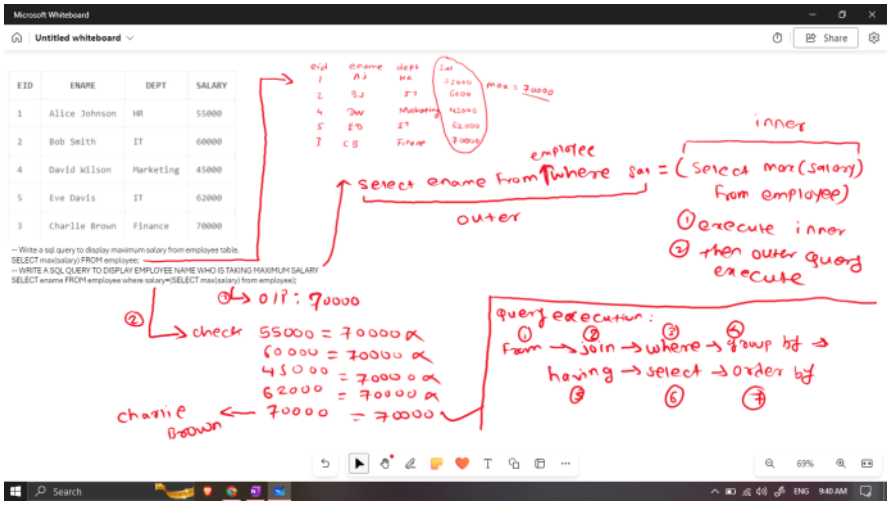
SQL QUERY

Sunday, January 5, 2025 9:05 AM

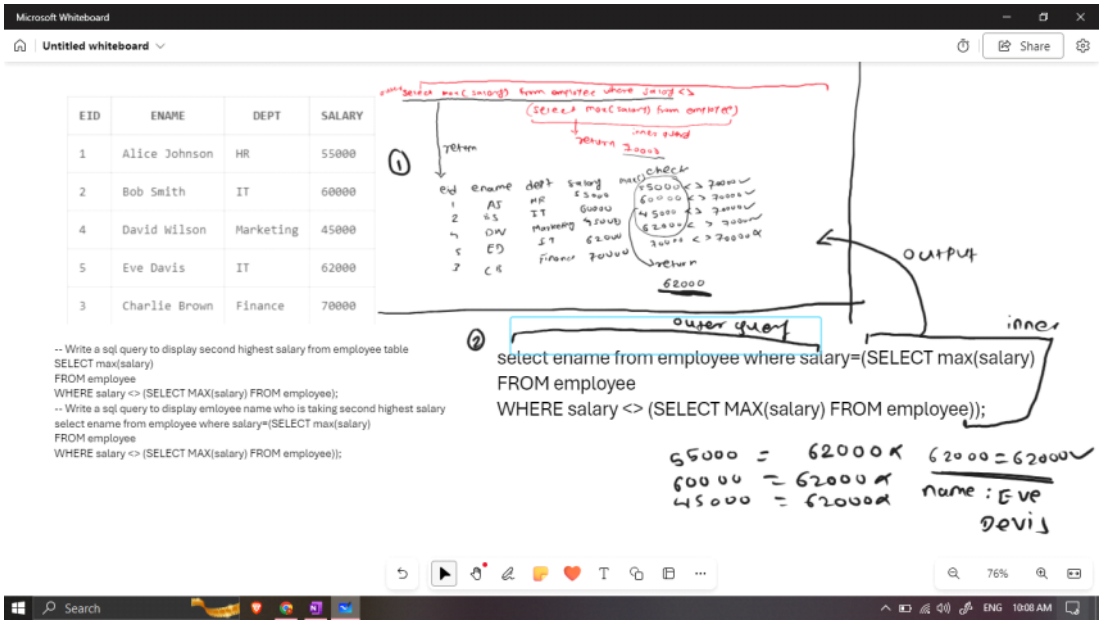
```
CREATE TABLE employee (
   eid INT PRIMARY KEY,
   ename VARCHAR(100),
   dept VARCHAR(50),
   salary DECIMAL(10, 2)
);

INSERT INTO employee (eid, ename, dept, salary) VALUES
(1, 'Alice Johnson', 'HR', 55000.00),
(2, 'Bob Smith', 'IT', 60000.00),
(3, 'Charlie Brown', 'Finance', 70000.00),
(4, 'David Wilson', 'Marketing', 45000.00),
(5, 'Eve Davis', 'IT', 62000.00);

SELECT * FROM EMPLOYEE;
```

