**CC-215-L: Database Systems Lab**

*Faculty of Computing & Information Technology*



**BS(CS) Morning - Fall 2021, Semester Fall 2024**

**LAB – 06**

**Topics:**

1. Joins
2. Sub Query Part1

**Course & Lab Instructor:** Dr. Asif Sohail

Hope you are fine and feeling yourself comfortable and exciting to play with Structured Query Language (SQL) in this first lab.

*You have been taught in the lecture about SQL and query writing. Why not to get all what you have learned with a hands-on experience?* So, let’s Start!

*Allowed time: 1 hour*

***Note: Do follow these instructions while you are sitting in the lab and performing the tasks.***

1. ***Gossips are not allowed. So be gentle and do what you know. The lab is not to deduct your sessional marks but to prepare you to achieve good marks in quizzes, mids and finals and finally have good grades. So, try to perform all your tasks in time and at your own.***
2. ***Teacher assistants are for your help, so be nice with them. Respect them as they are teaching you. Raise your hands if you have some problem and need help from TA. Avoid calling them by raising your voice and disturbing the environment of Lab.***
3. ***You must revise the content of the past lectures before starting the lab, it will help you resolve most of your general queries and give you the confidence that you can do it.***
4. ***Evaluation will be considered final and you cannot debate for the marks. So, focus on performing the tasks when the time is given to you.***
5. ***TA may deduct your marks for any kind of ill-discipline or misconduct from your side.***
6. ***Finally, pray before you start. And, Best of Luck!***

* **Kindly paste the query as well as result table screenshot as a result of each task**

