CHANDIGARH UNIVERSITY

University Institute of Engineering

Department of Computer Science & Engineering

Experiment:3

Date of Experiment: 21-08-2025

1. Aim of the practical:

[EASY] You are given with employee table with only one attribute that is emp_id which contains values as:

Employee (emp_id)

2
4
4
6
6
7
8
8
8

Task: find the maximum value for emp_id, but excluding the duplicate employee id's. (Only with subqueries) Output: 7

Explanation: if we exclude duplicates such as, 4, 6, and 8, & from the rest i.E., 2,7 the maximum is 7.

[MEDIUMI Department Salary Champions:

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.

[HARDI Merging Employee Histories: Who Earned Least?

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 EmplD) along with their lowest recorded salary across both systems. Objective:

1. Combine two tables A and B.
2. Return each EmplD with their lowest salary, and the corresponding Ename.

2. Tools used: SQL Server Management Studio

3. Queries:

```
/*EASY*/

CREATE TABLE Employee (
EMP_ID INT
);
INSERT INTO Employee (EMP_ID) VALUES
(2), (4), (4), (6), (6), (7), (8), (8), (8);

SELECT * FROM Employee;

SELECT MAX(EMP_ID) AS MaxEmpID
FROM Employee
WHERE EMP_ID NOT IN (
   SELECT EMP_ID
   FROM Employee
   GROUP BY EMP_ID
   HAVING COUNT(*) > 1
);

/*MEDIUM*/
CREATE TABLE EMP_TBL(
ID INT PRIMARY KEY,
NAME VARCHAR(60),
SALARY INT,
DEPT_ID INT
);
CREATE TABLE DEPT_TBL(
ID INT PRIMARY KEY,
DEPT_NAME VARCHAR(100)
);


INSERT INTO EMP_TBL(ID,NAME,SALARY,DEPT_ID) VALUES
(1,'JOE',70000,1),  (2,'JIM',90000,1),  (3,'HENRY',80000,2),
(4,'SAM',60000,2),  (5,'MAX',90000,1);

INSERT INTO DEPT_TBL(ID,DEPT_NAME) VALUES
(1,'IT'), (2,'SALES');

SELECT * FROM EMP_TBL;
SELECT * FROM DEPT_TBL;
```

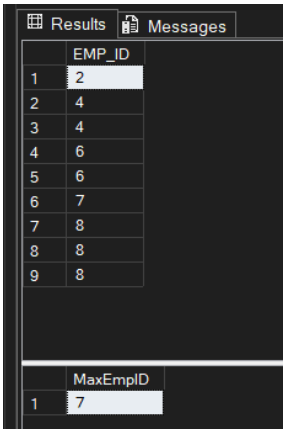
SELECT D.DEPT_NAME AS [DEPARTMENT NAME],
E.NAME AS [EMPLOYEE NAME], E.SALARY
FROM EMP_TBL AS E
INNER JOIN DEPT_TBL AS D
ON E.DEPT_ID = D.ID
WHERE E.SALARY IN(
SELECT MAX(SALARY)
FROM EMP_TBL WHERE DEPT_ID=E.DEPT_ID)
ORDER BY D.ID;

/*HARD*/
CREATE TABLE TBL_A(
EmpID INT PRIMARY KEY,
Ename VARCHAR(100),
Salary INT
);
CREATE TABLE TBL_B(
EmpID INT PRIMARY KEY,
Ename VARCHAR(100),
Salary INT
);

INSERT INTO TBL_A(EmpID,Ename,Salary) VALUES
(1,'AA',1000),
(2,'BB',300);
INSERT INTO TBL_B(EmpID,Ename,Salary) VALUES
(2,'BB',400),
(3,'CC',100);
SELECT * FROM TBL_A;
SELECT * FROM TBL_B;
SELECT EmpID,MIN(Ename) AS Ename,MIN(Salary) AS [Salary]
FROM (
  SELECT EmpID,Ename,Salary FROM TBL_A
    UNION ALL
    SELECT EmpID,Ename,Salary FROM TBL_B
) AS MERGEDDATA
GROUP BY EmpID;

