**Lab Cycle 2**

**Date : 13/03/2025**

**Experiment No: 5**

**AIM : Familiarization of Subquery ,Joins ,Views and Set Operations.**

Subqueries are nested queries used to retrieve data for the main query, often with aggregate functions. Joins combine data from multiple tables based on common keys, with types like INNER JOIN, LEFT JOIN, and RIGHT JOIN. Views are virtual tables that store pre-defined queries, improving readability and security. Set operations like UNION, INTERSECT, and EXCEPT help combine or filter results from multiple queries. Mastering these concepts enhances SQL efficiency and database management

1. Find all employees who locate in the location with the id 1700

SELECT employee\_id,first\_name,location\_id FROM employees WHERE location\_id = 1700;

A computer screen with white text

Description automatically generated

2. Find all employees who do not locate at the location 1700.

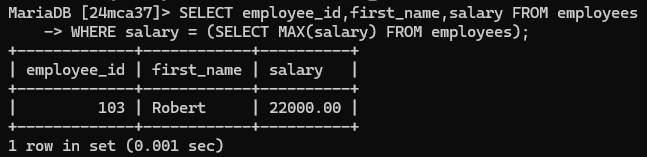
SELECT employee\_id,first\_name,location\_id FROM employees WHERE location\_id <> 1700;

A screen shot of a computer

Description automatically generated

3.Find the employees who have the highest salary.

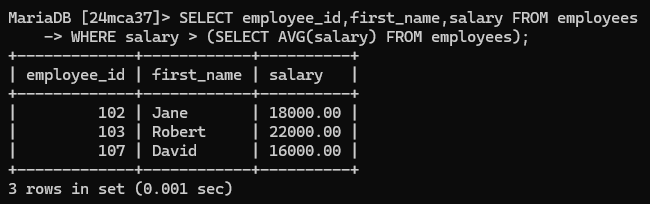
SELECT employee\_id,first\_name,salary FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);



4. Finds all employees who salaries are greater than the average salary of all employees.

SELECT employee\_id,first\_name,salary FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

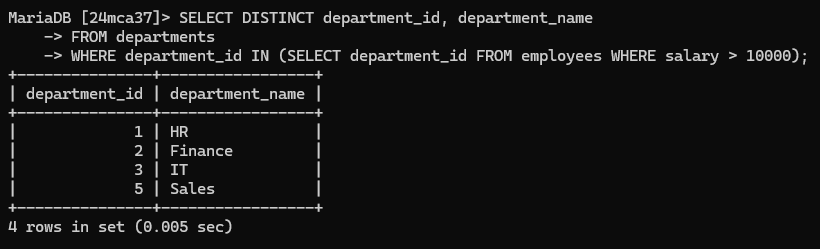


5. Finds all departments (Department Id, Name) which have at least one employee with the salary is greater than 10,000.

SELECT DISTINCT department\_id, department\_name

FROM departments

WHERE department\_id IN (SELECT department\_id FROM employees WHERE salary > 10000);