**Sample:**

**Display All the Employees from emp table**

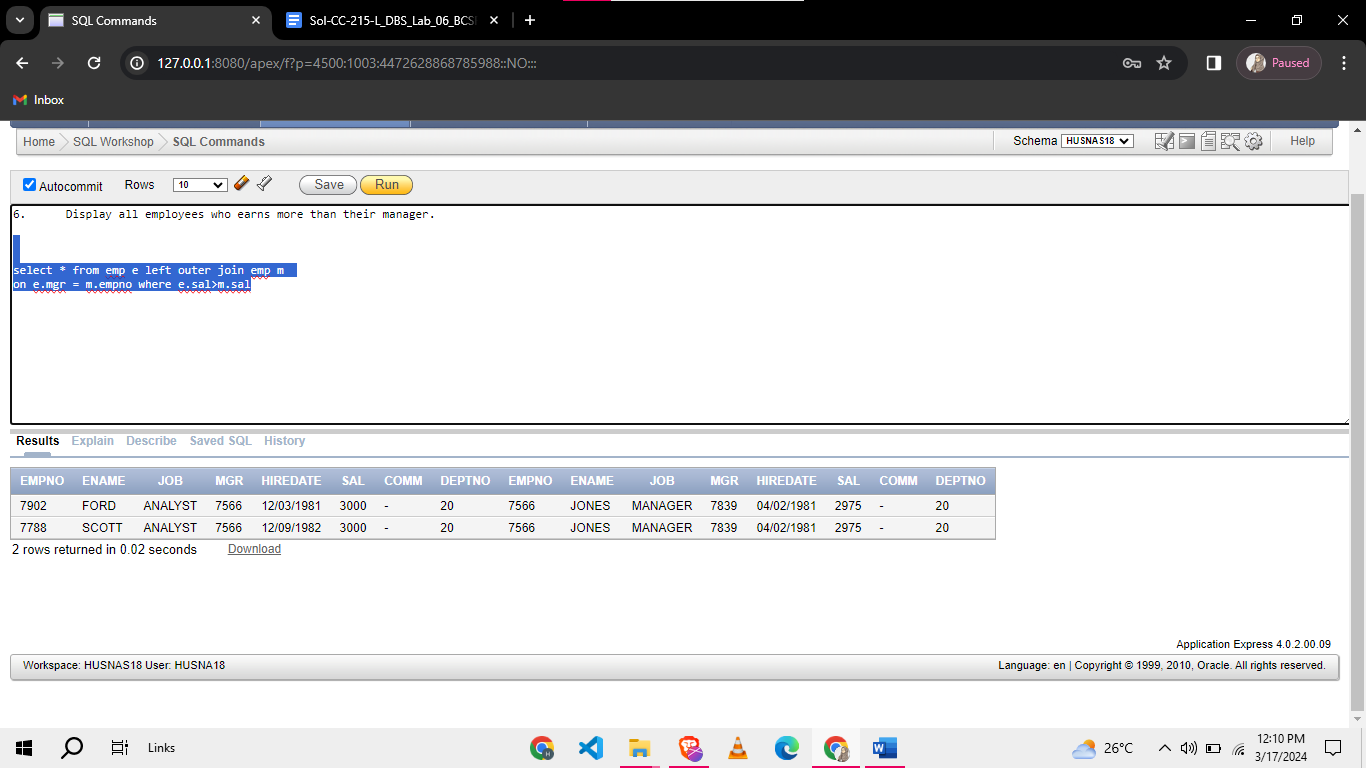
**Solution:**

**Select \* from emp**

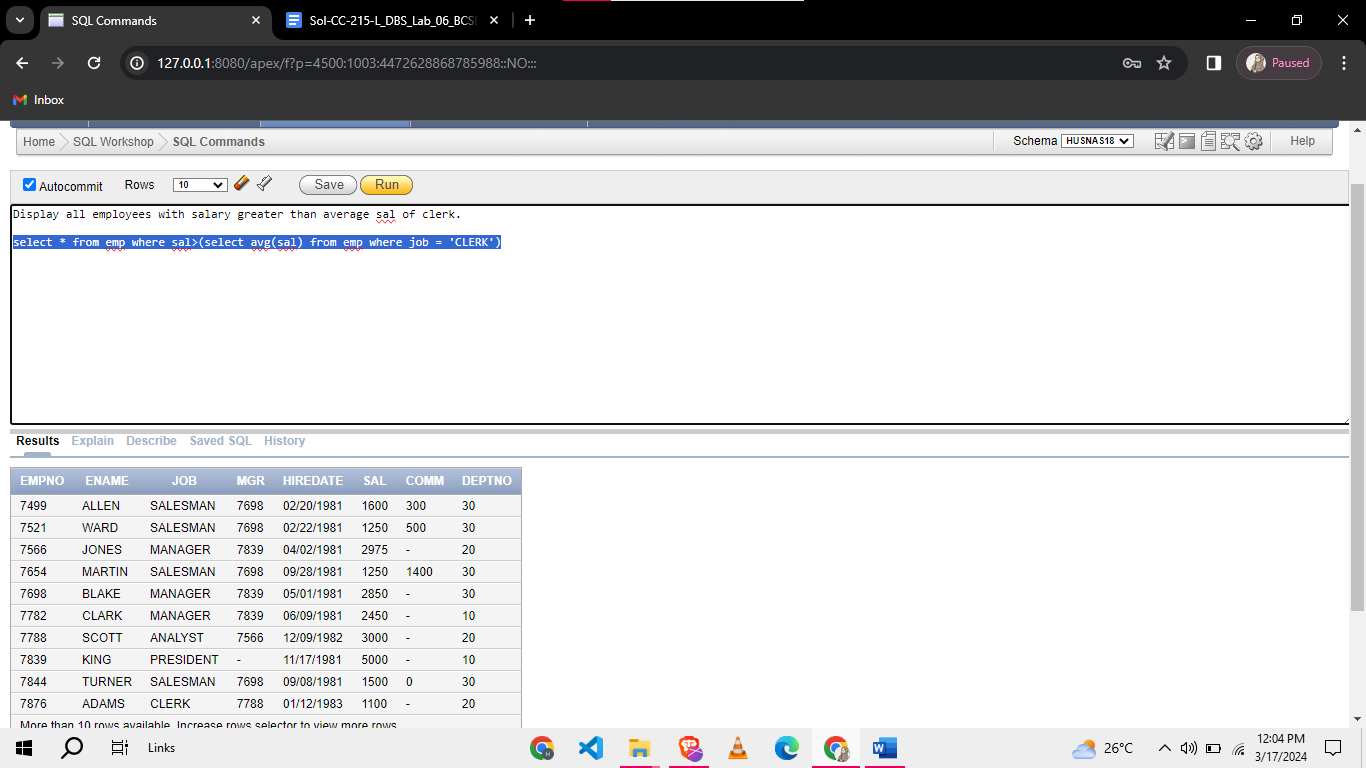
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7369 | SMITH | CLERK | 7902 | 12/17/1980 | 800 | - | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 02/20/1981 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 02/22/1981 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 04/02/1981 | 2975 | - | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 09/28/1981 | 1250 | 1400 | 30 |

