EURON

Company wise SQL interview Questions Bank

BY TUSHAR MAHURI

Tushar Mahuri

American Express

1. **Identify the VIP Customers for American Express**:
   Find customers who have made transactions exceeding $5000 each and have done so more than once. These customers are considered 'VIP' or 'Whale' customers.
2. **Employees Earning More Than Their Managers**:
   Identify employees whose salaries exceed those of their direct managers.
3. **Calculate Average Transaction Amount per Year per Client**:
   Compute the average transaction amount for each client, segmented by year, for the years 2020 to 2024.
4. **Find Products with Sales Greater Than Their Average Sales in the Last 12 Months**:
   Identify products whose total sales in the last 12 months exceed their average monthly sales.
5. **Determine the Churn Rate for Customers Who Made Their First Purchase in the Last 6 Months**:
   Calculate the churn rate for customers who made their first purchase within the last 6 months but have not made any purchase in the last 30 days.

Amazon

1. **Average Review Ratings per Product per Month**:
   Calculate the average review rating for each product for every month. Return the month, product_id, and average star rating rounded to two decimal places.
2. **Optimizing a Slow SQL Query**:
   Discuss various methods to optimize a slow SQL query when handling massive datasets at Amazon.
3. **SQL Constraints**:
   Explain SQL constraints and provide examples of different types used to enforce data integrity in databases.
4. **Highest-Grossing Items**:
   Find the top two highest-grossing products in each category for the year 2022. Include the category, product, and total spend.
5. **Difference Between RANK() and DENSE_RANK()**:
   Explain the difference between the `RANK()` and `DENSE_RANK()` functions in SQL with examples.

SQL Specific (Amazon):

1. **Top 10 Customers Across Product Types Based on Sales Amount**:
   Write a SQL query to find the top 10 customers across product types based on sales amount.

2. **Missed Last Ship Arrival (LSA)**:
   Identify customers whose delivery date exceeds 2 days from the shipping date.

Angle One

1. Calculate the average time taken by users to make their first investment in the app.
2. Identify users who encounter errors on 3 consecutive tries.
3. Calculate the churn rate of users after 30 days of signup.
4. Find the top 3 most popular investment options.
5. Identify users whose first 5 investments exceed $10,000 in total.
6. Find the percentage of users who logged in at least 5 times in the first week after signup.
7. Detect users who have invested at least once every month in the past year.

Meesho

1. **SQL Transactions**:
   What are SQL transactions, and how do you use `BEGIN`, `COMMIT`, and `ROLLBACK`?
2. **Ensuring Data Integrity with Transactions**:
   How do you ensure data integrity using transactions?
3. **Transactions in Consecutive Months**:
   Write a query to get the customers who made transactions in two consecutive months in 2023.
4. **Cumulative Sales Department-Wise**:
   Write a query to find the cumulative sales department-wise.

Swiggy Business Analyst

1. **Top Restaurants in Metro Cities**:
   Identify the top two restaurants by the number of orders in September 2021 across four metro cities (Delhi, Bangalore, Mumbai, Chennai). Return `city_name` and `restaurant_id`.

Order_Fact Table Columns:

- ORDER_ID: Unique order identifier
- CUSTOMER_ID: Unique customer identifier
- DT: Order date

- RESTAURANT_ID: Unique restaurant identifier
- COUPON_CODE: Coupon used during the order
- GMV_TOTAL: Gross Merchandise Value of the order
- CITY: City where the order was placed

Deloitte

1. **Difference Between RANK(), DENSE_RANK(), and ROW_NUMBER()**: Explain the difference between these functions using examples.

Using Table: Employee (EmpID, ManagerID, JoinDate, Dept, Salary): 2. Find the nth highest salary from the Employee table. 3. Find all employees under a specific manager, including their subordinates at any level. 4. Find the cumulative salary of employees department-wise, who have joined the company in the last 30 days. 5. Find the top 2 customers with the highest order amount for each product category, handling ties appropriately.

Goldman Sachs

1. Calculate the average of salaries department-wise.
2. Write a SQL query to see employee name and manager name using a self-join on the `employees` table with columns `emp_id`, `name`, and `manager_id`.
3. Find the newest joinee for every department using `LEAD()` or `LAG()`.

ZS Associates

1. Identify customers who have ordered more than the average number of products across all orders.
   Tables:
   - Orders (OrderID, CustomerID, OrderDate)
   - OrderDetails (OrderID, ProductID, Quantity)

Google

1. Write an SQL query to find the total amount spent by each customer who made more than 3 orders in the system. Return `customer_id` and `total_amount`, sorted by the

total amount in descending order.
Table: Orders (order_id, customer_id, order_date, amount)

Meesho Advanced

1. How would you optimize a slow-running query with multiple joins?
2. What is a recursive CTE, and can you provide an example of when to use it?
3. Explain the difference between clustered and non-clustered indexes and when to use each.
4. Write a query to find the second highest salary in each department.
5. How would you detect and resolve deadlocks in SQL?
6. Explain window functions and provide examples of ROW_NUMBER, RANK, and DENSE_RANK.
7. Describe the ACID properties in database transactions and their significance.
8. Write a query to calculate a running total with partitions based on specific conditions.