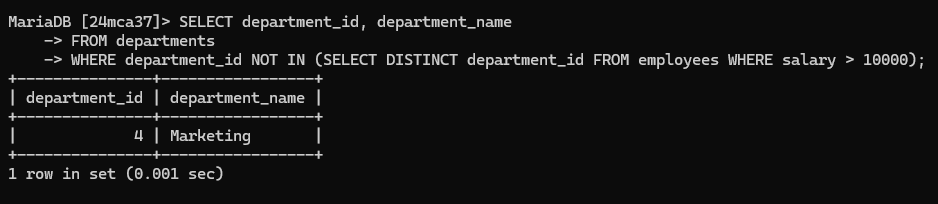


6. Finds all departments (Department Id, Name) that do not have any employee with the salary greater than 10,000.

SELECT department\_id, department\_name

FROM departments

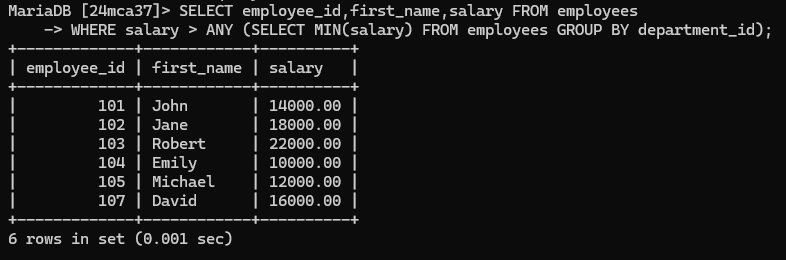
WHERE department\_id NOT IN (SELECT DISTINCT department\_id FROM employees WHERE salary > 10000);



7. Find all employees whose salaries are greater than the lowest salary of every department.

SELECT employee\_id,first\_name,salary FROM employees

WHERE salary > ANY (SELECT MIN(salary) FROM employees GROUP BY department\_id);



8. Find all employees whose salaries are greater than or equal to the highest salary of every department

SELECT employee\_id,first\_name,salary FROM employees

WHERE salary >= ALL (SELECT MAX(salary) FROM employees GROUP BY department\_id);

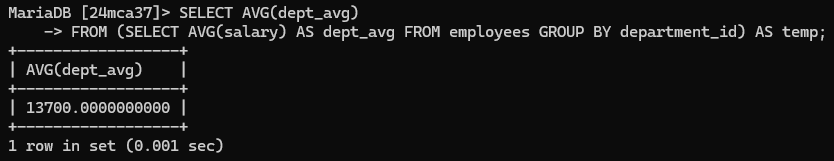
A screenshot of a computer program

Description automatically generated

9. Calculate the average of average salary of departments. (Hint: SQL subquery in the FROM clause)

SELECT AVG(dept\_avg)

FROM (SELECT AVG(salary) AS dept\_avg FROM employees GROUP BY department\_id) AS temp;

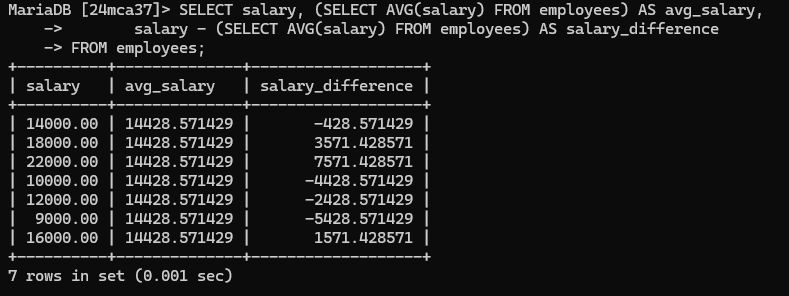


10. Finds the salaries of all employees, their average salary, and the difference between the salary of each employee and the average salary. (Hint: SQL Subquery in the SELECT clause).

SELECT salary, (SELECT AVG(salary) FROM employees) AS avg\_salary,

salary - (SELECT AVG(salary) FROM employees) AS salary\_difference

FROM employees;

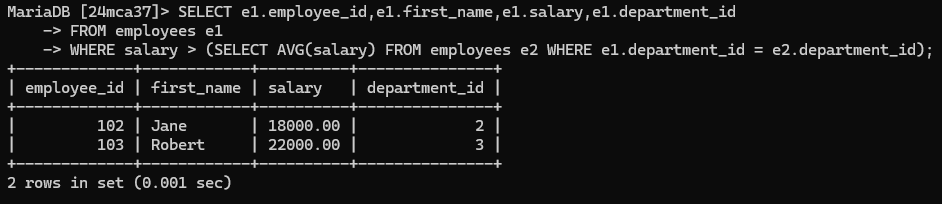


11. Finds all employees whose salary is higher than the average salary of the employees in their departments. (Hint : Use Correlated Subquery).