**Task 01: [08 Marks]**

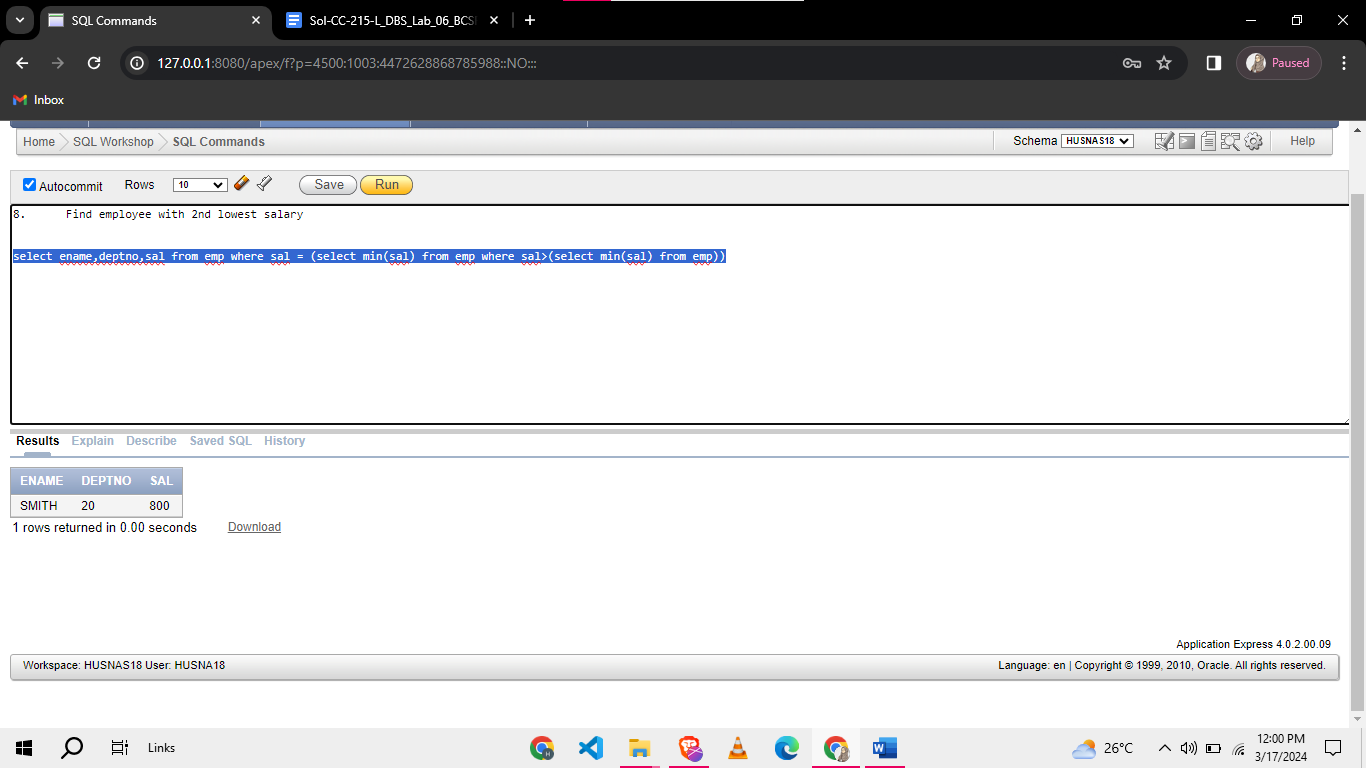
1. Find the Junior most employee.
2. List employees who have a salary greater than the salary of at least one employee in department 10.
3. Find all employees who manage at least one employee using subquery.
4. Find the job with lowest average salary.
5. Display department details which earns most commission.
6. Display all employees who earns more than their manager.



