SQL INTERVIEW QUESTIONS

1.Find the Second Highest Salary without using windows functions:-

```
SELECT MAX(salary) FROM employees WHERE
salary < (SELECT MAX(salary) FROM
employees);
```

Explanation:

This is a classic interview question. The inner query identifies the maximum salary, while the outer query
selects the maximum salary that is lower than that, giving the second highest.

2. Find the Nth Highest Salary:-

```
SELECT DISTINCT salary FROM employees e1
WHERE N-1 = ( SELECT COUNT(DISTINCT salary)
FROM employees e2 WHERE e2.salary >
e1.salary );
```

Explanation:

This query counts how many distinct salaries are higher than the current row’s salary. If that number equals N-1, the row corresponds to the Nth highest salary.

3. Employees Without Managers

```
SELECT * FROM employees WHERE manager_id

IS NULL;
```

Explanation:

Some employees, like CEOs or department heads, may not report to anyone. This query returns all rows where
manager_id is NULL.

4. Count Employees by Job Title

```
SELECT job_title, COUNT(*) AS total FROM
employees GROUP BY job_title;
```

Explanation:

GROUP BY clusters employees by job title, and COUNT(*) counts rows in each group. The result shows how many employees hold each role.

5. Find Duplicate Salaries

```
SELECT salary,COUNT(*) FROM employees

GROUP BY salary HAVING COUNT(*) > 1;
```

Explanation:

This groups records by salary and filters with HAVING to return only salaries that appear more than once. It’s
useful for identifying duplicates in a dataset.

6. Customers With No Orders

```
SELECT c.id, c.name FROM customers c WHERE

NOT EXISTS ( SELECT 1 FROM orders o WHERE

o.customer_id = c.id  );
```

Explanation:

The NOT EXISTS clause checks for the absence of related records. This query lists customers who have never placed an order.

7. Employees Hired in the Last 30 Days

```
SELECT * FROM employees WHERE hire_date

>= CURRENT_DATE - INTERVAL '30' DAY;
```

Explanation:

Using date arithmetic, this query filters employees whose hire dates fall within the last 30 days. It’s a practical way to track recent hires.

8. Running Total of Salaries

```
SELECT employee_id, salary,  SUM(salary) OVER
(ORDER BY employee_id) AS running_total
FROM employees;
```

Explanation:

The window function SUM() OVER creates a cumulative total of salaries ordered by employee ID. This is often used in reporting and analytics.

9. Top 3 Earners Per Department

```
SELECT * FROM ( SELECT e.*, RANK() OVER
(PARTITION BY department_id ORDER BY salary
DESC) AS rnk FROM employees e)  t WHERE rnk
<= 3;
```

Explanation:

RANK() OVER (PARTITION BY ...) assigns ranks within each department. Filtering where rnk <= 3 returns the top three earners in every department.

10. Find Customers Who Ordered This Year but Not Last Year

```
SELECT DISTINCT c.id, c.name FROM customers
c JOIN orders o ON c.id = o.customer_id WHERE
YEAR(o.order_date) = 2025 AND c.id NOT IN (
SELECT customer_id FROM orders  WHERE
YEAR(order_date) = 2024 );
```

Explanation:

This query identifies customers with orders in 2025 but excludes those who also ordered in 2024. It’s useful for tracking new or returning customers year-over-year.