EXPERIMENT : 3

NAME : Nikhil Verma   **UID: 23BCS14095**

BRANCH : BE-CSE   **SECTION/GROUP : KRG_1A**

SEMESTER : 5TH   **SUBJECT CODE : 23CSP-339**

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

[EASY]

Generate an employee relation with only one attribute i.e., EMP_ID. Then, find the max EMP_ID, but excluding the duplicates.

[MEDIUM]

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.

[HARD]

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.

2. Tools Used : SQL Server Management Studio

3. Code :

```
----------------------- EASY -------------------------
CREATE TABLE TBL_EMPLOYEE(
EMP_ID INT
 );
 INSERT INTO TBL_EMPLOYEE VALUES (2),(4),(4),(6),(6),(7),(8),(8);
 SELECT MAX(EMP_ID) as [Greatest Unique ID] FROM TBL_EMPLOYEE WHERE
EMP_ID IN
 (SELECT EMP_ID FROM TBL_EMPLOYEE GROUP BY EMP_ID HAVING
COUNT(EMP_ID)=1);

----------------------- MEDIUM -------------------------

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, d.id
from
 employee as e
inner join
department as D
 on e.department_id=d.id
 where e.salary in (Select max(salary) from employee group by department_id);

------------------------- HARD -------------------------

create table tbl_A (
 empid int PRIMARY key,
empname varchar(20),
salary int
 )
insert into tbl_A values (1,'AA',1000), (2, 'BB',300);
create table tbl_B (
 empid int PRIMARY key,
empname varchar(20),
salary int
 )
 insert into tbl_B values (2, 'BB',400), (3,'CC',100);
 select empid, min(empname) as empname, min(salary) as min_salary from
(select * FROM
 tbl_A
UNION
 select * from
```

```
tbl_b) as UNI
group by empid;
```

4. Output :
[EASY]

	Greatest Unique ID
1	7

[MEDIUM]

100 % ▼ ✔ No issues found

⊞ Results 📋 Messages

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

[HARD]

100 % ▼ ✔ No issues found

⊞ Results 📋 Messages

	empid	empname	min_salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

5. Learning Outcomes :

- Ability to design and create tables with appropriate attributes and constraints (Primary Key, Foreign Key).

- Skills in data insertion and handling duplicates effectively.

- Stronger understanding of aggregate functions like MAX, MIN, and how they apply in practical queries.

- Experience with GROUP BY and HAVING clauses to filter and summarize data.

- Proficiency in using subqueries and correlated subqueries to solve business problems.

- Ability to perform table joins (INNER JOIN) to relate data across multiple entities.

- Knowledge of UNION for combining datasets from different sources.