1. Display all employees with salary greater than average sal of clerk.

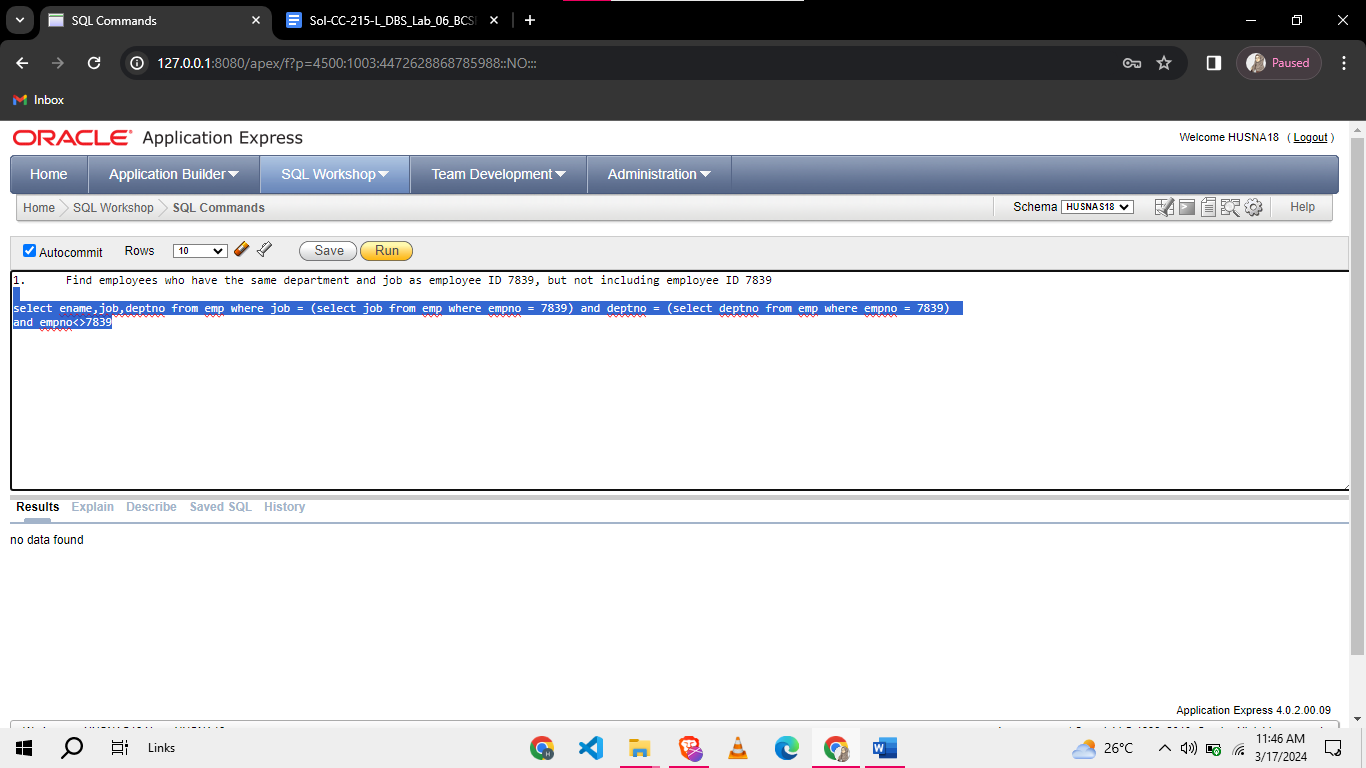


1. Find employee with 2nd lowest salary

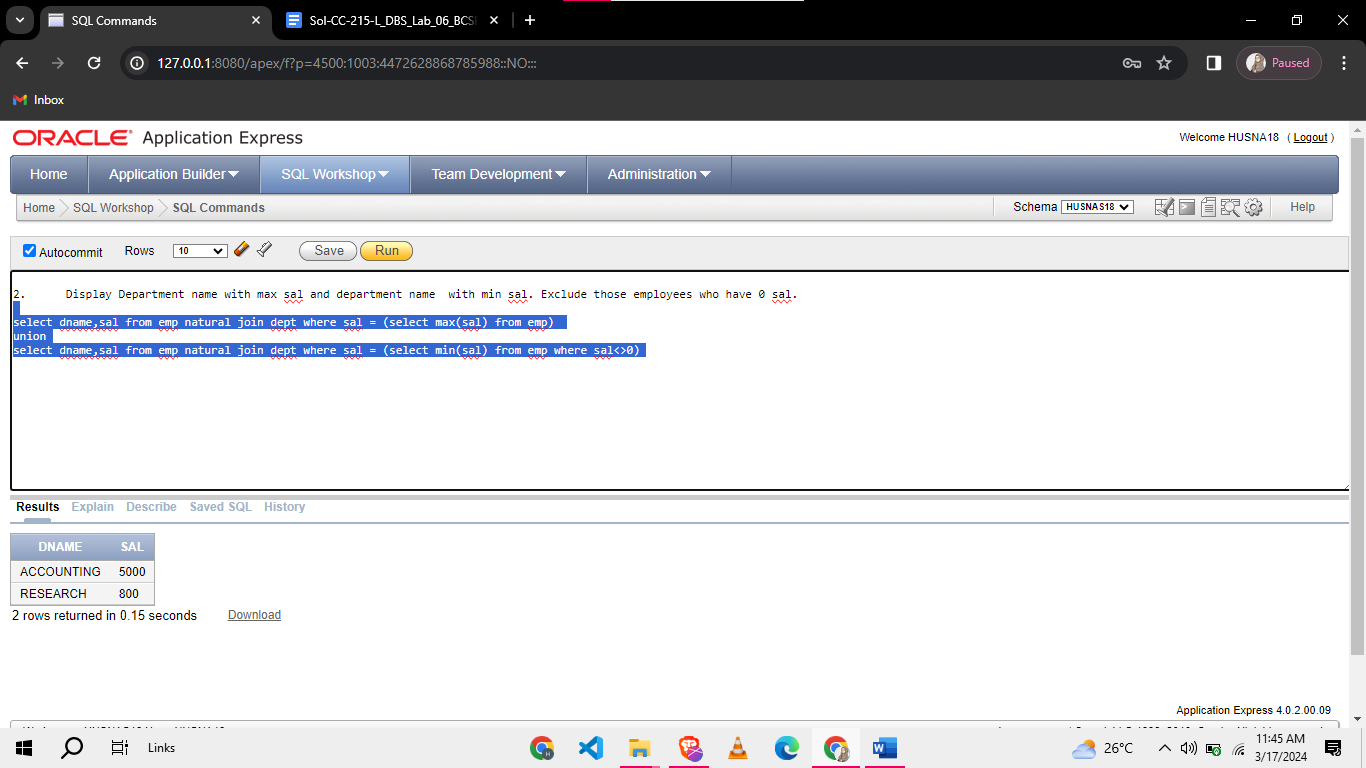


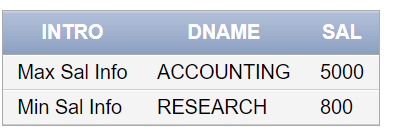