EID	ENAME	DEPT	SALARY
1	Alice Johnson	HR	55000
2	Bob Smith	IT	60000
4	David Wilson	Marketing	45000
5	Eve Davis	IT	62000
3	Charlie Brown	Finance	70000

```
-- Write a sql query to display maximum salary from employee table.
SELECT max(salary) FROM employee;
-- WRITE A SQL QUERY TO DISPLAY EMPLOYEE NAME WHO IS TAKING MAXIMUM SALARY
SELECT ename FROM employee where salary=(SELECT max(salary) from employee);
```




```
-- Write a sql query to display second highest salary from employee table
SELECT max(salary)
FROM employee
WHERE salary <> (SELECT MAX(salary) FROM employee);
-- Write a sql query to display emloyee name who is taking second highest salary
select ename from employee where salary=(SELECT max(salary)
FROM employee
WHERE salary <> (SELECT MAX(salary) FROM employee));
```

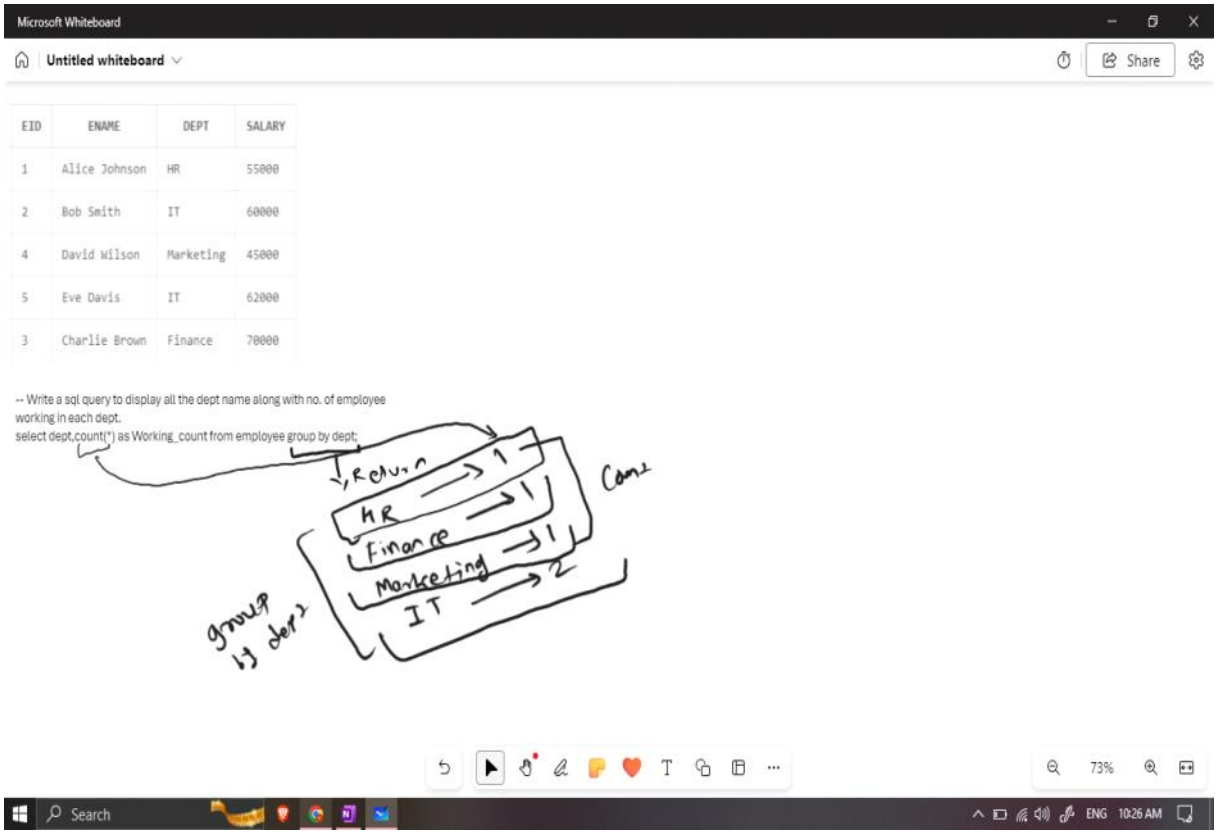


Group By Clause :
GROUP BY is a clause used to group rows that have the same values into summary rows, like finding the total (or aggregate) fo r each group.

Condition :

- **WHERE**: Filters rows before any grouping occurs.
- **HAVING**: Filters grouped data after the aggregation.