4. Output:

EASY-

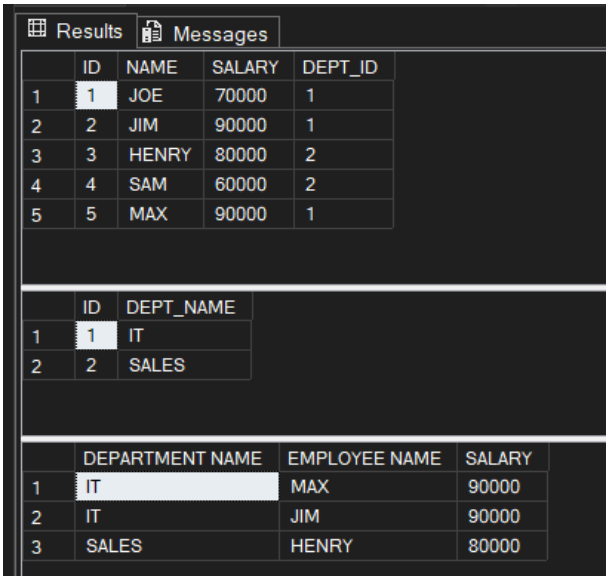


Results | Messages

	EMP_ID
1	2
2	4
3	4
4	6
5	6
6	7
7	8
8	8
9	8

	MaxEmpID
1	7

MEDIUM-



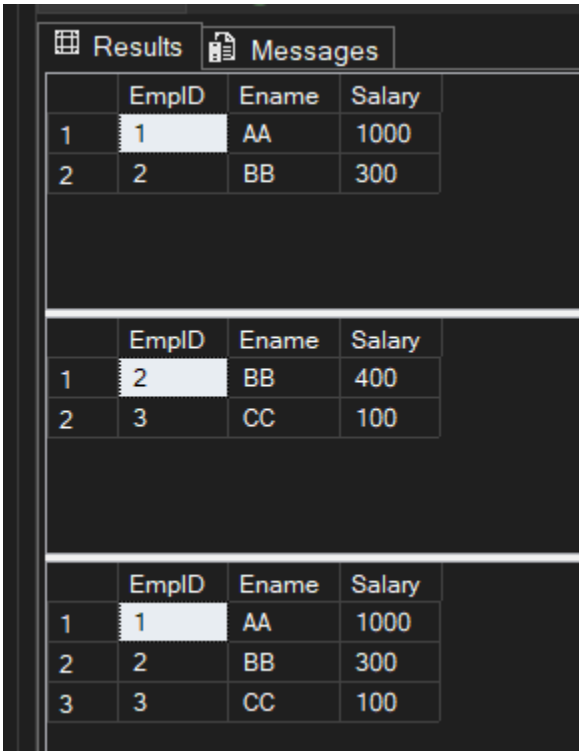
Results | Messages

	ID	NAME	SALARY	DEPT_ID
1	1	JOE	70000	1
2	2	JIM	90000	1
3	3	HENRY	80000	2
4	4	SAM	60000	2
5	5	MAX	90000	1

	ID	DEPT_NAME
1	1	IT
2	2	SALES

	DEPARTMENT NAME	EMPLOYEE NAME	SALARY
1	IT	MAX	90000
2	IT	JIM	90000
3	SALES	HENRY	80000

HARD-



Results | Messages

	EmpID	Ename	Salary
1	1	AA	1000
2	2	BB	300

	EmpID	Ename	Salary
1	2	BB	400
2	3	CC	100

	EmpID	Ename	Salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

Learning outcomes (What I have learnt):

- Leamed how to identify and exclude duplicate values using GROUP BY with HAVING COUNT(*)> 1.
- Understood the use of subqueries to filter results in SQL Server.
- Applied NOT IN with subqueries for excluding unwanted values.
- Practiced combining aggregate functions like MAX() with subquery filtering.
- Gained experience in designing complete SQL scripts with CREATE TABLE, INSERT, and queries.
- Learned how to create and use databases with multiple related tables.
- Understood how to insert and manage data in relational tables.
- Practiced the use of INNER JOIN to combine employee and department details.
- Gained knowledge of correlated subqueries for row-wise filtering within groups.
- Learned how to retrieve highest salary employees per department using MAX() with subqueries.
- Learned how to merge data from multiple tables using UNION ALL.
- Understood how to apply subqueries as derived tables for further processing.
- Practiced the use of aggregate functions (MIN) to handle duplicate records.
- Gained knowledge of grouping results with GROUP BY to consolidate employee data.
- Learned how to resolve data conflicts (same employee in multiple tables) by applying aggregation.