Munna Pandey

8:05 AM

Here are 30 essential SQL questions to help you prepare for your next interview:

Window Functions

1. Compute the rolling average of sales for the last three months.
2. Rank employees uniquely based on their salaries in descending order.
3. Identify the earliest and latest purchase dates for each customer.
4. Find the second highest salary in each department using window functions.
5. Calculate the percentage contribution of each employee to the company's total revenue.

Common Table Expressions (CTEs)

1. Use a CTE to separate full names into first and last names.
2. Write a CTE to determine the longest streak of consecutive sales by an employee.
3. Generate a sequence of Fibonacci numbers up to a specific value using a recursive CTE.
4. Use a CTE to detect and list duplicate entries in a table.
5. Calculate total sales per category and filter out categories with sales below a specific threshold using a CTE.

Joins (Inner, Outer, Cross, Self)

1. List all customers, highlighting who placed orders and who didn't (Full Outer Join).
2. Identify employees assigned to more than one project using a self-join.
3. Match orders with customers and display unmatched orders as well (Left Join).
4. Create unique product combinations using a Cross Join while excluding identical product pairs.
5. Retrieve employees along with their direct managers using a self-join.

Subqueries

1. Find customers whose total purchase value exceeds the average order value.
2. Retrieve employees with the lowest salary in their respective departments.
3. Identify products ordered more than 10 times using a subquery.
4. List regions where the highest sales value is below a specified threshold.

Aggregate Functions

1. Compute the median salary for each department.
2. Summarize monthly sales and rank them in descending order.
3. Count the number of unique customers for each product.
4. Identify the top five regions based on total sales.
5. Calculate the average order value for every customer.

Indexing and Performance

1. Write a query to locate duplicate entries in a column with an index.
2. Evaluate the effect of a composite index on query performance.
3. Identify high-cardinality columns that could benefit from indexing.
4. Compare query execution times before and after implementing a clustered index.
5. Write a query that bypasses indexing to observe performance variations.

Best platforms to practice SQL problems daily

- Dataford - https://lnkd.in/gtqdw864
- LeetCode - https://lnkd.in/gkCpv7NA
- HackerRank - https://lnkd.in/gnFS4frz

- SQLZoo - https://sqlzoo.net/

- Mode Analytics - https://lnkd.in/gRPrQrf5

- SQL Bolt - https://sqlbolt.com/

- GeeksforGeeks - https://lnkd.in/ggYbizNB