Experiment- 3

Student Name: RAVI **UID:23BCS10340**
Branch: CSE **Section/Group: KRG-3A**
Semester: 5th **Date of Performance: 21 Aug**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1) Problem Statement :

You are given an Employee table with a single attribute Emp_Id containing values: 2, 4, 4, 6, 6, 7, 8, 8, 8. Write a SQL query using only sub-queries to find the maximum Emp_Id after excluding all duplicate employee IDs.

TOOLS USED : Microsoft SQL Server

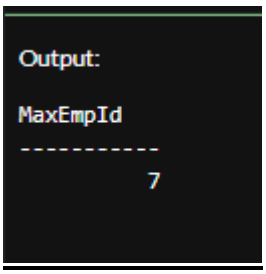
SQL CODE :

```
create table employee(
emp_id int);

INSERT INTO EMPLOYEE VALUES(2),(4),(4),(6),(6),(7),(8),(8),(8);

SELECT MAX(Emp_Id) AS MaxEmpId
FROM (
   SELECT Emp_Id
   FROM Employee
   GROUP BY Emp_Id
   HAVING COUNT(Emp_Id) = 1
) AS Unique_Employees;
```

OUTPUT:



2) Problem Statement :

Department Salary Champions

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.

TOOLS USED : Microsoft SQL Server

SQL CODE:

```
CREATE TABLE department (
    id INT PRIMARY KEY,
    dept_name VARCHAR(50)
);
CREATE TABLE employee (
    id INT,
    name VARCHAR(50),
    salary INT,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES department(id)
);


INSERT INTO department (id, dept_name) VALUES
(1, 'IT'),
(2, 'SALES');


INSERT INTO employee (id, name, salary, department_id) VALUES
```

```
(1, 'JOE', 70000, 1),

(2, 'JIM', 90000, 1),

(3, 'HENRY', 80000, 2),

(4, 'SAM', 60000, 2),

(5, 'MAX', 90000, 1);


SELECT D.dept_name, E.name, E.SALARY

FROM EMPLOYEE AS E

INNER JOIN

DEPARTMENT AS D

ON

E.department_id = D.id

WHERE E.SALARY IN   --90000

(

    SELECT MAX(E2.SALARY)

    FROM EMPLOYEE AS E2

    WHERE E2.department_id = E.department_id -- 90000

)

ORDER BY D.dept_name;
```

OUTPUT :

```
Output:

dept_name                                          name                                               SALARY
-------------------------------------------------- -------------------------------------------------- -----------
IT                                                 MAX                                                      90000
IT                                                 JIM                                                      90000
SALES                                              HENRY                                                    80000
```

3) Problem Statement :

Merging Employee Histories : Who Earned Least?

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.

TOOLS USED : Microsoft SQL Server

SQL CODE:

```
CREATE TABLE A (
   EmpId INT,
   EName VARCHAR(50),
   Salary INT
);
CREATE TABLE B (
   EmpId INT,
   EName VARCHAR(50),
   Salary INT
);
INSERT INTO A VALUES(1,'AA',1000),(2,'BB',300);
INSERT INTO B VALUES(2,'BB',400),(3,'CC',100);
SELECT EMPID, ENAME, MIN(SALARY) AS SALARY
FROM
(
SELECT *FROM A
UNION ALL
SELECT *FROM B
) AS INTERMEDIATE_RESULT
GROUP BY EMPID, ENAME
```

OUTPUT:

Output:

EMPID	ENAME	SALARY
1	AA	1000
2	BB	300
3	CC	100