**Task 02: [08 Marks]**

1. Find employees who have the same department and job as employee ID 7839, but not including employee ID 7839

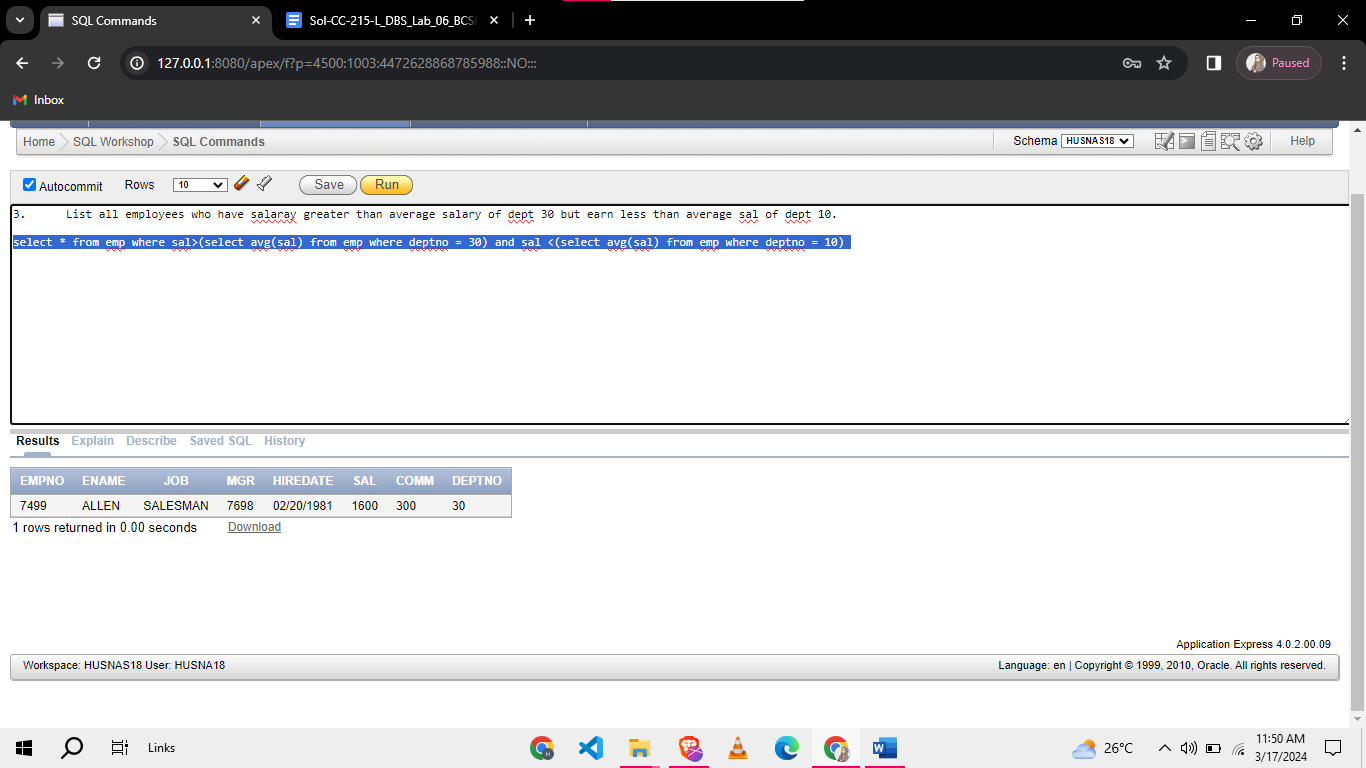


1. Display Department name with max sal and department name with min sal. Exclude those employees who have 0 sal.

