



**University Institute of Engineering**  
**Department of Computer Science & Engineering**

**EXPERIMENT:3**

**NAME : SATYA PRAKASH**  
**BRANCH : BE-CSE**  
**SEMESTER : 5<sup>TH</sup>**  
**SUBJECT NAME : ADBMS**

**UID : 23BCS11072**  
**SECTION : KRG\_1A**  
**SUBJECT : 23CSP-339**

**1. AIM:-**

[ EASY ]

You are given an EMP table that contains a list of employee IDs (EMP\_ID). Some employee IDs may appear multiple times, representing duplicate entries.

Write an SQL query (using **subqueries**) to:

- Identify and exclude all employee IDs that appear more than once in the table.
- From the remaining unique employee IDs, find the **highest employee ID**.

Return the result as a single column named single\_highest.

[MEDIUM]

Given tables:

- department(id, dept\_name)
- employee(id, name, salary, department\_id)

Write a SQL query to retrieve employees with the **highest salary in each department**, displaying their name, salary, and department name, sorted by department name.

[HARD]

Given tables:

- TABLE1(EMPID, Ename, Salary)
- TABLE2(EMPID, Ename, Salary)

Write a SQL query to combine the records from both tables, and for each EMPID, select the employee name and salary with the **minimum values**. The result should display one row per EMPID.

## 2.TOOLS USED :-

SQL server management studio.

## 3.CODE:-

--EASY--

```
use subqueries
CREATE TABLE
EMP( EMP_ID INT
)
INSERT INTO EMP (EMP_ID)
VALUES (2), (4), (4), (6), (6), (7), (8), (8);
```

--SELECT \*FROM EMP;

```
SELECT MAX(EMP_ID) AS [single_highest]FROM EMP
WHERE EMP_ID NOT IN(
SELECT EMP_ID FROM EMP
GROUP BY EMP_ID
HAVING COUNT(EMP_ID)>1
```

--MEDIUM--

```
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 E.name,E.salary,D.dept_name
FROM employee AS E
LEFT JOIN
department as D
On
E.department_id=D.id
```

```

WHERE E.salary IN
(
    SELECT MAX(E2.SALARY)
    FROM employee as E2
    WHERE E2.department_id =E.department_id
)
ORDER BY D.dept_name

```

**--HARD--**

```

CREATE TABLE
TABLE1( EMPID INT,
Ename VARCHAR(20),
Salary INT
)
CREATE TABLE
TABLE2( EMPID INT,
Ename VARCHAR(20),
Salary INT
)

INSERT INTO TABLE1(EMPID,Ename,Salary) VALUES
(1, 'AA',1000),
(2, 'BB',300);

INSERT INTO TABLE2(EMPID,Ename,Salary) VALUES
(2, 'BB',400),
(3, 'CC',100);

SELECT EMPID,min(Ename) as Ename,MIN(Salary) as Salary
FROM
(
SELECT *FROM TABLE1
UNION
SELECT *FROM TABLE2
) AS RES
GROUP BY EMPID

```

#### 4.OUTPUT:- [EASY]

100 %	✓ No issues found
Results	Messages
	single_highest
1	7

#### [MEDIUM]

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

[HARD]

Results		Messages	
	EMPID	Ename	Salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

## 5. LEARNING OUTCOMES:-

- Acquired hands-on experience in creating tables and inserting data in SQL.
- Practiced writing subqueries for filtering and data aggregation.
- Gained proficiency in using JOINS to combine and analyze data from multiple tables.
- Learned techniques to manage duplicates and consolidate results using UNION and aggregate functions.
- Strengthened problem-solving skills in retrieving and interpreting specific information from databases.