

SQL Interview Questions that have been asked Repetitively in interviews

1. What is a primary key, and why is it important?

Answer: A primary key is a unique identifier for a record in a table. It ensures that each record is distinct and prevents duplicate values. It also helps establish relationships between tables in a database.

2. What is a foreign key, and how does it help in database relationships?

Answer: A foreign key is a column (or set of columns) in one table that references the primary key in another table. It enforces referential integrity, ensuring that relationships between tables remain consistent.

3. What are the differences between SQL and NoSQL databases?

Answer: SQL databases are relational, structured, and use a fixed schema, making them ideal for complex queries. NoSQL databases are non-relational, schema-less, and optimized for unstructured or semi-structured data, offering flexibility and scalability.

4. Explain the differences between WHERE and HAVING clauses.

Answer: The WHERE clause filters rows before grouping, while the HAVING clause filters groups after aggregation. For example, WHERE filters data at the row level, and HAVING filters aggregated results like SUM or COUNT.

5. What is the purpose of the GROUP BY clause?

Answer: The GROUP BY clause groups rows with similar values into summary rows, often used with aggregate functions like SUM or COUNT to perform calculations on grouped data.

6. How do you find unique values in a column?

Answer: Use the DISTINCT keyword:

```
SELECT DISTINCT column_name
```

```
FROM table_name;
```

7. Explain different types of joins in SQL (INNER, LEFT, RIGHT, FULL OUTER).

Answer:

INNER JOIN: Returns matching records from both tables.

LEFT JOIN: Returns all records from the left table and matching records from the right table.

RIGHT JOIN: Returns all records from the right table and matching records from the left table.

FULL OUTER JOIN: Returns all records when there is a match in either table.

8. What is a self-join, and when would you use it?

Answer: A self-join is a join where a table is joined with itself. It is used to find relationships within the same table, such as employees reporting to managers.

9. How do you perform a cross join, and what kind of output does it generate?

Answer: A cross join combines every row from one table with every row from another table, generating a Cartesian product.

```
SELECT * FROM table1 CROSS JOIN table2;
```

10. Write a query to find common records between two tables using a join.

Answer: SELECT *

FROM table1

INNER JOIN table2

ON table1.common_column = table2.common_column;

11. What is a Cartesian product, and when does it occur?

Answer: A Cartesian product occurs when a cross join is performed, combining all rows from two tables without any condition.

12. Explain aggregate functions in SQL, like SUM, COUNT, AVG, MAX, and MIN.

Answer: SUM: Calculates the total.

COUNT: Counts rows.

AVG: Calculates the average.

MAX: Returns the highest value.

MIN: Returns the lowest value.

13. How do you use GROUP BY with aggregate functions? Provide an example.

Answer: SELECT department, SUM(salary)

FROM employees

GROUP BY department;

This groups employees by department and calculates the total salary for each group.

14. What is the purpose of COUNT DISTINCT?

Answer: COUNT DISTINCT returns the count of unique values in a column, eliminating duplicates.

15. How can you calculate a running total in SQL?

Use a window function:

Answer: SELECT column, SUM(value) OVER (ORDER BY column) AS running_total
FROM table;

16. What is a subquery, and how does it differ from a join?

Answer: A subquery is a query nested within another query, often used for filtering or aggregating. Joins combine data from multiple tables directly, while subqueries can operate independently within a larger query.

17. What is a correlated subquery, and how does it work?

Answer: A correlated subquery references columns from the outer query and is executed for each row of the outer query.

18. Give an example of using a subquery in a SELECT, FROM, or WHERE clause.

Answer: SELECT:

SELECT (SELECT MAX(salary) FROM employees) AS highest_salary;

FROM:

SELECT *

FROM (SELECT department,

AVG(salary) FROM employees GROUP BY department) AS avg_salary;

WHERE:

SELECT name

FROM employees

WHERE salary > (SELECT

AVG(salary) FROM employees);

19. How do you use subqueries to filter data in a WHERE clause?

Answer: Use a subquery to return a value or set of values for filtering:

SELECT *

FROM employees

WHERE department_id IN (SELECT id

FROM departments WHERE location = 'New York');

20. What is the difference between ROW_NUMBER(), RANK(), and DENSE_RANK()?

Answer: ROW_NUMBER(): Assigns a unique number to each row, starting from 1.

RANK(): Assigns a rank to rows, with gaps if there are ties.

DENSE_RANK(): Similar to RANK(), but no gaps in ranking for ties.