SELECT e1.employee\_id,e1.first\_name,e1.salary,e1.department\_id

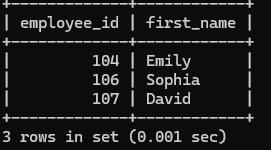
FROM employees e1

WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department\_id = e2.department\_id);



12. Find all employees who have no dependents.

SELECT employee\_id,first\_name FROM employees WHERE employee\_id NOT IN (SELECT employee\_id FROM dependents);



13. . Display first name, last name, department name of employees of the Department with id 1, 2 and 3.

SELECT first\_name, last\_name, department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

WHERE e.department\_id IN (1, 2, 3);

A black screen with white text

Description automatically generated

14. . Display the first name, last name, job title, and department name of employees who work in department with id 1, 2, and 3 and salary greater than 10000.

SELECT e.first\_name, e.last\_name, j.job\_title, d.department\_name

FROM employees e JOIN jobs j ON e.job\_id = j.job\_id

JOIN departments d ON e.department\_id = d.department\_id

WHERE e.department\_id IN (1, 2, 3) AND e.salary > 10000;

A black screen with white text

Description automatically generated

15. Display Department name, street address, postal code, country name and region name of all departments.

SELECT d.department\_name, l.street\_address, l.postal\_code, c.country\_name, r.region\_name FROM departments d

JOIN locations l ON d.location\_id = l.location\_id

JOIN countries c ON l.country\_id = c.country\_id

JOIN regions r ON c.region\_id = r.region\_id;

A screenshot of a computer

Description automatically generated

16. Write a SQL query to find out which employees have or do not have a department. Return first name, last name, department ID, department name.

SELECT e.first\_name, e.last\_name, e.department\_id, d.department\_name

FROM employees e

LEFT JOIN departments d ON e.department\_id = d.department\_id;

A black screen with white text

Description automatically generated

17. Write a SQL query to find those employees whose first name contains the letter ‘Z’. Return first name, last name, department, city, and state province.

SELECT first\_name, last\_name, department\_id, city, state\_province

FROM employees

JOIN locations ON employees.location\_id = locations.location\_id

WHERE first\_name LIKE '%Z%';

A screen shot of a computer

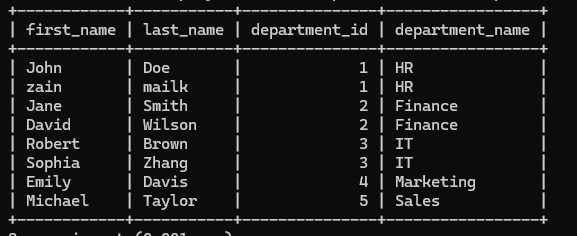
Description automatically generated

18**.** Write a SQL query to find all departments, including those without employees Return first name, last name, department ID, department name.

SELECT e.first\_name, e.last\_name, d.department\_id, d.department\_name

FROM departments d

LEFT JOIN employees e ON d.department\_id = e.department\_id;



19. Write a SQL query to find the employees and their managers. . Those managers do not work under any manager also appear in the list. Return the first name of the employee and manager.

SELECT e.first\_name AS Employee, m.first\_name AS Manager

FROM employees e

LEFT JOIN employees m ON e.manager\_id = m.employee\_id;

A screenshot of a computer screen

Description automatically generated

20. Write a SQL query to find the employees who work in the same department as the employee with the last name Taylor. Return first name, last name and department ID.

