**HANDS-ON PRACTICE QUESTIONS COVERING VARIOUS SQL CONCEPTS FOR BEGINNER AND INTERMEDIATE LEARNERS**

**Beginner Level**

1. SQL SELECT

- Write a query to display the first names of all employees from the `employee\_demographics` table.

2. SQL LIMIT

- Write a query to display the first three employees from the `employee\_salary` table.

3. SQL WHERE

- Write a query to find all employees whose age is greater than 40.

4. SQL Comparison Operators

- Write a query to find employees with a salary greater than 50,000.

5. SQL Logical Operators (AND, OR, NOT)

- Write a query to find employees who are female and older than 30.

- Write a query to find employees who are male or whose salary is less than 30,000.

- Write a query to find employees who are not office managers.

6. SQL LIKE

- Write a query to find all employees whose first name starts with the letter 'A'.

7. SQL IN

- Write a query to find employees who work in departments with IDs 1, 3, and 4.

8. SQL BETWEEN

- Write a query to find employees whose salary is between 40,000 and 70,000.

9. SQL IS NULL

- Write a query to find all employees who do not belong to any department.

10. SQL ORDER BY

- Write a query to display the employees ordered by their salary in ascending order.

11. SQL COUNT

- Write a query to count how many employees are in the `employee\_demographics` table.

12. SQL SUM

- Write a query to find the total salary paid to employees in department 1.

13. SQL MIN/MAX

- Write a query to find the highest salary in the `employee\_salary` table.

- Write a query to find the lowest age in the `employee\_demographics` table.

14. SQL AVG

- Write a query to calculate the average salary of employees.

15. SQL GROUP BY

- Write a query to group employees by gender and count how many employees are male and female.

16. SQL HAVING

- Write a query to find the departments where the total salary of employees exceeds 100,000.

17. SQL CASE

- Write a query to classify employees as 'Senior' if their age is 40 or above, and 'Junior' if their age is below 40.

18. SQL DISTINCT

- Write a query to display distinct occupations from the `employee\_salary` table.

**Intermediate Level**

19. SQL Joins

- Write a query to display the first name, last name, and department name of employees by joining `employee\_salary` and `parks\_departments`.

20. SQL INNER JOIN

- Write a query to display employees who are in departments (non-NULL department IDs) and their department names.

21. SQL Outer Joins (LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN)

- Write a query to display all employees and their department names, showing NULL where the employee does not belong to a department (LEFT JOIN).

- Write a query to display all departments and their employees, showing NULL where no employee is assigned to a department (RIGHT JOIN).

- Write a query to display all employees and departments, whether or not they are related (FULL OUTER JOIN).

22. SQL Joins Using WHERE or ON

- Write a query to display employees and their department names using the WHERE clause for the join condition.

23. SQL FULL OUTER JOIN

- Write a query to retrieve all employees and departments with an outer join, ensuring all unmatched rows from both tables are included.