**SQL Practice Questions:**

**Here is a sample schema for the tables referenced in the SQL queries below. This schema includes the employees, departments, and salaries tables, along with their attributes.**

**1. employees Table**

**This table stores information about employees.**

**CREATE TABLE employees (**

**employee\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**job\_title VARCHAR(100),**

**salary DECIMAL(10, 2),**

**hire\_date DATE,**

**department\_id INT,**

**manager\_id INT,**

**FOREIGN KEY (department\_id) REFERENCES departments(department\_id)**

**);**

**2. departments Table**

**This table stores information about the departments in the company.**

**sql**

**Copy code**

**CREATE TABLE departments (**

**department\_id INT PRIMARY KEY,**

**department\_name VARCHAR(100)**

**);**

**3. salaries Table**

**This table stores salary information for employees. You may store historical salary data or a current salary record.**

**CREATE TABLE salaries (**

**salary\_id INT PRIMARY KEY,**

**employee\_id INT,**

**salary DECIMAL(10, 2),**

**from\_date DATE,**

**to\_date DATE,**

**FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)**

**);**

**1. employees Table Data**

**INSERT INTO employees (employee\_id, first\_name, last\_name, job\_title, salary, hire\_date, department\_id, manager\_id)**

**VALUES**

**(1, 'John', 'Doe', 'Software Engineer', 60000, '2015-06-01', 1, 2),**

**(2, 'Jane', 'Smith', 'Software Engineer', 65000, '2016-07-15', 1, 2),**

**(3, 'Mike', 'Johnson', 'HR Manager', 55000, '2014-03-22', 2, NULL),**

**(4, 'Emily', 'Davis', 'Marketing Specialist', 48000, '2018-11-05', 3, 5),**

**(5, 'James', 'Brown', 'CEO', 120000, '2010-01-01', 1, NULL),**

**(6, 'Sophia', 'Williams', 'CTO', 110000, '2012-04-20', 1, 5);**

**2. departments Table Data**

**INSERT INTO departments (department\_id, department\_name)**

**VALUES**

**(1, 'Engineering'),**

**(2, 'Human Resources'),**

**(3, 'Marketing');**

**3. salaries Table Data**

**INSERT INTO salaries (salary\_id, employee\_id, salary, from\_date, to\_date)**

**VALUES**

**(1, 1, 60000, '2015-06-01', '2024-12-31'),**

**(2, 2, 65000, '2016-07-15', '2024-12-31'),**

**(3, 3, 55000, '2014-03-22', '2024-12-31'),**

**(4, 4, 48000, '2018-11-05', '2024-12-31'),**

**(5, 5, 120000, '2010-01-01', '2024-12-31'),**

**(6, 6, 110000, '2012-04-20', '2024-12-31');**

**Relationships:**

* **Employees are linked to Departments via department\_id.**
* **Employees may also have a manager\_id, which refers to another employee\_id (indicating the manager).**
* **Salaries are linked to Employees via employee\_id.**

**With this schema, you can now execute the SQL queries from the below list on this database structure.**

**Basic SQL Query Questions**

1. **Write a query to select all columns from a table called employees.**

SELECT \* FROM employees;

1. **Write a query to select the first name and last name of all employees from the employees table.**

SELECT first\_name, last\_name FROM employees;

1. **Write a query to find the total number of employees in the employees table.**

SELECT COUNT(\*) FROM employees;

1. **Write a query to select all employees who have a salary greater than 50,000.**

SELECT \* FROM employees WHERE salary > 50000;

1. **Write a query to find employees whose first name starts with the letter "A".**

SELECT \* FROM employees WHERE first\_name LIKE 'A%';

1. **Write a query to find employees whose salary is between 40,000 and 60,000.**

SELECT \* FROM employees WHERE salary BETWEEN 40000 AND 60000;

1. **Write a query to sort employees by their salary in descending order.**

SELECT \* FROM employees ORDER BY salary DESC;

1. **Write a query to find the maximum salary from the employees table.**

SELECT MAX(salary) FROM employees;

1. **Write a query to find the average salary of employees.**

SELECT AVG(salary) FROM employees;

1. **Write a query to find employees who do not have a manager (i.e., the manager\_id is NULL).**

SELECT \* FROM employees WHERE manager\_id IS NULL;

**Intermediate SQL Query Questions**

1. **Write a query to count the number of employees in each department.**

SELECT department\_id, COUNT(\*) FROM employees GROUP BY department\_id;

1. **Write a query to select the employees who have a salary greater than the average salary.**

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

1. **Write a query to select the second highest salary from the employees table.**

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

1. **Write a query to find the employees with the highest salary in each department.**

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

FROM employees

WHERE salary IN (SELECT MAX(salary) FROM employees GROUP BY department\_id);

1. **Write a query to select the employees whose first name is 'John' or 'Jane'.**

SELECT \* FROM employees WHERE first\_name IN ('John', 'Jane');

1. **Write a query to select employees whose salary is less than the average salary.**

SELECT \* FROM employees WHERE salary < (SELECT AVG(salary) FROM employees);

1. **Write a query to find the department with the highest number of employees.**

SELECT department\_id, COUNT(\*) AS employee\_count

FROM employees

GROUP BY department\_id

ORDER BY employee\_count DESC

LIMIT 1;

1. **Write a query to find the employee who has been with the company the longest.**

SELECT \* FROM employees ORDER BY hire\_date ASC LIMIT 1;

1. **Write a query to update the salary of an employee with employee\_id 101 to 55,000.**

UPDATE employees SET salary = 55000 WHERE employee\_id = 101;

1. **Write a query to delete all employees from the employees table who have a salary below 30,000.**

