**Basic SQL Scenario Questions**

1. **Customer Data Retrieval**: You are a data analyst at an e-commerce company. You’ve been asked to extract all customer names and email addresses from the customers table for a marketing campaign. Write the SQL query to achieve this.
2. **Employee Search**: The HR department has asked you to find all employees with the job title "Manager" from the employees table for a leadership training program. Provide the SQL query.
3. **Order Filtering**: You need to generate a report of all orders placed in the last 30 days from the orders table to track current sales. Write the SQL query.
4. **Product Availability**: Your inventory team has requested a check on the availability of a product with the ID 102 in the products table. Provide a SQL query to determine if this product exists.
5. **Sales Summary**: The sales team has asked you to calculate the total sales amount from the sales table for the previous quarter. Write a SQL query to calculate this.
6. **Regional Customers**: The marketing department wants to send a targeted email to customers in New York. Write a query to retrieve all customers located in "New York" from the customers table.
7. **Top Selling Products**: Your manager wants to identify the top 5 selling products by revenue. Write a SQL query to extract this information from the sales table.
8. **Employee Count**: The HR department has requested the total number of employees in your company. Write a SQL query to count all employees in the employees table.
9. **Pending Orders**: The operations team wants to track all orders that are still pending. Write a query to list all pending orders from the orders table.
10. **Customer Search**: You have been asked to find all customers whose last name starts with "S" for a special offer. Write the query to retrieve this data.

**Intermediate SQL Scenario Questions**

1. **Average Order Value**: Your manager wants to understand the average value of orders placed in the last year. Write a SQL query to calculate the average order value from the orders table.
2. **Product Categories**: The product team has asked for a list of all unique product categories in the products table. Write a query to find this.
3. **Employee-Department Join**: You’ve been asked to produce a list of employees along with their department names. Join the employees and departments tables to create this report.
4. **Product Sales Count**: The sales department wants to know how many times each product has been sold. Write a SQL query to find the number of sales for each product from the sales table.
5. **Monthly Sales Summary**: You’ve been asked to create a report summarizing total sales for each month in the past year. Write a SQL query to group and summarize the sales from the sales table by month.
6. **Customer Order Report**: Your manager asks for a list of customers along with the total number of orders they’ve placed. Write a SQL query to generate this report.
7. **High Earning Employees**: HR wants to know which employees earn more than the average salary in the company. Write a query to find employees whose salary is above the average from the employees table.
8. **Expired Promotions**: The marketing team wants a list of all expired promotions in the promotions table. Write a SQL query to find promotions with an end date in the past.
9. **Top Customers**: You’ve been asked to identify the top 3 customers by total order amount. Write a SQL query to retrieve the customers from the orders table.
10. **Low Stock Products**: The inventory team wants to know which products have stock levels below 10. Write a SQL query to find these products from the inventory table.

**Advanced SQL Scenario Questions**

1. **Employee Reporting Hierarchy**: Your manager asks you to create a report showing each employee and their direct reports. Use a recursive query to list all employees from the employees table, along with their direct reports.
2. **Running Total**: The finance department wants to see the running total of sales over the last year. Write a SQL query to calculate a running total of sales from the sales table.
3. **Sales by Region**: Your company operates in multiple regions, and you’ve been asked to summarize total sales by region for the last quarter. Write a SQL query to achieve this.
4. **Customer Retention Rate**: The marketing team wants to know the customer retention rate for the last year. Write a SQL query to calculate how many customers have made repeat purchases.
5. **Year-over-Year Growth**: Your manager asks you to calculate the year-over-year growth in sales for the past two years. Write a SQL query to calculate this from the sales table.
6. **Employee Tenure Calculation**: HR wants a report of each employee’s tenure in years. Write a SQL query to calculate the tenure of each employee from their hire date in the employees table.
7. **Customer Spending Above Threshold**: Marketing wants a list of customers who have spent more than $10,000 in the last year. Write a SQL query using Common Table Expressions (CTE) to identify these customers.
8. **Data Comparison**: The data engineering team has created a new table\_B, and they want to know if there are any records in table\_A that do not exist in table\_B. Write a query to find the differences.
9. **Sales Pivot Table**: Your manager wants a pivot table that shows total sales per product per month. Write a SQL query to create this report.
10. **Sales Growth Analysis**: The sales department wants to identify products that had increased sales in the last quarter compared to the previous one. Write a query to perform this analysis.

