SQL Interview Preparation Queries

-- 📄 SQL Interview Preparation Queries

-- 1. Display name of Employees, Department, Country, and department manager for all employees.

SELECT

e.Name AS Employee,

d.department,

c.country,

m.Name AS Manager

FROM emi\_employee e

JOIN emi\_Dept d ON e.dept\_id = d.dept\_id

JOIN emi\_country c ON d.c\_id = c.c\_id

LEFT JOIN emi\_employee m ON e.mngr\_id = m.emp\_id;

-- 2. Display Manager of the Employee

SELECT

e.Name AS Employee,

m.Name AS Manager

FROM emi\_employee e

JOIN emi\_employee m ON m.emp\_id = e.mngr\_id;

-- 3. Total salary paid in each department

SELECT

d.Department, SUM(e.Salary) AS Total\_Salary\_Paid

FROM

emi\_employee e

RIGHT JOIN

emi\_dept d ON e.dept\_id = d.dept\_id

GROUP BY

d.Department

ORDER BY

Total\_Salary\_Paid;

-- 4. Total number of employees in each department

SELECT

d.Department, COUNT(e.emp\_id) AS No\_of\_Employees

FROM

emi\_employee e

RIGHT JOIN

emi\_dept d ON e.dept\_id = d.dept\_id

GROUP BY

d.Department

ORDER BY

No\_of\_Employees;

-- 5. Average salary per department

SELECT

d.Department, AVG(e.Salary) AS Avg\_Salary

FROM

emi\_employee e

RIGHT JOIN

emi\_dept d ON e.dept\_id = d.dept\_id

GROUP BY

d.Department;

-- 6. Min and Max salary per department with more than 2 employees

SELECT

d.department, MIN(e.Salary) AS Min\_Salary, MAX(e.Salary) AS Max\_Salary

FROM

emi\_employee e

JOIN

emi\_dept d ON e.dept\_id = d.dept\_id

GROUP BY

d.Department

HAVING

COUNT(e.emp\_id) > 2;

-- 7. Employees who are not managers

SELECT emp\_id

FROM emi\_employee

WHERE emp\_id NOT IN (SELECT DISTINCT mngr\_id FROM emi\_employee WHERE mngr\_id IS NOT NULL);

-- 8. Departments where average salary > 20000

SELECT

d.department, AVG(e.salary) AS Avg\_Salary

FROM

emi\_dept d

JOIN

emi\_employee e ON d.dept\_id = e.dept\_id

GROUP BY

d.department

HAVING

AVG(e.salary) > 20000

ORDER BY

AVG(e.salary);

-- 9. Departments with more than 1 employee earning > 25000

SELECT d.department

FROM emi\_dept d

JOIN emi\_employee e ON e.dept\_id = d.dept\_id

WHERE e.salary > 25000

GROUP BY d.department

HAVING COUNT(e.emp\_id) > 1;

-- 10. Managers managing more than 2 employees

SELECT mngr\_id AS ManagerId

FROM emi\_employee

GROUP BY mngr\_id

HAVING COUNT(emp\_id) > 2

ORDER BY ManagerId;

-- 11. Department Managers of employees who are not managers

SELECT DISTINCT dm.department\_manager

FROM emi\_employee e

JOIN emi\_dept d ON e.dept\_id = d.dept\_id

JOIN emi\_dept\_mngr dm ON d.dept\_id = dm.dept\_id

WHERE e.emp\_id NOT IN (

SELECT DISTINCT mngr\_id FROM emi\_employee WHERE mngr\_id IS NOT NULL

);

-- 12. Number of employees and average salary for each country

SELECT

c.country, COUNT(e.emp\_id) AS Num\_Employees, AVG(e.salary) AS Avg\_Salary

FROM emi\_employee e

JOIN emi\_dept d ON e.dept\_id = d.dept\_id

JOIN emi\_country c ON d.c\_id = c.c\_id

GROUP BY c.country;

-- 13. Salary analytics using RANK functions (Window Function Example)

SELECT

name,

salary,

ROW\_NUMBER() OVER (ORDER BY salary DESC) AS RowNum,

RANK() OVER (ORDER BY salary DESC) AS RankNum,

DENSE\_RANK() OVER (ORDER BY salary DESC) AS DenseRankNum

FROM emi\_employee;

-- 14. Second highest salary

SELECT MAX(salary) AS second\_highest

FROM emi\_employee

WHERE salary < (

SELECT MAX(salary) FROM emi\_employee

);

-- 15. Top 3 highest salaries

SELECT \* FROM (

SELECT name, salary, RANK() OVER (ORDER BY salary DESC) AS rnk

FROM emi\_employee

) WHERE rnk <= 3;

-- 16. Update salary of Technical department by 25%

UPDATE emi\_employee

SET salary = salary \* 1.25

WHERE dept\_id IN (

SELECT dept\_id FROM emi\_dept WHERE department = 'Technical'

);

-- 17. Employees with same salary as someone in 'd1' department

SELECT name, salary

FROM emi\_employee

WHERE salary IN (

SELECT salary FROM emi\_employee WHERE dept\_id = 'd1'

);

-- 18. DUAL queries

SELECT CEIL(12.90) AS Ceil\_Value FROM DUAL;

-- Oracle doesn’t have RAND(), use DBMS\_RANDOM.VALUE

SELECT DBMS\_RANDOM.VALUE AS Random\_Value FROM DUAL;

# 💡 Concepts

## Window Functions

Window functions perform a calculation across a set of table rows that are somehow related to the current row.  
  
Examples:  
- ROW\_NUMBER()  
- RANK()  
- DENSE\_RANK()  
- NTILE()  
- LAG() / LEAD()  
  
Syntax:  
SELECT column, ROW\_NUMBER() OVER (PARTITION BY dept\_id ORDER BY salary DESC) AS RowNum  
FROM employees;

## CTE (Common Table Expressions)

CTE provides a way to write auxiliary statements for use in a larger query.  
  
Syntax:  
WITH DepartmentSalary AS (  
 SELECT dept\_id, AVG(salary) AS avg\_salary  
 FROM employees  
 GROUP BY dept\_id  
)  
SELECT \* FROM DepartmentSalary WHERE avg\_salary > 20000;