Walmart

1. **Inventory Analysis**:
   Write an SQL query to find products that were out of stock (i.e., `quantity_in_stock = 0`) on the day after they were sold. Return `product_id`, `store_id`, and the date when they went out of stock.
   Tables:
   - Sales (sale_id, product_id, store_id, sale_date, quantity_sold)
   - Inventory (product_id, store_id, inventory_date, quantity_in_stock)

Basic SQL Concepts

1. What is SQL, and why is it important in data analytics?
2. Explain the difference between `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, and `FULL OUTER JOIN`.
3. What is the difference between `WHERE` and `HAVING` clauses?
4. How do you use `GROUP BY` and `HAVING` in a query?
5. Write a query to find duplicate records in a table.
6. How do you retrieve unique values from a table using SQL?
7. Explain the use of aggregate functions like `COUNT()`, `SUM()`, `AVG()`, `MIN()`, and `MAX()`.
8. What is the purpose of the `DISTINCT` keyword in SQL?

Tushar Mahuri

Intermediate SQL

1. Write a query to find the second-highest salary from an employee table.
2. What are subqueries, and how do you use them?
3. What is a Common Table Expression (CTE)? Provide an example of when to use it.
4. Explain window functions like `ROW_NUMBER()`, `RANK()`, and `DENSE_RANK()`.
5. How do you combine results of two queries using `UNION` and `UNION ALL`?
6. What are indexes in SQL, and how do they improve query performance?
7. Write a query to calculate the total sales for each month using `GROUP BY`.

Advanced SQL

1. How do you optimize a slow-running SQL query?
2. What are views in SQL, and when would you use them?
3. What is the difference between a stored procedure and a function in SQL?
4. Explain the difference between `TRUNCATE`, `DELETE`, and `DROP` commands.
5. What are windowing functions, and how are they used in analytics?
6. How do you use `PARTITION BY` and `ORDER BY` in window functions?
7. How do you handle `NULL` values in SQL, and which functions help with that (e.g., `COALESCE`, `ISNULL`)?

Frequently Asked Advanced Questions

1. **Top N Analysis**:
   - Find the top N products by sales.
   - Identify the top N products within each category.
   - Find the top N employees by salaries.
2. **Year-over-Year (YoY) Growth**:
   - Calculate year-over-year growth for overall sales.
   - Compute YoY growth for each product category.
   - Identify products with higher sales compared to the previous month.
3. **Running Totals and Rolling Averages**:
   - Calculate running sales over months.
   - Find rolling N-month sales for each product category.
4. **Pivoting Data**:
   - Convert rows into columns, e.g., show year-wise sales for each category in separate columns.

5. **Join Analysis**:
    - Determine the number of records resulting from different kinds of joins (`INNER`, `LEFT`, `RIGHT`, `FULL OUTER`).

Basic SQL Questions

1. Discuss the role of the `WHERE` clause in SQL queries and provide examples of its usage.
2. Explain the concept of database transactions and the ACID properties.
3. What is the purpose of the `GROUP BY` clause in SQL? Provide an example.
4. Explain the difference between an `INNER JOIN` and a `LEFT JOIN` with examples.
5. Describe the benefits of using subqueries in SQL and provide a scenario where they would be useful.
6. Discuss the differences between the `CHAR` and `VARCHAR` data types in SQL.
7. Explain the purpose of the `ORDER BY` clause in SQL queries and provide examples.
8. Describe the importance of data integrity constraints such as `NOT NULL`, `UNIQUE`, and `CHECK` constraints in SQL databases.
9. Discuss the advantages and disadvantages of using stored procedures.
10. Explain the difference between an aggregate function and a scalar function in SQL, with examples.
11. Discuss the role of the `COMMIT` and `ROLLBACK` statements in SQL transactions.
12. Explain the purpose of the `LIKE` operator in SQL and provide examples of its usage.
13. Describe the concept of normalization forms (1NF, 2NF, 3NF) and why they are important in database design.
14. Describe the benefits of using database triggers and provide examples of their usage.
15. Discuss the concept of database concurrency control and how it is achieved in SQL databases.

Intermediate SQL Questions

16. Discuss the differences between a clustered and non-clustered index in SQL.
17. Explain the concept of data warehousing and how it differs from traditional relational databases.
18. Discuss the advantages of using parameterized queries in SQL applications.
19. Write a query to retrieve all employees who have a salary greater than $100,000.
20. Explain the role of the `SELECT INTO` statement in SQL and provide examples of its usage.
21. Describe the differences between a database view and a materialized view in SQL.

22. Write a query to count the number of distinct products sold in the past week.
23. Write a query to find the top 10 customers with the highest total order amount.
24. Write a query to display the total number of orders placed in the last month.
25. Write a query to find the average order value for each customer.