SELECT first\_name, last\_name, department\_id

FROM employees

WHERE department\_id = (SELECT department\_id FROM employees WHERE last\_name = 'Taylor');

A black screen with white text

Description automatically generated

21. Write a SQL query to calculate the difference between the maximum salary of the job and the employee's salary. Return job title, employee name, and salary difference

SELECT j.job\_title, e.first\_name, e.last\_name, (j.max\_salary - e.salary) AS salary\_difference FROM employees e JOIN jobs j ON e.job\_id = j.job\_id;

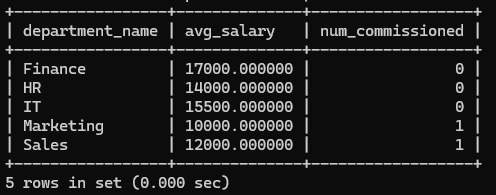


22. Write a SQL query to calculate the average salary, the number of employees receiving commissions in that department. Return department name, average salary and number of employees of all departments.

SELECT d.department\_name, AVG(e.salary) AS avg\_salary, COUNT(e.commission\_pct) AS num\_commissioned

FROM employees e JOIN departments d ON e.department\_id = d.department\_id

GROUP BY d.department\_name;



23.Create a view which contains employee name, employee id, phone number, job title, department name, manager name of employees belongs to department whose location is in ‘Delhi’ and display the details,

CREATE VIEW delhi\_employees AS

SELECT e.first\_name, e.last\_name, e.employee\_id, e.phone\_number, j.job\_title, d.department\_name, m.first\_name AS manager\_name

FROM employees e JOIN jobs j ON e.job\_id = j.job\_id

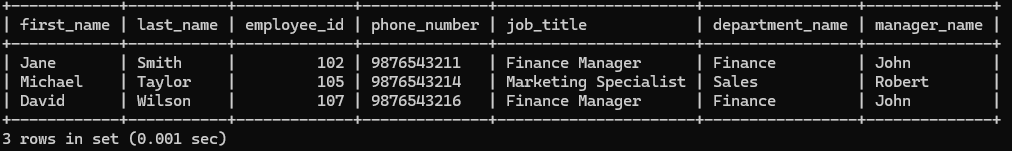
JOIN departments d ON e.department\_id = d.department\_id

JOIN locations l ON d.location\_id = l.location\_id

LEFT JOIN employees m ON e.manager\_id = m.employee\_id

WHERE l.city = 'Delhi';

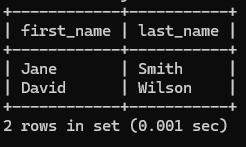
SELECT \* FROM delhi\_employees;



24.Use the above created view to obtain the names of employees whose job title is ‘Manager’ and department is ‘Finance’.

SELECT first\_name, last\_name FROM delhi\_employees

WHERE job\_title = 'Finance Manager' AND department\_name = 'Finance';



25.Check whether it is possible to update the phone number of employee whose first name is ‘Smith’ by using the above created view.

UPDATE delhi\_employees SET phone\_number = '1234567890' WHERE first\_name = 'Smith';

A screen shot of a phone number

Description automatically generated

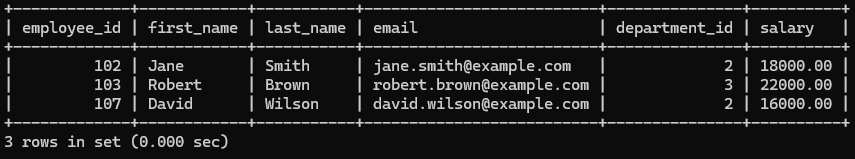
26.Display the details of employee who have no dependents.

SELECT employee\_id,first\_name,last\_name,email,department\_id,salary FROM employees WHERE employee\_id NOT IN (SELECT employee\_id FROM dependents);



27.Display the details of employee who manager id is 101 or 201. (Use Union Clause)

SELECT employee\_id,first\_name,last\_name,email,department\_id,salary FROM employees WHERE manager\_id = 101UNION SELECT employee\_id,first\_name,last\_name,email,department\_id,salary FROM employees WHERE manager\_id = 201;



28.Display the details of employees who have at least one dependent.

SELECT employee\_id,first\_name,last\_name,email,department\_id,salary FROM employees WHERE employee\_id IN (SELECT employee\_id FROM dependents);

A screen shot of a computer

Description automatically generated