

**Experiment 1.2**

**Student Name: Avin Mehla**  **UID:** 23BAI70080

**Branch:** BE-AIT-CSE **Section/Group:** 23AML\_KRG-1A

**Semester:** 5th **Date of Performance:** 30 July 2025

**Subject Name:** ADBMS **Subject Code:** 23CSP-333

# MEDIUM - LEVEL

1. **Problem Title:** Organizational Hierarchy Explorer
2. **Problem Description:** 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:

* 1. The employee’s name and department
  2. Their manager’s name and department (if applicable)
  3. This will help the HR department visualize the internal reporting hierarchy.

1. **SQL Commands:** 
   1. Create the database and use it:

|  |  |  |  |
| --- | --- | --- | --- |
|  | create database AIT\_1A; | |  |
| use AIT\_1A; |  |

* 1. Create tables Employee and adding Foreign key:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | CREATE TABLE Employee ( | |  | | | | | |
| EmpID INT PRIMARY KEY, | | | |  | | | |
| EmpName VARCHAR(50) NOT NULL, | | | | |  | | |
| Department VARCHAR(50) NOT NULL, | | | | | |  | |
| ManagerID INT NULL); | | |  | | |
|  | ALTER TABLE Employee ADD CONSTRAINT FK\_EMPLOYEE | | | | | | | |  |
| FOREIGN KEY (ManagerID) | |  | | | | | |
| references Employee(EmpID); | | |  | | | | |

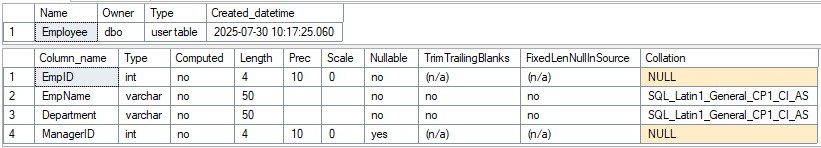
* 1. Insert the values in the tables:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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); | | |  | | |

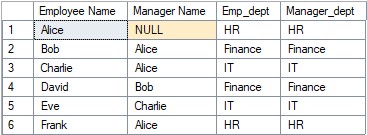
* 1. Selecting the Employee with their respective managers:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | SELECT E1.EmpName [Employee Name], E2.EmpName [Manager Name], | | | | |  |
| E1.Department [Emp\_dept], | | |  | |
| E1.Department [Manager\_dept] | | | |  |
| from Employee as E1 | |  | |
|  | left outer join Employee as E2 on E1.ManagerID = E2.EmpID; | | | |

1. **Output:**



*Figure 1 Employee Table Description*



*Figure 2 Output of the Select Query*

1. **Learning Outcome:**

* 1. I learnt how to link and add constraints like primary key after the table creation.
  2. I learnt about different types of joints.
  3. I learnt how to use LEFT OUTER JOIN to retrieve combined data from related tables.

# HARD - LEVEL

1. **Problem Title:** Financial Forecast Matching with Fallback Strategy
2. **Problem Description:**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.

1. **SQL Commands:** 
   1. Create the tables.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CREATE TABLE Year\_tbl ( | | | | | |  | |
| ID INT, | |  | | | |
| YEAR INT, | | | |  | |
| NPV INT | |  | |
| ); |  |
|  |
| -- Create Queries table (requested values) | | | | | | |  |
| CREATE TABLE Queries ( | | | | |  | |
| ID INT, | |  | | |
| YEAR INT | | |  | |
| ); |  | |

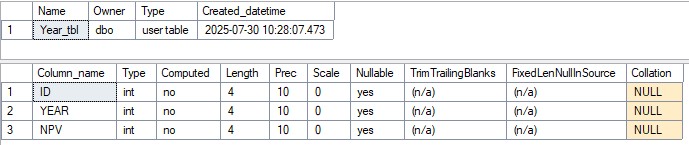
* 1. Insert the values.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | INSERT INTO Year\_tbl (ID, YEAR, NPV) VALUES | | | | | | |  |
| (1, 2018, 100), | | | | |  | |
| (7, 2020, 30), | | | |  |
| (13, 2019, 40),  (1, 2019, 113),  (2, 2008, 121), | | | | |
| (3, 2009, 12), | | | | , |
| ) | | | |
|  | | |  |
|  | INSERT INTO Queries (ID, YEAR) VALUES | | | | | | |  |
| (1, 2019),  (2, 2008),  (3, 2009),  (7, 2018),  (7, 2019), | |  | | | | |
|  | (7, 2020), | |  | | | | | | |
| (13, 2019); | | |  | | | | | |

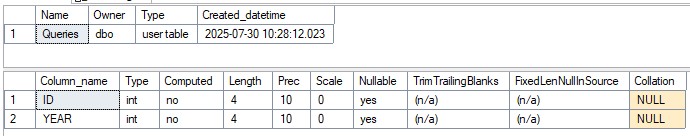
* 1. Use a subquery to count the number of courses under each department.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | select q.id, q.year, Isnull(y.NPV, 0) [NPV] | | |  | |
| from Queries as q | |  |
| left outer join Year\_tbl as y on q.id = y.id and q.YEAR = y.YEAR | | | |  |
|  | order by q.id; |  | |

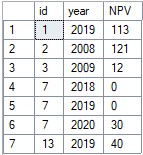
1. **Output:**



*Figure 1 Year\_tbl description*



*Figure 2 Queries table description*



*Figure 3 Select Query*

1. **Learning Outcomes:**

* 1. I learned how to perform left join and understand the table.
  2. I learned some of the build functions of the Microsoft SQL server.
  3. I learned about aliases in the SQL queries.