# SQL Questions - Joins, Window Functions, and CTE

## Joins

Q1. Get employee names and their department names using INNER JOIN.

Tables: employees(emp_id, name, dept_id), departments(dept_id, dept_name)

Q2. List all employees and their managers' names.

Tables: employees(emp_id, name, manager_id) (Self Join)

Q3. Get the list of employees and their project names (if any). Show employees even if they're not assigned to a project.

Tables: employees(emp_id, name), projects(project_id, project_name, emp_id)

Q4. Find products and their corresponding suppliers using RIGHT JOIN.

Tables: products(prod_id, name, supp_id), suppliers(supp_id, supp_name)

Q5. Retrieve all customers and their orders. Include customers who haven't placed any orders.

Tables: customers(cust_id, name), orders(order_id, cust_id, amount)

Q6. Find employees who don't belong to any department.

Hint: Use LEFT JOIN + WHERE dept_id IS NULL

Q7. List all employees and all departments, including unmatched records from both sides.

Join: FULL OUTER JOIN (or simulate with UNION if not supported)

## Window Functions

Q8. Find the highest salary in each department using RANK() or DENSE_RANK().

Table: employees(emp_id, name, salary, dept_id)

Q9. For each employee, show their salary and the average salary in their department.

Function: AVG(salary) OVER (PARTITION BY dept_id)

Q10. Show running total of sales amounts for each customer based on order date.

Tables: orders(order_id, cust_id, amount, order_date)

Q11. Find the difference in salary between each employee and the previous one in their department.

Function: LAG(salary)

Q12. Show cumulative number of orders per day using COUNT(*) OVER (ORDER BY order_date).

Q13. For each product, find the minimum and maximum price using MIN() and MAX() window functions.

Q14. Assign a row number to each employee in their department ordered by salary.

Function: ROW_NUMBER() OVER (PARTITION BY dept_id ORDER BY salary DESC)

## CTE (Common Table Expressions)

Q15. Using a CTE, get the top 3 highest-paid employees per department.

Q16. With a recursive CTE, generate numbers from 1 to 10.

Q17. Use a CTE to find employees who earn more than the average salary of their department.

Q18. Using a CTE, find departments where the average salary is above 50,000.

Q19. With a recursive CTE, find the hierarchy of employees reporting to a given manager.

Table: employees(emp_id, name, manager_id)

Q20. Create a CTE that ranks products by sales and filters only those in the top 3 per category.

Tables: sales(prod_id, amount, category_id)