



# **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

## **Experiment 3**

**Student Name: Suman**  
**Branch: BE CSE**  
**Semester: 5**  
**Subject : ADBMS**

**UID: 23BCS12099**  
**Section/Group: KRG-3-A**  
**Date of Performance: 14-8-25**  
**Subject Code: 23CSP-333**

### **1. Aim:**

1. Generate an employee relation with only one attribute i.e., EMP\_ID. Then, find the max EMP\_ID, but excluding the duplicates.
2. Create two tables, Department(ID, name) and Employees(ID, name, salary, deptID). Then output the highest earners from each department.
3. Create two tables A and B with the attributes (EmpID, EmpName, Salary) and output the lowest salary of each employee across the two tables.

### **2. Requirements (Hardware/Software):**

Microsoft SQL server

### **3. Procedure:**

#### **Q.1. Code:**

```
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);
```

**Q.2. Code:**

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

-- Create Employee Table

```
CREATE TABLE employee (  
  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 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);
```

**Q.3. Code:**

```
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:

Q.1.

Output:

```
Greatest Unique ID
-----
7
```

Q.2.

Output:

dept_name	name
IT	JIM
IT	MAX
SALES	HENRY

Q.3.

Output:

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

## **5. Learning Outcome:**

- **Understand the role of sub-queries in simplifying complex SQL operations.**
- **Apply sub-queries in SELECT, WHERE, and FROM clauses to retrieve specific data.**
- **Utilize sub-queries for filtering, aggregation, and conditional logic.**
- **Analyze query performance implications when using sub-queries versus joins.**