DELETE FROM employees WHERE salary < 30000;

**JOIN Queries**

1. **Write a query to join the employees and departments tables to display employee names and their department names.**

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

FROM employees e

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

1. **Write a query to perform an INNER JOIN between employees and departments tables and display employee names and their department names.**

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

FROM employees e

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

1. **Write a query to perform a LEFT JOIN between employees and departments and display all employees and their department names (if available).**

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

FROM employees e

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

1. **Write a query to perform a RIGHT JOIN between employees and departments and display all departments and their employees (if available).**

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

FROM employees e

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

1. **Write a query to perform a FULL OUTER JOIN between employees and departments and display all employees and departments.**

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

FROM employees e

FULL OUTER JOIN departments d ON e.department\_id = d.department\_id;

1. **Write a query to join three tables: employees, departments, and salaries, and display the employee name, department, and salary.**

SELECT e.first\_name, e.last\_name, d.department\_name, s.salary

FROM employees e

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

JOIN salaries s ON e.employee\_id = s.employee\_id;

1. **Write a query to find employees who do not belong to any department (use LEFT JOIN).**

SELECT e.first\_name, e.last\_name

FROM employees e

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

WHERE d.department\_id IS NULL;

**Group By and Aggregate Functions**

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

SELECT department\_id, SUM(salary) AS total\_salary

FROM employees

GROUP BY department\_id;

1. **Write a query to find the number of employees in each department.**

SELECT department\_id, COUNT(\*) AS num\_employees

FROM employees

GROUP BY department\_id;

1. **Write a query to find the minimum, maximum, and average salary in the employees table.**

SELECT MIN(salary) AS min\_salary, MAX(salary) AS max\_salary, AVG(salary) AS avg\_salary

FROM employees;

1. **Write a query to group employees by their department and count how many employees are in each department.**

SELECT department\_id, COUNT(\*) AS num\_employees

FROM employees

GROUP BY department\_id;

1. **Write a query to find the employee with the highest salary in each department.**

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

FROM employees e

WHERE salary IN (SELECT MAX(salary) FROM employees WHERE department\_id = e.department\_id GROUP BY department\_id);

1. **Write a query to find the departments that have more than 10 employees.**

SELECT department\_id, COUNT(\*) AS num\_employees

FROM employees

GROUP BY department\_id

HAVING COUNT(\*) > 10;

**Subqueries**

1. **Write a query to find employees who earn more than the average salary of all employees.**

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

1. **Write a query to find the department with the highest average salary.**

SELECT department\_id

FROM employees

GROUP BY department\_id

ORDER BY AVG(salary) DESC

LIMIT 1;

1. **Write a query to find employees whose salary is higher than the salary of the employee with employee\_id 100.**

SELECT \* FROM employees WHERE salary > (SELECT salary FROM employees WHERE employee\_id = 100);

1. **Write a query to find employees who do not have any subordinates (i.e., employees who are not managers).**

SELECT \* FROM employees WHERE employee\_id NOT IN (SELECT DISTINCT manager\_id FROM employees WHERE manager\_id IS NOT NULL);

1. **Write a query to find the employee with the lowest salary.**

SELECT \* FROM employees WHERE salary = (SELECT MIN(salary) FROM employees);

1. **Write a query to find employees who work in the same department as employee with employee\_id 101.**

SELECT \* FROM employees WHERE department\_id = (SELECT department\_id FROM employees WHERE employee\_id = 101);

1. **Write a query to find employees who have the same salary as the highest-paid employee.**

SELECT \* FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);

**Advanced SQL Query Questions**

1. **Write a query to find employees who have the same job title as the employee with employee\_id 101.**

SELECT \* FROM employees WHERE job\_title = (SELECT job\_title FROM employees WHERE employee\_id = 101);

1. **Write a query to find employees who were hired in the last 6 months.**

SELECT \* FROM employees WHERE hire\_date > CURRENT\_DATE - INTERVAL 6 MONTH;

1. **Write a query to find the top 3 highest-paid employees.**

SELECT \* FROM employees ORDER BY salary DESC LIMIT 3;

1. **Write a query to find employees who have been with the company for more than 5 years.**

SELECT \* FROM employees WHERE hire\_date < CURRENT\_DATE - INTERVAL 5 YEAR;

1. **Write a query to update the salary of employees who have been with the company for more than 5 years by 10%.**

UPDATE employees

SET salary = salary \* 1.1

WHERE hire\_date < CURRENT\_DATE - INTERVAL 5 YEAR;

1. **Write a query to find the average salary of employees in each department, but only for departments with more than 5 employees.**

SELECT department\_id, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department\_id

HAVING COUNT(\*) > 5;

1. **Write a query to calculate the total salary paid to employees for each year (group by hire year).**

SELECT YEAR(hire\_date) AS hire\_year, SUM(salary) AS total\_salary

FROM employees

GROUP BY YEAR(hire\_date);

1. **Write a query to find the employees who were hired in the same month as employee with employee\_id 101.**

SELECT \* FROM employees WHERE MONTH(hire\_date) = MONTH((SELECT hire\_date FROM employees WHERE employee\_id = 101));

1. **Write a query to find the employee who earns the most in each department (use window functions).**

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

RANK() OVER (PARTITION BY department\_id ORDER BY salary DESC) AS rank

FROM employees

WHERE rank = 1;

1. **Write a query to find employees who have the same job title and salary as another employee in the company.**

SELECT e1.first\_name, e1.last\_name, e1.salary, e1.job\_title

FROM employees e1

JOIN employees e2 ON e1.job\_title = e2.job\_title AND e1.salary = e2.salary

WHERE e1.employee\_id != e2.employee\_id;