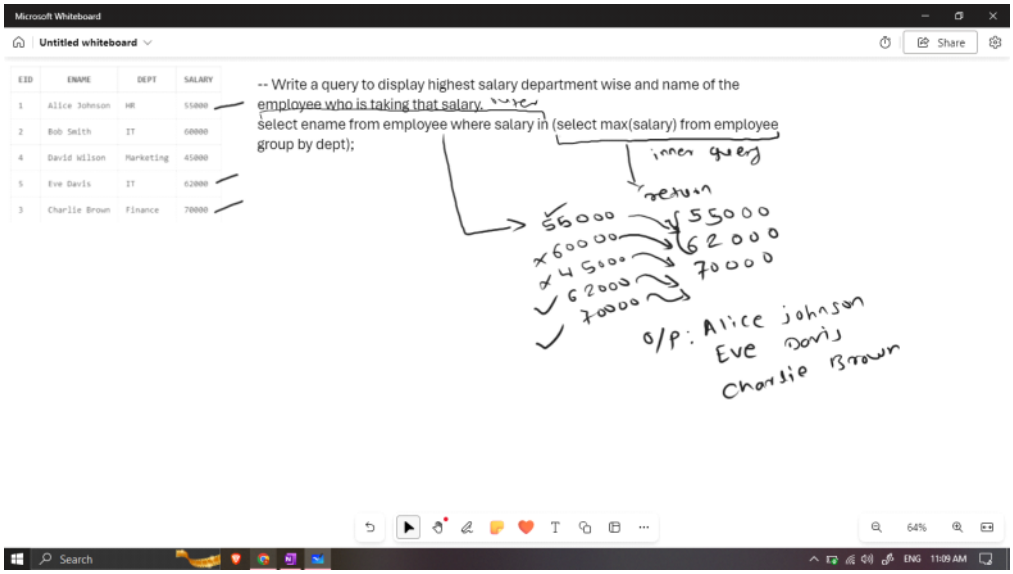
-- Write a sql query to display all the dept name along with no. of employee working in each dept.
select dept,count(*) as Working_count from employee group by dept;



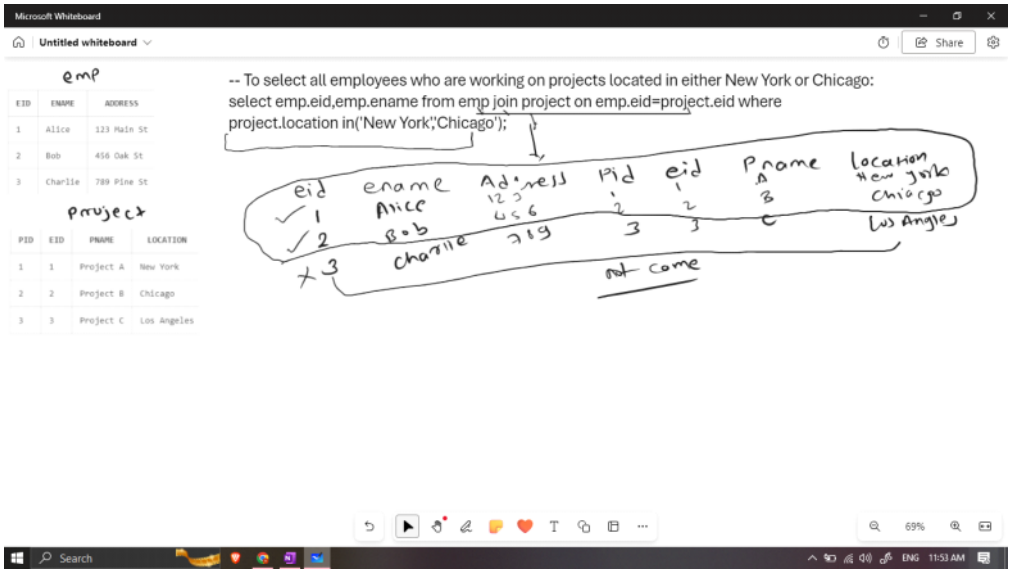
Having clause used with group by and
-- Write a sql query to display all the dept name where no. of employee are less than 2.
select dept from employee group by dept having count(*) < 2;

-- Write a query to display highest salary department wise and name of the employee who is taking that salary.

select ename from employee where salary in (select max(salary) from employee group by dept);



In clause :
-- To select all employees who are working on projects located in either New York or Chicago:
select * from emp join project on emp.eid=project.eid where project.location in('New York','Chicago');



Not in :
-- To select all employees who are NOT working on projects located in New York or Chicago:
select * from emp join project on emp.eid=project.eid where project.location not in('New York','Chicago');

Opposite to in clause

Here's a comparison between **Nested Queries**, **Correlated Queries**, and **Joins** in SQL presented in a table format:

Aspect	Nested Query	Correlated Query	Join
Definition	A query where one query is embedded inside another query.	A query where the subquery depends on the outer query.	A query that combines data from two or more tables based on a related column.
Example	SELECT * FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);	SELECT * FROM employees e WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e2.department = e.department);	SELECT employees.name, departments.name FROM employees INNER JOIN departments ON employees.dept_id = departments.dept_id;
Subquery Type	The subquery is executed once for the outer query.	The subquery is executed for each row of the outer query.	No subquery involved.
Dependency on Outer Query	Subquery does not depend on the outer query.	Subquery is dependent on the outer query.	No dependency on outer query.

