

**University Institute of Engineering**  
**Department of Computer Science & Engineering**

**EXPERIMENT: 2**

**NAME** : Himanshu Dubey                      **UID** : 23BCS12967  
**SECTION** : KRG\_2A                              **SEMESTER:** 5<sup>TH</sup>  
**SUBJECT CODE:** 23CSP-339                      **SUBJECT** : ADBMS

**I. Aim Of The Practical :**

**[MEDIUM] Organizational Hierarchy Explorer**

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.

**[HARD] Financial Forecast Matching with Fallback Strategy**

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:

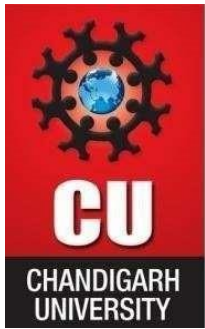
1. Year\_tbl: Actual recorded NPV's of various financial instruments over different years:

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:



# University Institute of Engineering

## Department of Computer Science & Engineering

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.

II. Tools Used: SQL Server Management Studio

III. Code:

```
-----MEDIUM: Organizational Hierarchy Explorer create
database db2
create table emp_tbl(empid int primary key, emp_name varchar(max),
department char(10), manager_id int) insert into emp_tbl values
(1, 'Alice', 'HR', NULL),
(2, 'Bob', 'Finance', 1),
(3, 'Chotu', 'I.T.', 1),
(4, 'Dhokad', 'Finance', 2),
(5, 'Ennu', 'IT', 3),
(6, 'Fulli', 'HR', 1);

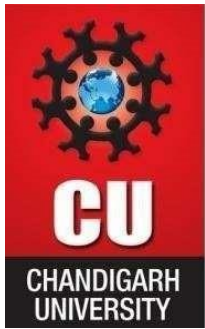
alter table emp_tbl
add constraint fk_emp foreign key(manager_id)
references emp_tbl(empid)

select * from emp_tbl

select e.emp_name as employee_name, e.department as employee_dept,
f.emp_name as manager_name, f.department as manager_dept
from emp_tbl as e left outer join emp_tbl as f on
e.manager_id=f.empid

-----HARD: Financial Forecast Matching with Fallback

create table year_tbl(id int , year int, npv int)
insert into year_tbl(id , year, npv) values
```



## University Institute of Engineering

### Department of Computer Science & Engineering

```
(1,2018,100),  
(7,2020,30),  
(13,2019,40),  
(1,2019,113),  
(2,2008,121),  
(3,2009,12),  
(11,2020,99),  
(7,2019,0);
```

```
create table quer_tbl(qid int , year int)  
insert into quer_tbl values  
(1,2019),  
(2,2008),  
(3,2009),  
(7,2018),  
(7,2019),  
(7,2020),  
(13,2019);
```

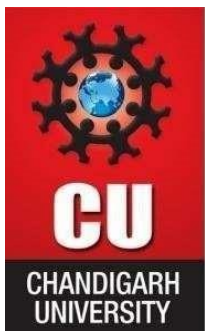
```
select * from year_tbl select  
* from quer_tbl
```

```
select y.id, y.year, isnull(y.npv,0)  
from year_tbl as y inner join  
quer_tbl as q on  
y.id=q.qid and  
y.year=q.year order  
by y.id
```

Output :

	empid	emp_name	department	manager_id
1	1	Alice	HR	NULL
2	2	Bob	Finance	1
3	3	Chotu	I.T.	1
4	4	Dhokad	Finance	2
5	5	Ennu	IT	3
6	6	Fulli	HR	1

[ MEDIUM ]



# University Institute of Engineering

## Department of Computer Science & Engineering

Tables: .....Employee table Final:

	employee_name	employee_dept	manager_name	manager_dept
1	Alice	HR	NULL	NULL
2	Bob	Finance	Alice	HR
3	Chotu	I.T.	Alice	HR
4	Dhokad	Finance	Bob	Finance
5	Ennu	IT	Chotu	I.T.
6	Fulli	HR	Alice	HR

[ HARD ]

	id	year	npv
1	1	2018	100
2	7	2020	30
3	1...	2019	40
4	1	2019	113
5	2	2008	121
6	3	2009	12
7	1...	2020	99
8	7	2019	0

Tables:

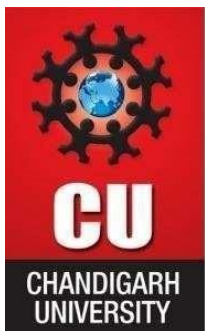
	qid	year
1	1	2019
2	2	2008
3	3	2009
4	7	2018
5	7	2019
6	7	2020
7	13	2019

.....Year's table

	id	year	(No column name)
1	1	2019	113
2	2	2008	121
3	3	2009	12
4	7	2019	0
5	7	2020	30
6	1...	2019	40

.....Queries table

Final: .....NPV table



**University Institute of Engineering**  
**Department of Computer Science & Engineering**

**IV. Learning Outcomes :**

- Understand and implement self-joins to model hierarchical relationships within a single table (e.g., employees reporting to other employees).
- Construct relational queries to fetch meaningful information such as employeemanager 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