**SQL Interview Questions:**

**1. What is SQL?**

**Answer:**  
SQL (Structured Query Language) is used to interact with relational databases — for tasks like querying, inserting, updating, and deleting data.

**2. What is the difference between WHERE and HAVING?**

**Answer:**

* WHERE filters rows **before** grouping.
* HAVING filters rows **after** GROUP BY.

📝 Example:

SELECT department, COUNT(\*)

FROM employees

GROUP BY department

HAVING COUNT(\*) > 5;

**3. What is the difference between INNER JOIN, LEFT JOIN, and RIGHT JOIN?**

**Answer:**

* INNER JOIN: Returns records common to both tables.
* LEFT JOIN: All from left table + matched from right.
* RIGHT JOIN: All from right table + matched from left.

**4. What is the use of GROUP BY?**

**Answer:**  
Used to group rows that have the same values into summary rows like count, avg, sum, etc.

**5. What is a primary key?**

**Answer:**  
A column or set of columns that uniquely identify each row. Cannot be NULL or duplicate.

**6. What is a foreign key?**

**Answer:**  
A column in one table that refers to the primary key in another table, establishing a relationship.

**7. Difference between DELETE, TRUNCATE, and DROP?**

| **Command** | **Deletes Rows** | **Removes Structure** | **Rollback Possible** |
| --- | --- | --- | --- |
| DELETE | ✅ | ❌ | ✅ |
| TRUNCATE | ✅ (All) | ❌ | ❌ |
| DROP | ✅ (All) | ✅ | ❌ |

**🚀 Intermediate/Advanced SQL Interview Questions**

**8. What is normalization?**

**Answer:**  
The process of organizing data to reduce redundancy. Common forms:

* 1NF: Atomic columns
* 2NF: Remove partial dependency
* 3NF: Remove transitive dependency

**9. How do you find duplicate records in a table?**

sql

CopyEdit

SELECT name, COUNT(\*)

FROM employees

GROUP BY name

HAVING COUNT(\*) > 1;

**10. Write a query to find the second highest salary.**

SELECT MAX(salary)

FROM employees

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

**11. What is a subquery and correlated subquery?**

**Answer:**

* **Subquery**: A query inside another.
* **Correlated subquery**: Inner query uses column from outer query.

SELECT name

FROM employees e1

WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department = e2.department);

**12. What is indexing?**

**Answer:**  
Indexing improves the speed of data retrieval but may slow inserts/updates. Think of it like a book index.

CREATE INDEX idx\_name ON employees(name);

**13. What are window functions?**

**Answer:**  
Functions like RANK(), ROW\_NUMBER(), and LEAD() used with OVER() clause.

SELECT name, department, RANK() OVER(PARTITION BY department ORDER BY salary DESC) as dept\_rank

FROM employees;

**14. What is the difference between UNION and UNION ALL?**

* UNION: Removes duplicates.
* UNION ALL: Keeps all rows (faster).

**15. What are stored procedures and functions?**

* **Stored Procedure**: Can perform actions (INSERT/UPDATE), doesn’t return a value.
* **Function**: Must return a value, used in SELECT statements.

**JOINS Interview Questions**

**1. What are different types of SQL joins?**

* **INNER JOIN**: Only matching rows.
* **LEFT JOIN**: All rows from left + matching from right.
* **RIGHT JOIN**: All rows from right + matching from left.
* **FULL OUTER JOIN**: All rows when there’s a match in either table.
* **SELF JOIN**: Joining table with itself.

**2. Example: INNER JOIN**

SELECT e.name, d.department\_name

FROM employees e

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

**3. LEFT JOIN with NULL check**

SELECT e.name, d.department\_name

FROM employees e

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

WHERE d.id IS NULL;

➡️ Finds employees **not assigned** to any department.

**🔍 SUBQUERIES Interview Questions**

**4. Simple Subquery**

SELECT name

FROM employees

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

**5. Correlated Subquery**

SELECT name

FROM employees e

WHERE salary > (SELECT AVG(salary) FROM employees WHERE department\_id = e.department\_id);

➡️ Compares each employee's salary with **department average**.

**6. IN vs EXISTS**

* IN: Slower on large subquery result sets.
* EXISTS: Checks for existence, usually faster.

SELECT name

FROM employees

WHERE EXISTS (SELECT 1 FROM departments WHERE employees.department\_id = departments.id);

**🌐 REAL-WORLD SQL SCENARIOS**

**7. Find Top 3 Salaries in Each Department**

SELECT name, salary, department\_id

FROM (

SELECT name, salary, department\_id,

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

FROM employees

) AS ranked

WHERE rank <= 3;

**8. Get Employee Count per Department**

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

FROM employees

GROUP BY department\_id;

**9. Find Duplicate Records**

SELECT name, COUNT(\*)

FROM employees

GROUP BY name

HAVING COUNT(\*) > 1;

**10. Get Employees Who Didn’t Submit Timesheets**

SELECT e.name

FROM employees e

LEFT JOIN timesheets t ON e.id = t.employee\_id

WHERE t.id IS NULL;