Execution	The subquery executes first, and its result is used by the outer query.	The subquery is evaluated for each row processed by the outer query.	Data is combined row by row based on a condition (e.g., matching columns).
Performance	Generally more efficient when the subquery result is small or indexed.	Can be slower since the subquery is executed for every row.	Can be faster than correlated queries, especially with indexed join columns.
Use Case	Suitable for cases where you need to filter or aggregate data based on a condition that doesn't change for each row.	Suitable for cases where each row's result depends on the outer query's values.	Suitable when you need to retrieve related data from multiple tables.
SQL Structure	Subquery is enclosed in parentheses and used in the WHERE, HAVING, or SELECT clause.	Subquery is enclosed in parentheses and used in the WHERE, HAVING, or SELECT clause, but it references columns from the outer query.	The JOIN keyword is used with a condition to combine rows from multiple tables.
Example Use Case	Finding employees with a salary greater than the average salary.	Finding employees with a salary greater than the average salary in their department.	Retrieving employee names and department names from employees and departments tables.

The LIKE operator in SQL is used to search for a specified pattern in a column. It is often used in combination with the WHER E clause to filter records based on a partial match to a string.

Syntax:

```
SELECT column_name
FROM table_name
WHERE column_name LIKE pattern;
```

Wildcards Used with LIKE:

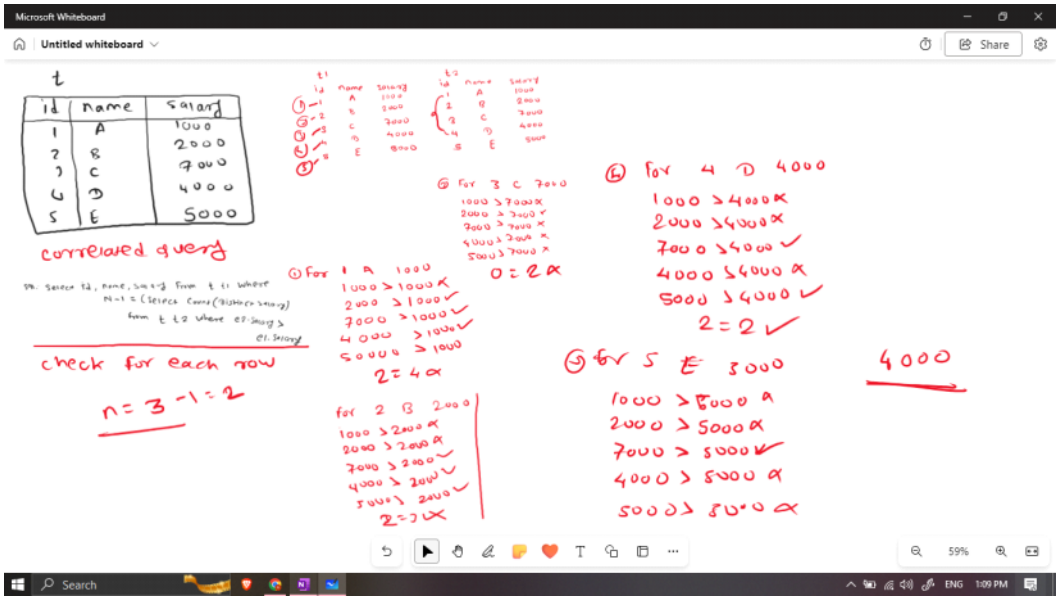
1. **%** – Represents zero or more characters.
   - Example: LIKE 'a%' will find any values that start with "a" (e.g., "apple", "abc", etc.).
2. **_** (underscore) – Represents a single character.
   - Example: LIKE '_r%' will find any values where the second character is "r" (e.g., "green", "brow", etc.).

Examples:

1. **Find names that start with 'J':**
   ```
   SELECT name
   FROM employees
   WHERE name LIKE 'J%';
   ```
2. **Find names that end with 'son':**
   ```
   SELECT name
   FROM employees
   WHERE name LIKE '%son';
   ```
3. **Find names that have 'a' in the second position:**
   ```
   SELECT name
   FROM employees
   WHERE name LIKE '_a%';
   ```
4. **Find names that contain 'a' at any position:**
   ```
   SELECT name
   FROM employees
   WHERE name LIKE '%a%';
   ```

Find nth highest salary :

```
SELECT name, salary
FROM #Employee e1
WHERE N-1 = (SELECT COUNT(DISTINCT salary) FROM #Employee e2
WHERE e2.salary > e1.salary)
```



Another ways :https://medium.com/@rganesh0203/how-many-ways-to-find-nth-highest-salary-in-mssql-f153bccfef6b