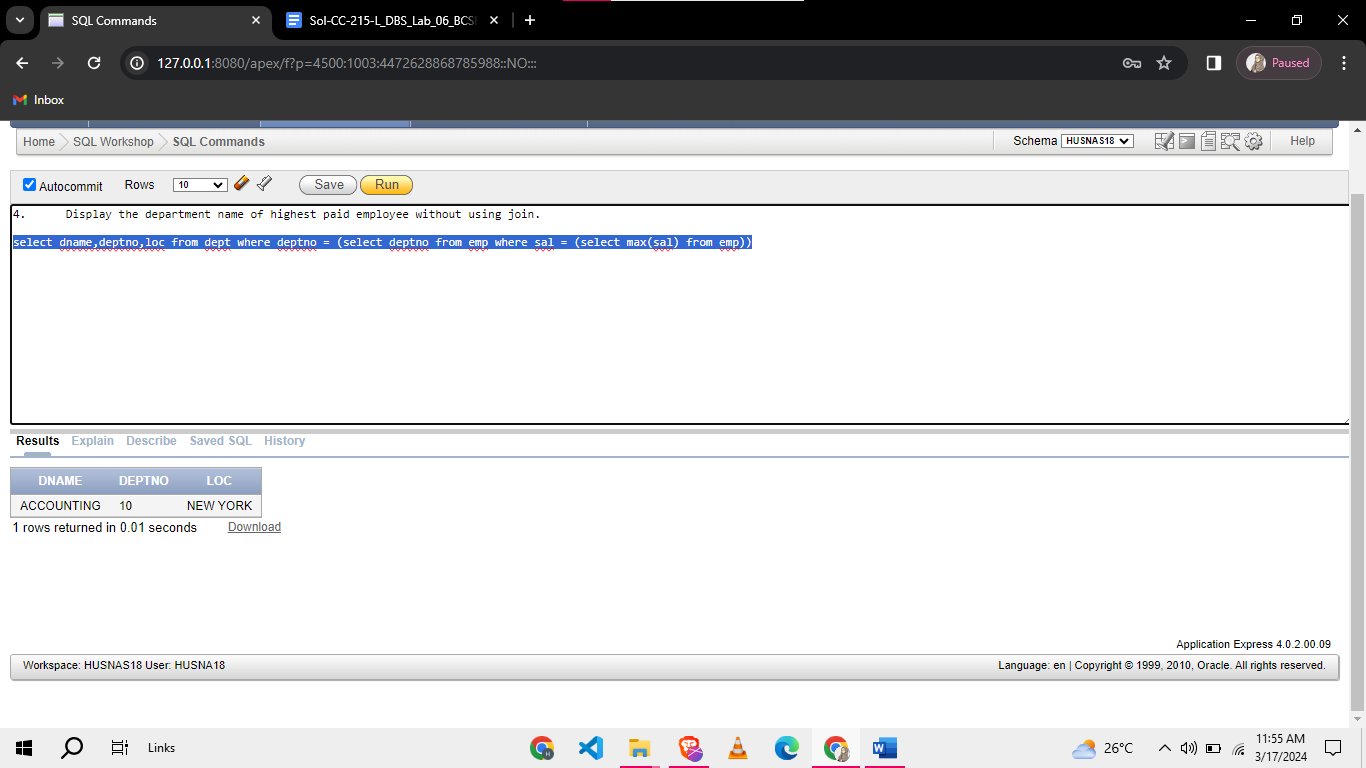




1. List all employees who have salaray greater than average salary of dept 30 but earn less than average sal of dept 10.

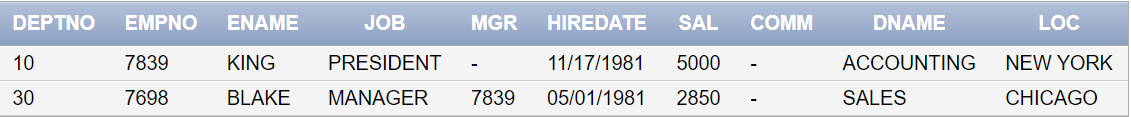


1. Display the department name of highest paid employee without using join.



**Task 03: [09 Marks]**

1. Display Data of those employees who manages 3 or more than 3 employees (3)



**Now let’s try something new. In following Questions, you are given with different Schemas and the Quires. Your Task is to predict the output of each query according to Data**

