

Experiment 3

Student Name: Harshdeep Singh

UID: 23BAI70623

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G1

Semester: 5th

Date of Performance: 20th Aug, 2025

Subject Name: ADBMS

Subject Code: 23CSP-333

1. **AIM:** In a bustling corporate organization, each department strives to retain the talented (most and well-compensated) employees. You have access to two key records: one lists every employee along with their salary & 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 final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

2. **Tools Used:**

SQL Server Management Studio 21 (SSMS) code editor.

3. **Experiment:**

Medium-Level

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 final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

Hard-Level

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 EmpID) along with their lowest recorded salary across both systems.

Objective

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

4. Solution:

Medium-Level

```
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)  
); go  
  
-- Insert into Department Table  
INSERT INTO department (id, dept_name) VALUES  
(1, 'IT'),  
(2, 'SALES');  
go  
  
-- 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);  
go  
select d.dept_name, e.name, e.salary  
  
from employee e INNER JOIN department d on e.department_id = d.id where  
salary in  
(  
    select max(e2.salary)  
    from employee e2 where e2.department_id = e.department_id  
)  
order by d.dept_name;  
  
select d.dept_name, e.name, e.salary  
from employee e INNER JOIN department d on e.department_id = d.id where  
salary in  
(  
    select max(e2.salary)  
    from employee e2 group by e2.department_id  
)  
  
order by d.dept_name;
```

Hard-Level

```
create table tab_a(  
    EmpID int primarykey,  
    Ename varchar(5),
```

```

Salary int
); go
creat
e
table
tab_b
(
EmpID
int
prima
ry
key,
Ename varchar(5),
Salary int
); go insert into tab_a values(1, 'AA', 1000), (2, 'BB', 300); go insert
into tab_b values(2, 'BB', 400), (3, 'CC', 100); go
/*

```

SET OP with CTE
columns with duplicate values similar to other columns, have to be
used together in the group-by or with aggregated functions.

```

*/
with res_set as ( select
a.* from tab_a a
UNION ALL
select b.* from tab_b b
)
select EmpID, Ename, min(Salary) as Salary from res_set r
group by EmpID, Ename;

```

5. Output:

Medium-Level

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

Hard-Level

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

6. Learning Outcomes:

- Learned about Sub-Queries and their types.
- Learned about their use cases and execution procedure.
- Learn about how to apply Common Table Expression (WITH).
- Understood how to deal with large datasets.