Experiment - 3

Student Name: Gurshaan Singh
UID: 23BCS14195
Branch: BE-CSE
Section/Group: KRG-1 (B)
Semester: 5th
Date: 19-08-25
Subject Name: ADBMS
Subject Code: 23CSP-333

Aim:

Q1. Max Value (Easy)
We are given the table Employee (EMP_ID): 2, 3, 4, 5, 6, 7, 8.
Task: Find the max value of EMP_ID but excluding duplicates (using sub queries).
HINT: GROUP BY - GROUPS OF UNIQUE ELEMENTS
OUTPUT: 7.

Q2. Department Salary Champions (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 result should present the department name, employee name, and salary of these top-tier professionals, arranged by department.

Q3. Merging Employee Histories: Who Earned Least? (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 EMP_ID) along with their lowest recorded salary across both systems.

- Combine the two tables A and B.
- Return each EMP_ID with their lowest salary, and the corresponding Ename.

Objective:

Q1: Max Value (Easy)
The objective is to find the maximum unique EMP_ID from the Employee table by using subqueries, ensuring that duplicate values are excluded.

Q2: Department Salary Champions (Medium)

The objective here is to identify the highest-paid employee or employees in each department, displaying the department name, employee name, and salary while handling cases where multiple employees share the top salary, and arranging the results by department.

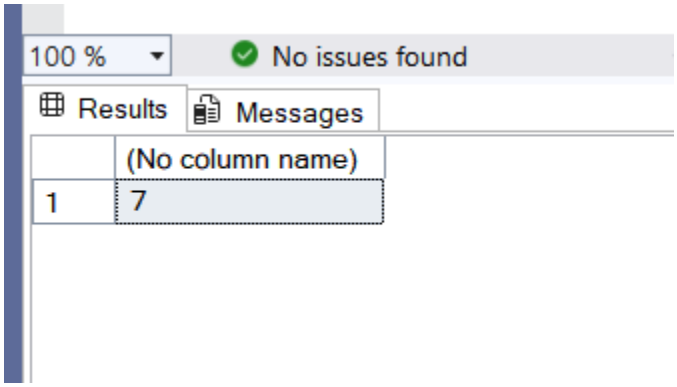
Q3: Merging Employee Histories: Who Earned Least? (Hard)

The objective here is to merge salary records from two legacy HR systems, identify each unique employee, and determine their lowest recorded salary across both systems, displaying the EMP_ID, employee name, and the minimum salary.

DBMS Code & Output:

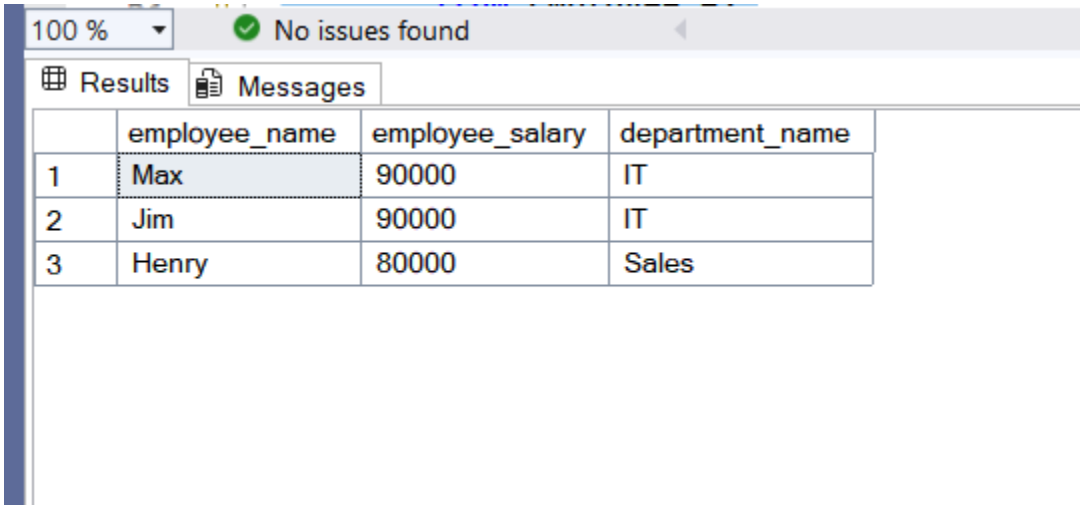
Q1: Max Value (Easy)

```
use KRG_1B;

-- Employees table
create table Employee (
   emp_id int
);

-- Sample data
insert into
   Employee
values
   (1),(2),(3),(4),(5),(6),(7),(8),(8),(9),(9);

-- Query to find the maximum EMP_ID excluding duplicates
select
   max(emp_id)
from (
   select emp_id
   from Employee
   group by emp_id
   having count(emp_id) = 1
) as UniqueEmployees;
```



