DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Experiment 2

Student Name: Sukhjinder Singh

UID: 23BAI70078

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Date of Performance: 29th July, 2025

Subject Name: ADBMS

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 also an employee in the same table).

2. Tools Used:

SQL Server Management Studio 21 (SSMS) code editor.

3. Experiment:

Task is to generate a report that **maps employees to their respective managers**, showing:

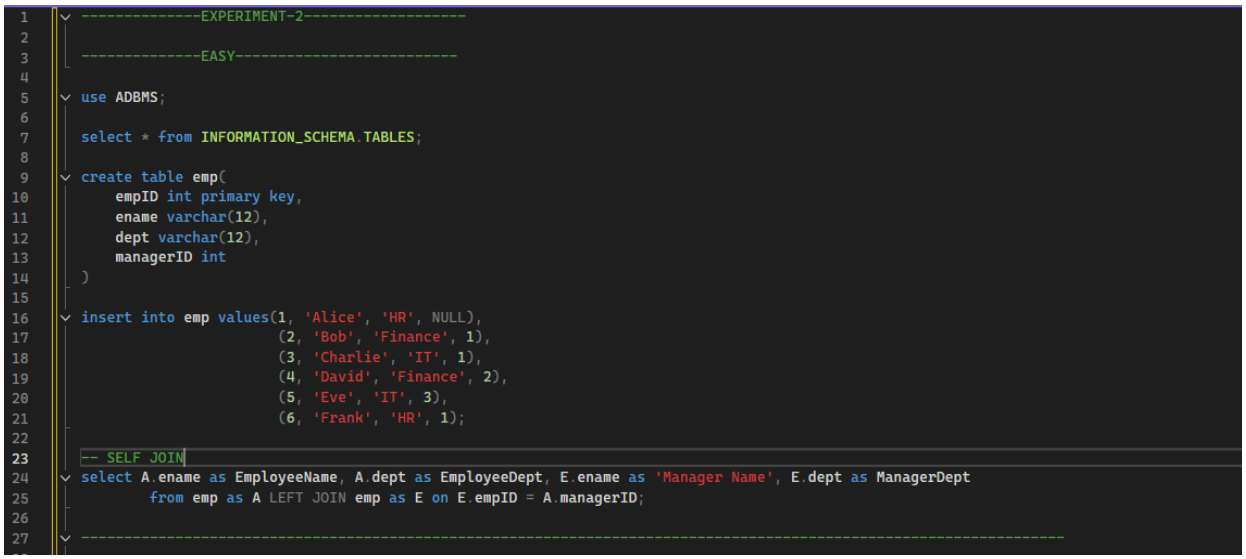
The employee's name and department.

Their manager's name and department (if applicable).

This will help the HR department visualize the internal reporting hierarchy.

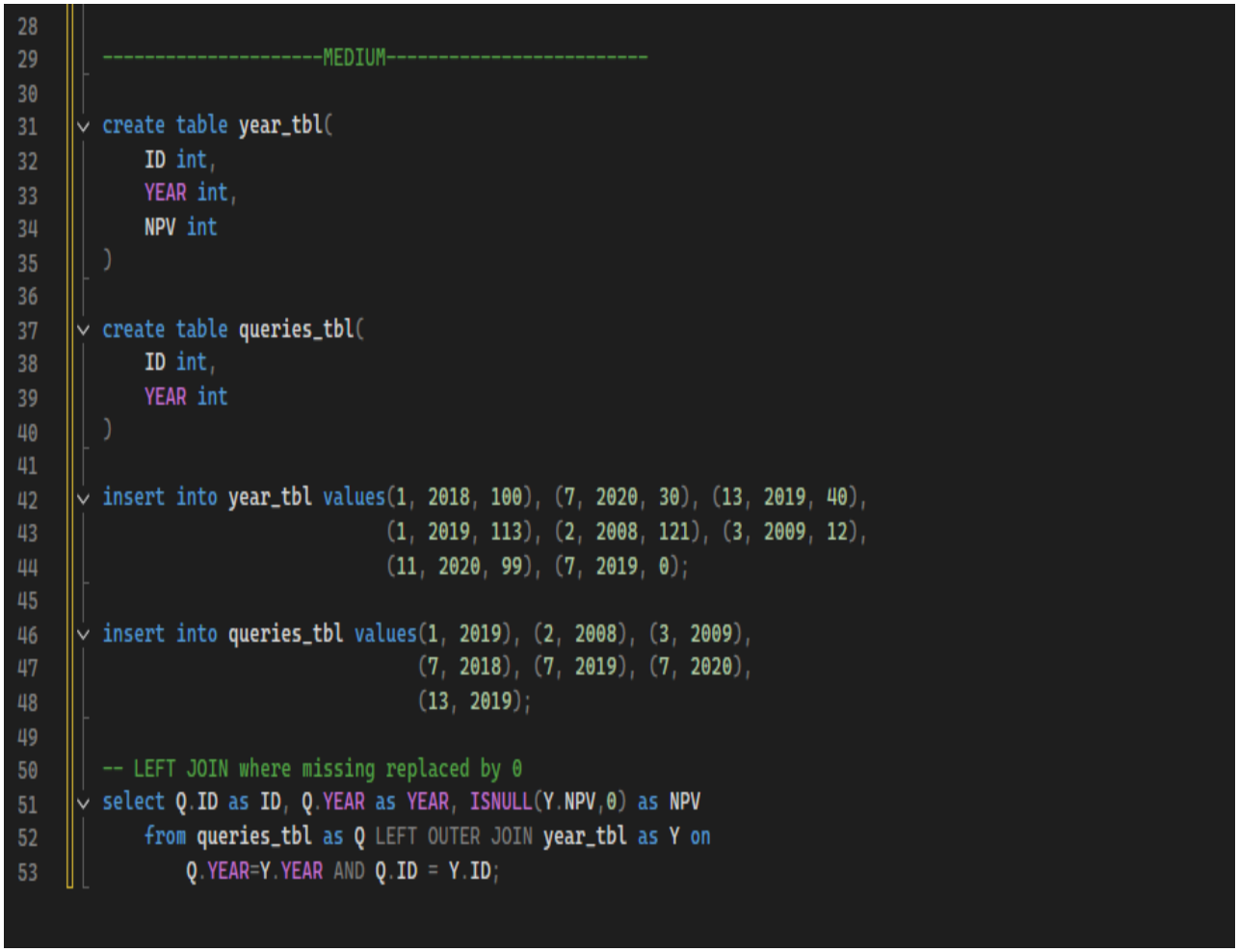
4. Solution:

Easy-Level



```
--------------EXPERIMENT-2------------------

--------------EASY--------------------------

use ADBMS;

select * from INFORMATION_SCHEMA.TABLES;

create table emp(
    empID int primary key,
    ename varchar(12),
    dept varchar(12),
    managerID int
)

insert into emp 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);

-- SELF JOIN
select A.ename as EmployeeName, A.dept as EmployeeDept, E.ename as 'Manager Name', E.dept as ManagerDept
       from emp as A LEFT JOIN emp as E on E.empID = A.managerID;

-----------------------------------------------------------------------------------------------------------
```

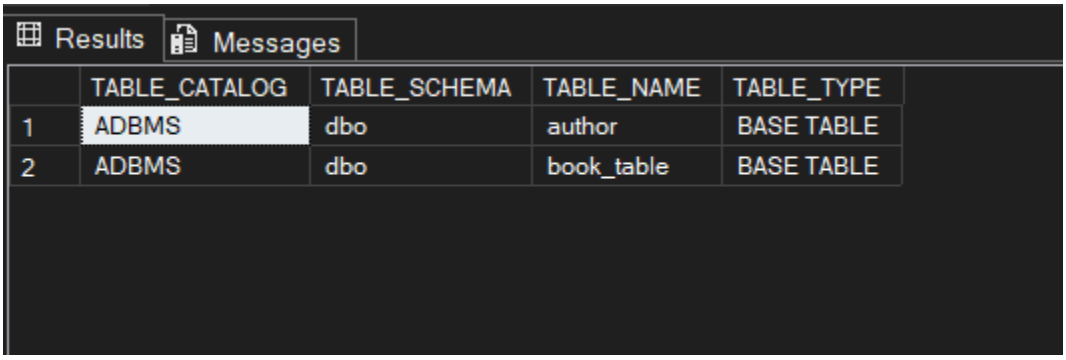
Medium-Level



```
--------------------MEDIUM-----------------------

create table year_tbl(
    ID int,
    YEAR int,
    NPV int
)

create table queries_tbl(
    ID int,
    YEAR int
)

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

insert into queries_tbl values(1, 2019), (2, 2008), (3, 2009),
                              (7, 2018), (7, 2019), (7, 2020),
                              (13, 2019);

-- LEFT JOIN where missing replaced by 0
select Q.ID as ID, Q.YEAR as YEAR, ISNULL(Y.NPV,0) as NPV
    from queries_tbl as Q LEFT OUTER JOIN year_tbl as Y on
        Q.YEAR=Y.YEAR AND Q.ID = Y.ID;
```

