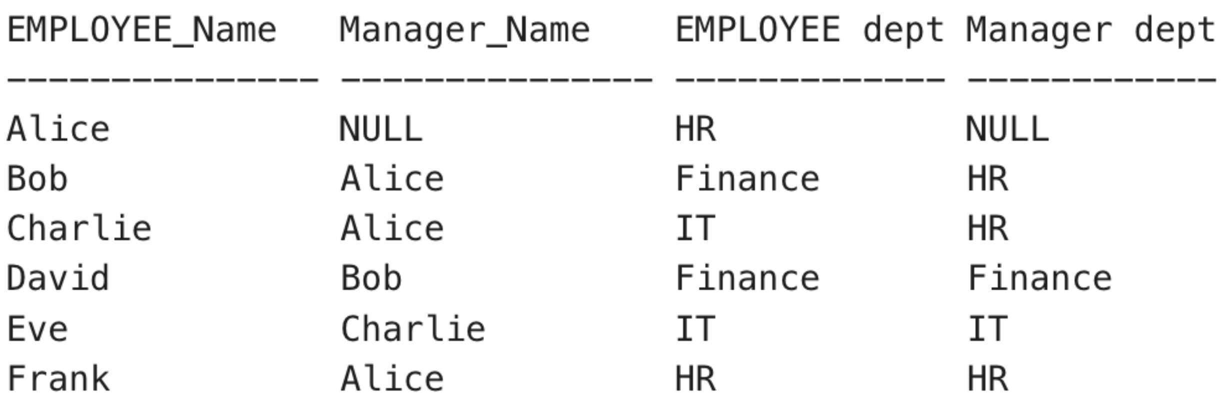
- The employee's name and department
- The manager's name and department (if applicable)
- This will help the HR department visualize the internal reporting hierarchy.

2. Tools Used: SQL Server Management Studio

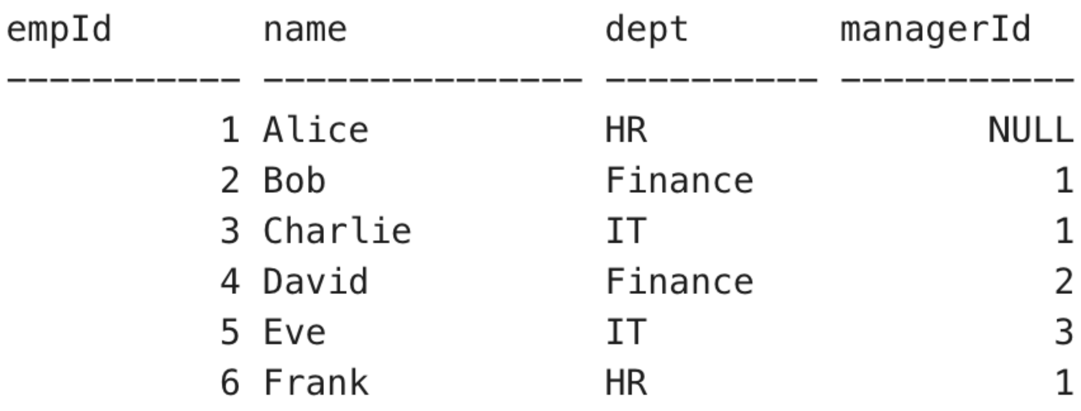
3. Code:

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


-- create
CREATE TABLE EMPLOYEE (
  empId int  primary KEY,
  name varchar(15),
  dept varchar(10),
  managerId int

);


INSERT INTO EMPLOYEE(empId,name,dept,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);

-- fetch
SELECT * FROM EMPLOYEE;

ALTER TABLE EMPLOYEE
ADD constraint FK_EMPLoYEE FOREIGN KEY(managerId)
references EMPLOYEE(empId)

select E1.name as[EMPLOYEE_Name], E2.name
as[Manager_Name],E1.dept as[EMPLOYEE dept], E2.dept
as[Manager dept]
from EMPLOYEE as E1
Left Outer join
EMPLOYEE as E2
on E1.managerId=E2.empId
```

4. **Output:**



```
EMPLOYEE_Name   Manager_Name    EMPLOYEE dept Manager dept
--------------- --------------- ------------- ------------
Alice           NULL            HR            NULL
Bob             Alice           Finance       HR
Charlie         Alice           IT            HR
David           Bob             Finance       Finance
Eve             Charlie         IT            IT
Frank           Alice           HR            HR
```



```
empId       name            dept       managerId
----------- --------------- ---------- -----------
          1 Alice           HR                NULL
          2 Bob             Finance              1
          3 Charlie         IT                   1
          4 David           Finance              2
          5 Eve             IT                   3
          6 Frank           HR                   1
```

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.