**(1).**

|  |  |  |  |
| --- | --- | --- | --- |
| **customer\_id** | **customer\_name** | **email** | **phone** |
| 1 | John Smith | john@example.com | 123-456-789 |
| 2 | Alice Brown | alice@example.com | 456-789-123 |
| 3 | Bob Jones | bob@example.com | 789-123-456 |

|  |  |  |  |
| --- | --- | --- | --- |
| **tour\_id** | **tour\_name** | **destination** | **price** |
| 1 | Beach Retreat | Maldives | 2000 |
| 2 | Mountain Hike | Switzerland | 2500 |
| 3 | City Tour | New York | 1000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **booking\_id** | **customer\_id** | **tour\_id** | **booking\_date** | **num\_tickets** |
| 101 | 1 | 1 | 2024-05-10 | 2 |
| 102 | 2 | 2 | 2024-06-15 | 3 |
| 103 | 1 | 3 | 2024-07-20 | 1 |
| 104 | 3 | 2 | 2024-08-05 | 2 |

**From above Tables predict Output of following:**

SELECT c.customer\_name,

t.tour\_name,

b.booking\_date,

b.num\_tickets

FROM bookings b

JOIN customers c ON b.customer\_id = c.customer\_id

JOIN tours t ON b.tour\_id = t.tour\_id

WHERE t.destination = 'Switzerland';

**(2).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Emp\_id** | **Emp\_name** | **Salary** | **Dept\_id** |
| 1 | John | 50000 | 1 |
| 2 | Alice | 60000 | 1 |
| 3 | Bob | 45000 | 2 |
| 4 | Emily | 70000 | 2 |
| 5 | David | 55000 | 1 |

|  |  |
| --- | --- |
| **Dept\_id** | **Dept\_name** |
| 1 | Engineering |
| 2 | Marketing |

**What will be the output of following one:**

SELECT d.dept\_name,

COUNT(e.emp\_id) AS num\_employees,

AVG(e.salary) AS avg\_salary

FROM departments d

LEFT JOIN employees e ON d.dept\_id = e.dept\_id

GROUP BY d.dept\_name

ORDER BY num\_employees DESC;

**(3).**

**What will be output of following.**

|  |  |  |
| --- | --- | --- |
| customer\_id | name | City |
| 101 | Alice | New York |
| 102 | Bob | Los Angeles |
| 103 | Charlie | Chicago |

|  |  |  |  |
| --- | --- | --- | --- |
| Order\_id | Customer\_id | Order\_data | Total\_amount |
| 1 | 101 | 2023-01-15 | 500 |
| 2 | 102 | 2023-01-16 | 700 |
| 3 | 101 | 2023-01-17 | 300 |
| 4 | 103 | 2023-01-18 | 900 |

|  |  |  |
| --- | --- | --- |
| Product\_id | Product\_name | category |
| 1 | Laptop | Electronics |
| 2 | Phone | Electronics |
| 3 | Shirt | Apparel |
| 4 | Jeans | Apparel |

|  |  |
| --- | --- |
| order\_id | **product\_id** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |

SELECT c.city, COUNT(o.order\_id) AS total\_electronics\_orders

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_id IN (

SELECT op.order\_id

FROM order\_products op

JOIN products p ON op.product\_id = p.product\_id

WHERE p.category = 'Electronics'

)

GROUP BY c.city

ORDER BY total\_electronics\_orders DESC;