Advanced SQL Questions

26. How do you write a query to find duplicate rows in a table?
27. How would you perform a left join and filter out nulls in SQL?
28. What is a window function in SQL, and how do you use it for ranking data?
29. How do you calculate the cumulative sum for a column in SQL?
30. What is the difference between `UNION` and `UNION ALL` in SQL?
31. Write a query to find the top 3 departments with the highest average salary.
32. How would you calculate a running total of orders for each customer?
33. Create a query to identify employees who have worked on all projects.
34. Write an SQL query to pivot a table without using the `PIVOT` function.
35. How would you find the median salary for each department?
36. Explain the order of execution of SQL queries.
37. Provide a use case for each of the functions: `RANK`, `DENSE_RANK`, and `ROW_NUMBER`.
38. Write a query to find the cumulative sum (running total).
39. Write a query to find the most sold product or the highest employee salary.
40. Write a query to find the 2nd/nth highest salary of employees.

Complex and Real-Time Scenarios

41. How would you identify if there are any duplicate rows in a table?
42. Explain the difference between `INNER`, `LEFT`, `RIGHT`, and `FULL OUTER JOIN`.
43. Write a query using `LAG` to find transactions where the value is greater than the previous transaction.
44. Write a query to find the 2nd highest salary, handling ties using `RANK` or `DENSE_RANK`.
45. Write a query to find the running difference using a window function.
46. Write a query to display year-over-year or month-over-month growth.
47. Write a query to find the rolling average of daily sign-ups.
48. Write a query to find the running difference using a self-join.
49. Write a query to find the cumulative sum using a self-join.
50. Write a query to calculate the median without using any median function.

Tushar Mahuri

Analytical and Business Use Cases

51. Calculate Daily Active Users (DAU), Weekly Active Users (WAU), or Monthly Active Users (MAU).
52. Write a query to calculate Week-over-Week (WoW), Month-over-Month (MoM), or Year-over-Year (YoY) percentage change.
53. Write a query to calculate churned or retained users.
54. Write a query to pivot or unpivot data.
55. Write a query to calculate what percentage of total users came from organic search.
56. Write a query using self-joins or left joins to answer specific questions.
57. Write a query to bucket data into categories like high, medium, and low ratings using `CASE` statements.
58. Write a query to find nulls, duplicates, and unique values.
59. Write a query to find the 2nd highest salary from an employee table.
60. How would you retrieve the top 5 highest-paid employees from an employee table?

Data Updates and Modifications

61. Write a query to retrieve employees who joined in the last 30 days.
62. Write a query to update the salary of all employees by 10%.
63. Write a query to delete duplicate rows from a table based on a specific column.
64. Write a query to find employees who have not been assigned to any department.
65. How can you handle `NULL` values in SQL when performing calculations?
66. How do you use the `CASE` statement in SQL? Provide an example.
67. Explain the difference between `WHERE` and `HAVING` clauses in SQL.
68. If one table has 10 rows and another has 20 rows, what is the minimum and maximum number of rows returned after joining both tables?
69. Write a query to find the 3rd highest salary of employees based on department.
70. Given a table with employee ID, employee name, and manager ID, write a query to find the manager's name and the number of reporters.

Optimization and Performance

71. How would you optimize a slow SQL query?
72. Write a SQL query to find employees who have the same name and work in the same department.
73. How do you use indexing to improve SQL query performance?
74. What is the difference between `SUBQUERY` and `JOIN`?
75. How do you use stored procedures to improve SQL query performance?

Miscellaneous and Advanced Concepts

76. Write a SQL query to find customers who have placed an order but have not made a payment.
77. Write a SQL query to find employees who work in the same department as their manager.
78. How do you use window functions to solve complex queries?
79. What is the difference between `TRUNCATE` and `DELETE`?
80. Write a SQL query to find employees who have not taken any leave in the last 6 months.
81. Explain the difference between `DROP` and `TRUNCATE`.
82. What are constraints in SQL?
83. Describe the use of the `SELECT` statement in SQL.
84. What is a primary key in SQL?
85. What is a foreign key in SQL?
86. What is a Common Table Expression (CTE) in SQL?
87. How do you implement pagination in SQL queries?
88. Describe a SQL query challenge you faced related to optimizing database performance.
89. What is a recursive stored procedure in SQL?
90. What are the subsets of SQL?

Design and Big Data Concepts

91. Explain the process and considerations for denormalizing a database.
92. Discuss the implications and solutions for dealing with `NULL` values in SQL operations.
93. How do you handle large datasets and optimize queries for big data in SQL?
94. Explain the concept of materialized views in SQL and their use cases.
95. Discuss strategies for database sharding and partitioning in SQL and their impact on performance.

Scenarios and Business Insights

96. Write a SQL query such that 4 IPL teams play with each other just once.
97. How would you calculate a cumulative sum in SQL (using a window function or join)?
98. How can you delete duplicate records from a table?
99. How do you find even and odd IDs from a table?
100. Find employees who had a higher sales performance in the current year compared to the previous year.
101. Find departments where all employees earn above the average salary.

102. List the top N products by sales for each year.
103. Identify products with sales increasing for three consecutive months.
104. Find products with the largest difference between highest and lowest monthly sales.
105. Calculate the number of days between the first and last purchase for each customer from a table (`customer_id`, `purchase_date`).