



# University Institute of Engineering

## Department of Computer Science & Engineering

### EXPERIMENT : 2

**NAME : GURVEER SINGH MANGAT**

**UID: 23BCS11074**

**BRANCH : BE-CSE**

**SECTION/GROUP : KRG\_1A**

**SEMESTER : 5<sup>TH</sup>**

**SUBJECT CODE : 23CSP-333**

**SUBJECT NAME : ADBMS**

### 1. Aim Of The Practical :

#### **Ques 1:** Organizational Hierarchy Explorer [ MEDIUM ]

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.

EmpID	Ename	Department	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

## Ques 2: Financial Forecast Matching with Fallback Strategy [ HARD ]

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.

ID	YEAR	NPV	ID	YEAR
1	2018	100	1	2019
7	2020	30	2	2008
13	2019	40	3	2009
1	2019	113	7	2018
2	2008	121	7	2019
3	2009	12	7	2020
11	2020	99	13	2019
7	2019	0		

## 2. Tools Used : SQL Server Management Studio

## 3. Code :

```
--Medium Question
CREATE TABLE Employee(
Emp_ID INT identity(1,1) PRIMARY KEY,
Ename varchar(30) NOT NULL,
Department varchar(20),
ManagerID INT,
foreign key (ManagerID) references Employee(Emp_ID)
);
insert into employee(Ename,Department,ManagerID) values
('Alice','HR',null),
('Bob','Finance',1),
('Charlie','IT',1),
('David','Finance',2),
('Eve','IT',3),
('Frank','HR',1)

select E1.EName as [Employee_Name], E2.EName as [Manager_Name],
E1.Department as [Emp_Department], E2.Department as [Manager_Dept]
from Employee as E1
```

```

left outer join Employee as E2
on E1.ManagerID=E2.Emp_ID; --always put foreign key column on the left
side always works

```

--Hard Question

```

create table Year_tbl(
ID INT not null,
Year INT,
NPV INT
);
create table Queries_tbl(
ID int NOT NULL,
Year INT
);
insert into Year_tbl(ID,Year,NPV) values
(1,2018,100),
(7,2020,30),
(13,2019,40),
(1,2019,113),
(2,2008,121),
(3,2009,112),
(11,2020,99),
(7,2019,0)

```

```

insert into Queries_tbl(ID,Year) values
(1,2019),
(2,2008),
(3,2009),
(7,2018),
(7,2019),
(7,2020),
(13,2019)

```

```

select Y.ID as [ID],Y.Year as [Year],isnull(Q.npv,0)
from Queries_tbl as Y
left outer join
Year_tbl as Q
on Y.ID=Q.ID
and
Y.Year =Q.Year;

```

#### 4. Output :

[ MEDIUM ]

	Employee_Name	Manager_Name	Emp_Department	Manager_Dept
1	Alice	NULL	HR	NULL
2	Bob	Alice	Finance	HR
3	Charlie	Alice	IT	HR
4	David	Bob	Finance	Finance
5	Eve	Charlie	IT	IT
6	Frank	Alice	HR	HR

[ HARD ]

	ID	Year	(No column name)
1	1	2019	113
2	2	2008	121
3	3	2009	112
4	7	2018	0
5	7	2019	0
6	7	2020	30
7	13	2019	40

## 5. Learning Outcomes :

- Learn how to use self joins to represent hierarchical structures within a single table, such as employees and their managers.
- Write relational queries to retrieve manager–employee relationships, making use of left join to include entries with null values.
- Create and fill tables with the create table and insert into commands for real-world scenarios like hierarchical data and time-based datasets.
- Perform joins across multiple tables to compare and align information from different sources, such as actual versus planned figures like NPV for specific years.
- Manage missing values in join results by using functions like isnull to replace them with defaults.
- Implement conditional joins that match on more than one column such as matching by both id and year for precise data linking.
- Build SQL-based problem-solving techniques to extract valuable insights from HR systems and financial data in enterprise environments