Data Analyst

Top Advanced SQL Interview Questions & Answers

ADVANCE SQL

"Here are some of the most commonly asked advanced SQL interview questions along with clear, concise answers. These questions are designed for data analyst, data engineer, and BI developer roles, focusing on topics like window functions, complex joins, subqueries, performance optimization, and real-world business logic."

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

Answer:

- ROW_NUMBER() gives a unique number to each row.
- RANK() assigns the same rank for ties but skips the next number(s).
- DENSE_RANK() assigns the same rank for ties and does not skip.

Example:

```
SELECT name, salary,
  RANK() OVER (ORDER BY salary DESC) AS rank,
  DENSE_RANK() OVER (ORDER BY salary DESC) AS dense_rank,
  ROW_NUMBER() OVER (ORDER BY salary DESC) AS row_num
FROM employees;
```

2. How do you get the second highest salary from a table?

Answer:

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

Or using RANK():

```
SELECT salary
FROM (
  SELECT salary, RANK() OVER (ORDER BY salary DESC) AS rnk
  FROM employees
) AS ranked
WHERE rnk = 2;
```

---

3. What is a CTE and when should you use it?

Answer:

A CTE (Common Table Expression) is a temporary result set used to simplify complex queries.

Syntax:

```
WITH cte_name AS (
  SELECT ...
)
SELECT * FROM cte_name;
```

Use it when:

- Breaking down complex queries
- Writing recursive queries
- Improving code readability

---

4. What are window functions? Give an example.

Answer:

Window functions perform calculations across a set of table rows related to the current row.

Example: Running Total

```
SELECT name, salary,
  SUM(salary) OVER (ORDER BY salary) AS running_total
FROM employees;
```

---

5. Difference between WHERE and HAVING clause?

Answer:

- WHERE filters rows before aggregation.
- HAVING filters groups after aggregation.

Example:

```
SELECT department, COUNT(*) AS emp_count
FROM employees
WHERE status = 'active'
GROUP BY department
HAVING COUNT(*) > 10;
```

6. Write a query to find duplicate records in a table.

Answer:

```
SELECT name, COUNT(*)
FROM employees
GROUP BY name
HAVING COUNT(*) > 1;
```

7. What is a correlated subquery?

Answer:

A correlated subquery depends on the outer query.

Example:

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

8. How to improve performance of SQL queries?

Answer:

- Use indexes on frequently filtered/joined columns.
- Avoid SELECT *
- Use WHERE clause to limit rows.
- Reduce use of nested subqueries.

- Use JOINs instead of correlated subqueries when possible.
- Analyze execution plan.

---

9. What is the difference between INNER JOIN, LEFT JOIN, and FULL OUTER JOIN?

Answer:

- INNER JOIN: Only matching rows from both tables.
- LEFT JOIN: All from left + matches from right.
- FULL OUTER JOIN: All rows from both, with NULLs where no match.

---

10. What is the purpose of PARTITION BY in SQL?

Answer:

PARTITION BY is used with window functions to divide data into groups (partitions), and the function is applied within each group.

Example:

```
SELECT name, department, salary,
  RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS dept_rank
FROM employees;
```

---

11. How do you pivot data in SQL?

Answer:

Using CASE WHEN or PIVOT (if supported):

Example (manual pivot):

```
SELECT
  department,
  SUM(CASE WHEN gender = 'Male' THEN 1 ELSE 0 END) AS male_count,
  SUM(CASE WHEN gender = 'Female' THEN 1 ELSE 0 END) AS female_count
FROM employees
GROUP BY department;
```

12. What is the difference between UNION and UNION ALL?

Answer:

- UNION: Combines results and removes duplicates.
- UNION ALL: Keeps duplicates and is faster.

13. Write a query to get top 3 selling products by revenue in each category.

Answer:

```
SELECT *
FROM (
  SELECT category, product, SUM(revenue) AS total_revenue,
         RANK() OVER (PARTITION BY category ORDER BY SUM(revenue) DESC) AS rnk
  FROM sales
  GROUP BY category, product
) AS ranked
WHERE rnk <= 3;
```

14. What is normalization? What are the types?

Answer:

Normalization is the process of organizing data to reduce redundancy.

Common types:

- 1NF: Atomic values
- 2NF: Remove partial dependencies
- 3NF: Remove transitive dependencies

15. How do you find gaps in a sequence in a table?

Answer:

```
SELECT curr.id + 1 AS missing_id
FROM employees curr
```

```
LEFT JOIN employees next ON curr.id + 1 = next.id
WHERE next.id IS NULL;
```

(Dipankar pal)
(dippal351@gmail.com)
(www.linkedin.com/in/dipankar-data-analyst)