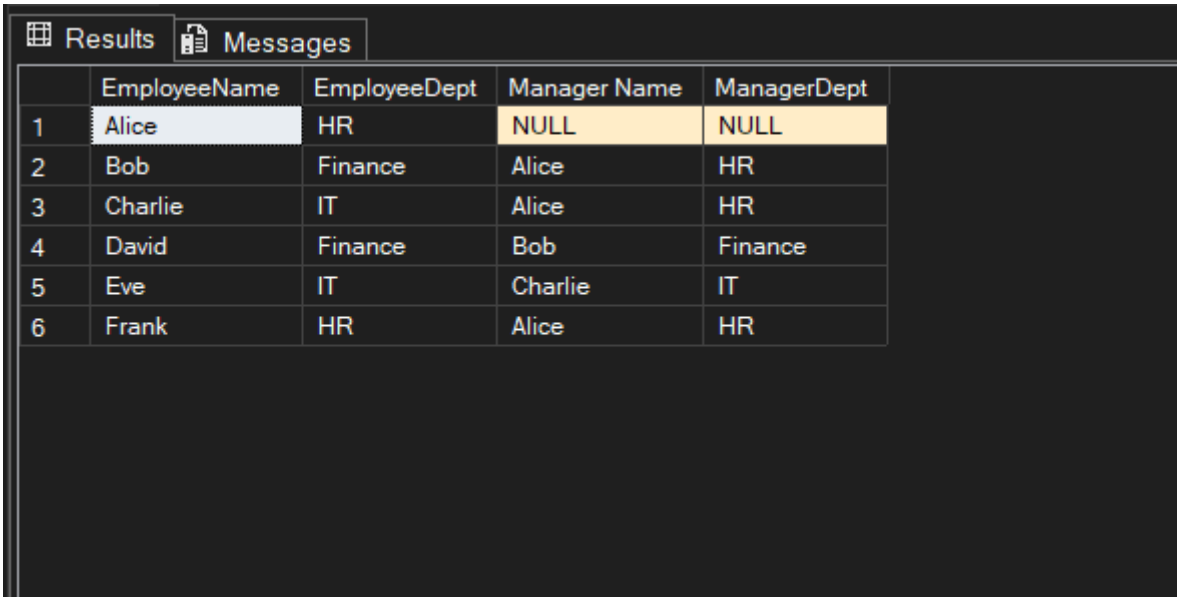
5. **Output:**

Easy-Level



Results | Messages

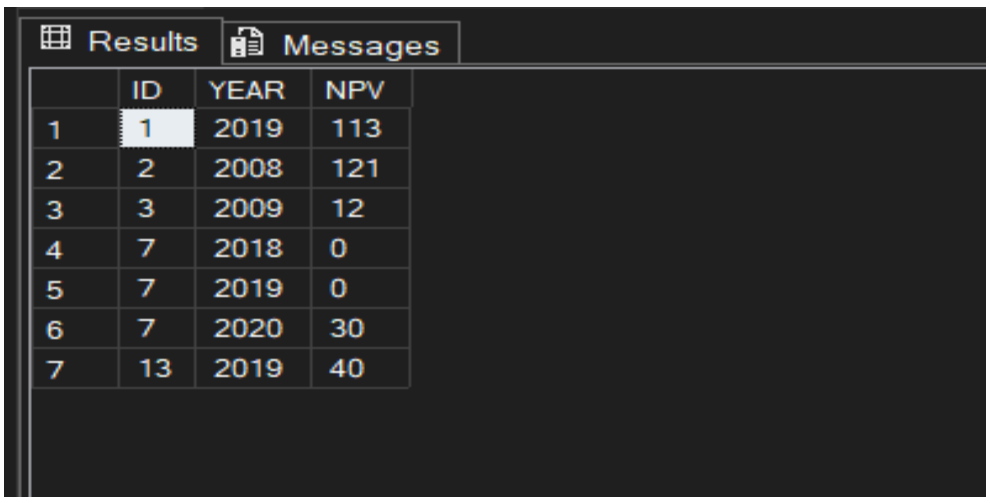
	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	ADBMS	dbo	author	BASE TABLE
2	ADBMS	dbo	book_table	BASE TABLE



Results | Messages

	EmployeeName	EmployeeDept	Manager Name	ManagerDept
1	Alice	HR	NULL	NULL
2	Bob	Finance	Alice	HR
3	Charlie	IT	Alice	HR
4	David	Finance	Bob	Finance
5	Eve	IT	Charlie	IT
6	Frank	HR	Alice	HR

Medium-Level



Results | Messages

	ID	YEAR	NPV
1	1	2019	113
2	2	2008	121
3	3	2009	12
4	7	2018	0
5	7	2019	0
6	7	2020	30
7	13	2019	40

6. **Learning Outcomes:**
    - Understood the concept of joins.
    - Learn’t about various types of joins such as LEFT and SELF join.
    - Learn’t how to apply the joins and add various constraints to them as per the user.
    - Learn’t how to replace the NULL value with desired value.