EXPERIMENT - 3

Name: Neha Sharma	**UID**: 23BCS10766
Branch: CSE	**Section**: KRG1-B
Semester: 5	**Date of Performance**: 19/08/2025
Subject: ADBMS	**Subject Code**: 23CSP-333

Question 1 : Medium-Level Problem

Problem Title: **Department Salary Champions**

Scenario:

In a bustling corporate organization, each department strives to retain the most talented (and well-compensated) employees. You have access to two key records: one lists every employee along with their salary and department, while the other details the names of each department. Your task is to identify the top earners in every department.

If multiple employees share the same highest salary within a department, all of them should be celebrated equally. The final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

Solution :

Program Code:

```
CREATE TABLE depart (
    id INT PRIMARY KEY,
    dept_name VARCHAR(50)
);


CREATE TABLE empl (
    id INT,
```

```sql
 name VARCHAR(50),
    salary INT,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES depart(id)
);
INSERT INTO depart (id, dept_name) VALUES
(1, 'IT'),
(2, 'SALES');


 INSERT INTO empl(id, name, salary, department_id) VALUES
(1, 'JOE', 70000, 1),
(2, 'JIM', 90000, 1),
(3, 'HENRY', 80000, 2),
(4,'Sam',60000,2),
(4,'MAX',90000,1);


select d.dept_name,e.name,e.salary
from empl e
left join
depart d
on d.id = e.department_id
where e.salary in(
select max(t.salary)
from empl t
group by t.department_id
)
order by dept_name;
```

Output:

	dept_name	name	salary
1	IT	JIM	90000
2	IT	MAX	90000
3	SALES	HENRY	80000

Question 2 : Hard - Level Problem

Problem Title : **NPV Lookup with Missing Data Handling**

Scenario:

Two legacy HR systems (A and B) have separate records of employee salaries. These records may overlap.   Management wants to merge these datasets and identify each unique employee (by EmpID) along with their lowest recorded salary across both systems.

Objective

1. Combine two tables A and B.

 2. Return each EmpID with their lowest salary, and the corresponding Ename.

Solution:

Program Code:

```
create table A(
emp_id int,
ename varchar(30),
salary int
);
```

```
create table B(
emp_id int,
ename varchar(30),
salary int
);

insert into A(emp_id,ename,salary)
values(1,'AA',1000),(2,'BB',300);

insert into B(emp_id,ename,salary)
values(1,'BB',400),(2,'CC',100);

select emp_id,min(ename),min(salary)
from (select *
from A
union all
select *
from B) as ta
group by emp_id;
```

Output:

	EmpID	Ename	Min_Salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

Learning Outcomes

- Learnt how to join or merge two tables using set operations such as UNION, UNION ALL, INTERSECT, and EXCEPT.

- Understood the use of subqueries and nested queries to solve problems like finding top earners or lowest salaries.

- Practiced using aggregate functions like MAX() and MIN() along with GROUP BY for department-wise or employee-wise analysis.

- Learnt how to handle duplicate records and ensure correct results while combining datasets.

- Understood how SQL queries can be applied in real-world HR systems to analyze employee data and reporting hierarchies.