**Problem-Solving SQL Scenario Questions**

1. **Detect Duplicate Customers**: The customer support team has noticed multiple accounts with the same email address. Write a query to find duplicate customer email addresses in the customers table.
2. **Remove Duplicate Records**: After identifying duplicates in the customers table, you’ve been tasked with removing them. Write a query to remove the duplicate records.
3. **Stock Level Adjustment**: The inventory team has asked you to increase the stock level of all products in the products table by 10%. Write a SQL query to perform this update.
4. **Monthly Sales Growth**: Finance wants to track the month-over-month sales growth. Write a SQL query to calculate the percentage growth in sales from one month to the next.
5. **Sales Targets**: The sales team wants to know which employees have exceeded their sales targets. Write a SQL query to compare actual sales against targets in the sales and employees tables.
6. **Customer Segmentation**: Marketing is segmenting customers into ‘high’, ‘medium’, and ‘low’ spenders based on their total purchase amounts. Write a SQL query to assign customers to these categories.
7. **Order Prioritization**: The operations team has asked for a list of orders that should be prioritized for shipment based on delivery date. Write a SQL query to retrieve these orders.
8. **Inactive Customers**: Marketing wants to target customers who haven’t made a purchase in the last year. Write a SQL query to identify these customers.
9. **Inventory Alerts**: The inventory manager wants a report that flags products with stock levels below the reorder point. Write a SQL query to generate this alert.
10. **Promotion Eligibility**: HR wants to identify employees eligible for promotion based on performance ratings. Write a query to retrieve employees meeting the eligibility criteria.

**Data Modeling and Design SQL Scenario Questions**

1. **Schema Design for Online Store**: You’ve been tasked with designing a database schema for a new online bookstore. Describe the tables and relationships you would create to store books, authors, and orders.
2. **Many-to-Many Relationship**: In your company’s database, you need to track which books were written by which authors. Write a SQL query to create a many-to-many relationship between authors and books.
3. **Referential Integrity Check**: You’ve been asked to ensure that every order in the orders table has a corresponding valid customer in the customers table. Write a SQL query to enforce referential integrity.
4. **Data Migration**: Your team is migrating from an old system. You’ve been tasked with writing a SQL query to move data from the old\_orders table to the new orders table.
5. **Archiving Old Transactions**: Your database is growing too large, and you need to archive transactions older than 5 years. Write a SQL query to archive these records from the transactions table.
6. **User Roles**: The company is implementing role-based access control. Write a SQL query to create roles and assign them to users in the database.
7. **Reporting View**: The finance team needs a view that summarizes customer sales data for easy reporting. Write a SQL query to create this view.
8. **Tracking Historical Data**: Your company wants to keep track of price changes over time. Write a SQL query to maintain historical records of product prices.
9. **Foreign Key Constraints**: You are tasked with creating foreign key constraints between the sales, customers, and products tables. Write the SQL query to set up these constraints.
10. **Schema Alterations**: The marketing team has introduced a customer loyalty program. You need to modify the customers table to include a new column for loyalty points. Write the SQL query.