Q2: Department Salary Champions (Medium)

```
use KRG_1B;

-- Departments table
create table Department (
    id int primary key,
    dept_name varchar(50)
);

-- Employees table with foreign key reference to Department
create table Employee (
    id int,
    name varchar(50),
    salary int,
    department_id int,
    foreign key (department_id) references Department(id)
);

-- Sample data for departments
insert into Department
values
    (1, 'IT'),
    (2, 'Sales');

-- Sample data for employees
insert into Employee
values
    (1, 'Joe', 70000, 1),
    (2, 'Jim', 90000, 1),
    (3, 'Henry', 80000, 2),
    (4, 'Sam', 60000, 2),
    (5, 'Max', 90000, 1);

-- Display all employees
select * from Employee;
-- Display all departments
select * from Department;

-- Query to find the highest-paid employee(s) in each department
select
    e.name as 'employee_name',
    e.salary as 'employee_salary',
    d.dept_name as 'department_name'
from
    Employee e
inner join
    Department d
on
    e.department_id = d.id
where
    e.salary in (
        select max(e2.salary)
        from Employee e2
        where e2.department_id = e.department_id
    )
order by
    d.dept_name;
```

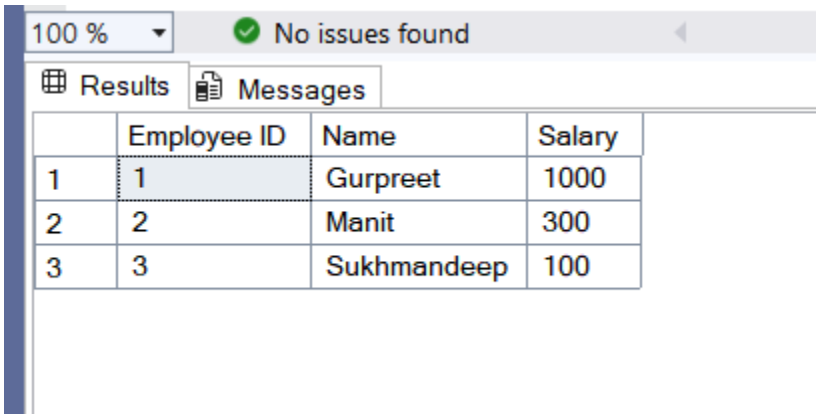


	employee_name	employee_salary	department_name
1	Max	90000	IT
2	Jim	90000	IT
3	Henry	80000	Sales

Q3: Merging Employee Histories: Who Earned Least? (Hard)

```
use KRG_1B;

-- TableA to store employee salaries from system A
create table TableA (
    emp_id int primary key,
    ename varchar(50),
    salary int
);

-- TableB to store employee salaries from system B
create table TableB (
    emp_id int primary key,
    ename varchar(50),
    salary int
);

-- Sample data for TableA
insert into TableA
values
    (1, 'Gurpreet', 1000),
    (2, 'Manit', 300);

-- Sample data for TableB
insert into TableB
values
    (2, 'Manit', 400),
    (3, 'Sukhmandeep', 100);

-- Query to find each unique employee with their lowest salary across both tables
select
    emp_id as 'Employee ID',
    ename as 'Name',
    min(salary) as 'Salary'
from (
    select * from TableA
```

```
    union all
    select * from TableB
) as Combined_Salaries
group by
    emp_id, ename;
```



	Employee ID	Name	Salary
1	1	Gurpreet	1000
2	2	Manit	300
3	3	Sukhmandeep	100