WORKSHEET 3

Student Name: Supriya Dutta	**UID: 23BCS13291**
Branch: CSE(3rd Year)	**Section/Group: Krg-1-A**
Semester: 5th	**Date of Performance: 14/08/25**
Subject Name: ADBMS	**Subject Code: 23CSP-333**

1. **AIM:**

[EASY]

Task:

Write and execute a SQL query that returns the maximum EMP_ID value from the Employee table excluding any EMP_IDs that appear more than once (i.e., only consider IDs that occur exactly once). Verify the result with the given sample data.

Objectives:

- Understand grouping and filtering duplicate rows using GROUP BY and HAVING.
- Use a subquery in the WHERE clause to select only non-duplicated EMP_IDs.
- Return the maximum of those unique EMP_IDs using an aggregate function.

Expected Outcome:

- The query should return **7** for the provided sample data (since 7 is the largest EMP_ID that occurs exactly once).

[MEDIUM]

Task:

Write and execute a SQL query to return the employee(s) with the **highest salary in each department**, along with their department name, using the given employeee and department tables.

Objectives:

- Practice joining two related tables using INNER JOIN.
- Use correlated subqueries or aggregation (MAX) to find department-wise maximum salaries.
- Display employee name, salary, and corresponding department name.

Expected Outcome:

- For the given data, the result should show:
  - Department IT: Employees **JIM** and **MAX** with salary 90000.
  - Department SALES: Employee **HENRY** with salary 80000.

[HARD]
Task:
Given two tables TableA and TableB containing employee IDs, names, and salaries, write an SQL query to return each employee's **ID, name, and their lowest salary** across both tables.
Objectives:

- Understand how to combine data from multiple tables using UNION ALL.
- Use aggregation functions (MIN) to find the lowest salary per employee.
- Group results by employee ID and name to ensure correct mapping.

Expected Outcome:

- For the given data:
  - Employee **AA** → Salary 1000
  - Employee **BB** → Salary 300 (minimum of 300 and 400)
  - Employee **CC** → Salary 100

2. **Tools Used :** SQL Server Management Studio

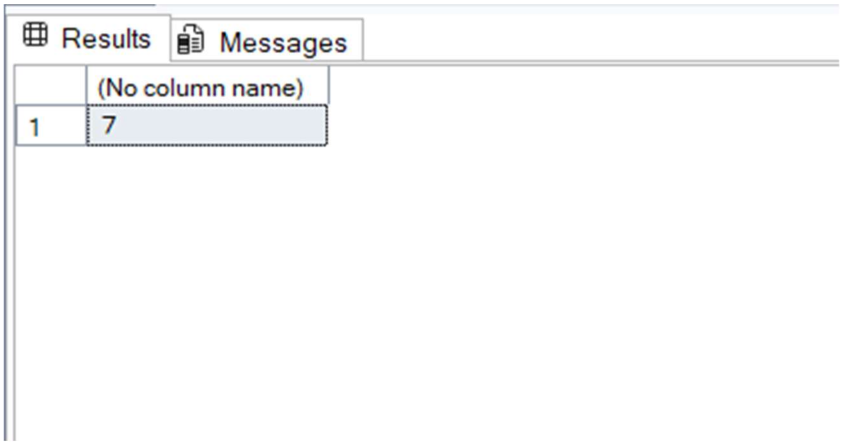
DBMS SCRIPT:

```
--Q1 (Easy level)

CREATE TABLE Employee(
EMP_ID INT
)

INSERT into Employee(EMP_ID) values (2)
INSERT into Employee(EMP_ID) values (4)
INSERT into Employee(EMP_ID) values (4)
INSERT into Employee(EMP_ID) values (6)
INSERT into Employee(EMP_ID) values (6)
INSERT into Employee(EMP_ID) values (7)
INSERT into Employee(EMP_ID) values (8)
INSERT into Employee(EMP_ID) values (8)

-- return the max empid exlcuding the duplicates using subqueries e.g in this case 7

SELECT Max(EMP_ID) from Employee
where EMP_ID IN
(
select EMP_ID from Employee
group by EMP_ID
having count(*)=1
)

--Q2 (medium level)
CREATE TABLE department (
   id INT PRIMARY KEY,
   dept_name VARCHAR(50)
);

-- Create Employee Table
CREATE TABLE employeee (
    id INT,
    name VARCHAR(50),
    salary INT,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES department(id)
);

-- Insert into Department Table
INSERT INTO department (id, dept_name) VALUES
(1, 'IT'),
(2, 'SALES');

-- Insert into Employee Table
INSERT INTO employeee (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);

--max salary dept wise

--approach 1:
SELECT d.dept_name, e.name, e.salary
FROM employeee e
JOIN
department d
ON
e.department_id = d.id
WHERE e.salary = (
    SELECT MAX(salary)
    FROM employeee
    WHERE department_id = e.department_id
);

--approach 2
SELECT d.dept_name, e.name, e.salary, d.id
FROM employeee as e
inner JOIN
department as d
ON
e.department_id = d.id
```

```
WHERE e.salary in (
    SELECT MAX(E2.salary)
    FROM Employeee as E2
    group by E2.department_id
);



--q3(hard level)

create table TableA(
    Emp_id int,
    Ename varchar(50),
    salary int
)
create table TableB(
    Emp_id int,
    Ename varchar(50),
    salary int
)

INSERT into TableA(Emp_id, Ename, salary) values
(1, 'AA', 1000),
(2, 'BB', 300);

INSERT into TableB(Emp_id, Ename, salary) values
(2, 'BB', 400),
(3, 'CC', 100);

--return each empid with their lowest salary and corrsponding ename

select Emp_id ,Ename, min(salary) as min_salary
from
(
    select emp_id, ename, salary from TableA
    union all
    select emp_id, ename, salary from TableB
)
TableA
group by
Emp_id, Ename;
```

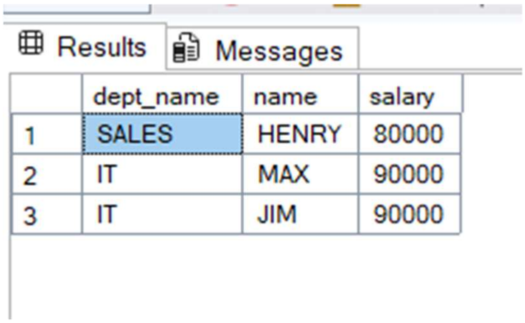
3. OUTPUT:

--Medium Level:



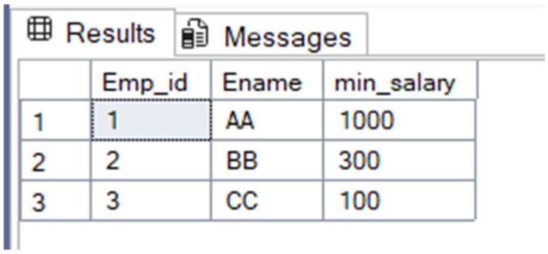
	(No column name)
1	7

--Medium Level:



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

--Hard Level:



	Emp_id	Ename	min_salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

4. Learning Outcomes:

- **Apply filtering techniques** using `GROUP BY` and `HAVING` to eliminate duplicate records in SQL queries.
- **Utilize subqueries effectively** within `WHERE` clauses to extract meaningful subsets of data.
- **Implement aggregate functions** (`MAX`, `MIN`, etc.) in combination with joins and subqueries to answer complex queries.
- **Integrate data from multiple tables** using `JOIN` and `UNION ALL` to derive consolidated results.
- **Analyze and design queries for real-world scenarios**, ensuring accurate results from relational database structures.