**SQL Performance Optimization Scenario Questions**

1. **Slow Query Investigation**: You’ve been alerted to a slow query affecting the orders table. How would you investigate and optimize the query performance?
2. **Indexing Strategy**: Your company has a large sales table with millions of records, and queries are slow. Write a SQL query to add indexes to improve performance.
3. **Optimizing Joins**: The employees table is frequently joined with the departments table, and performance is lagging. What indexing strategy would you use to optimize join queries?
4. **Query Execution Plan**: You’ve been asked to review a slow-performing query. Write a SQL query to generate the execution plan and analyze it.
5. **Partitioning Data**: The sales table is growing rapidly, and performance is degrading. How would you use partitioning to improve query performance on this table?
6. **Caching Results**: Some queries are run repeatedly with the same parameters. How would you cache the results to improve performance?
7. **Denormalization**: To speed up reporting, your manager asks you to denormalize certain parts of the database. Describe how you would approach denormalization in SQL.
8. **Full Table Scans**: You notice a query is performing a full table scan when filtering on the order\_date column. What steps would you take to avoid the full table scan?
9. **Materialized Views**: Your manager asks if a materialized view could speed up a slow aggregate query. How would you create and maintain a materialized view in SQL?
10. **Batch Processing**: Large datasets need to be processed in batches. How would you write a SQL query to process data in smaller chunks to optimize performance?

Let me know if you’d like more advanced scenarios or specific industry-based cases!

4o

**You said:**

complete the remaining 40

**ChatGPT said:**

ChatGPT

**Advanced SQL Scenario Questions (Continued)**

1. **Database Sharding**: Your company has outgrown its single-server database setup, and the engineering team wants to implement sharding for the orders table. How would you approach database sharding, and what SQL challenges might you face?
2. **Query Parallelism**: The sales table has grown so large that your queries are taking too long to run. Describe how you could use query parallelism to speed up your SQL queries.
3. **Non-Relational Data**: You’re working with a database that stores non-relational data in JSON format in the logs table. Write a SQL query to extract specific fields from this JSON data.
4. **Database Compression**: The database storage costs are getting too high. The team wants to enable data compression on certain tables. How would you implement and test SQL-based data compression strategies?
5. **Concurrency Control**: There have been issues with simultaneous updates to the inventory table leading to data conflicts. Describe how you would manage database concurrency to ensure data integrity.
6. **Deadlock Resolution**: The engineering team reports frequent deadlocks when multiple processes attempt to update the same records in the transactions table. How would you approach resolving these deadlocks in SQL?
7. **Query Caching**: You notice a slow report query that is run multiple times throughout the day. How would you implement query caching to optimize the performance of frequently executed queries?
8. **Data Warehouse Migration**: Your company is moving from an OLTP system to a data warehouse for reporting purposes. How would you design ETL processes using SQL to migrate data efficiently?
9. **Snapshot Isolation**: The finance department needs consistent reports generated from multiple tables at the same time. Describe how you would implement snapshot isolation in your SQL queries.
10. **Index Fragmentation**: Your performance analysis reveals high index fragmentation in the products table. How would you resolve this issue and improve query performance?

**Data Analysis and Business Intelligence SQL Scenario Questions**

1. **Customer Lifetime Value**: Marketing wants to calculate the lifetime value (LTV) of customers over the last 3 years. Write a SQL query to calculate the LTV based on purchase history.
2. **AB Testing Analysis**: You’re tasked with analyzing the results of an A/B test to determine which version of a product page performs better. Write a SQL query to summarize the test results and determine statistical significance.
3. **Sales Funnel Analysis**: You’ve been asked to provide a report showing conversion rates through the sales funnel (e.g., visits, sign-ups, purchases). Write a SQL query to calculate the conversion rates at each stage.
4. **Churn Prediction**: The customer service team wants to predict customer churn based on interaction data. Write a SQL query to identify patterns of customer churn from the customer\_interactions table.
5. **Customer Segmentation for Targeting**: Marketing wants to segment customers into high-value and low-value groups based on purchase amounts. Write a SQL query to group customers and provide insights for targeting.
6. **Year-over-Year Sales Growth**: Your CEO asks you to calculate year-over-year sales growth by region for the last 5 years. Write a SQL query to generate this report from the sales and regions tables.
7. **Fraud Detection**: The finance department suspects fraudulent transactions in the payments table. Write a SQL query to flag suspicious transactions based on patterns such as large amounts or duplicate entries.
8. **Campaign Performance Analysis**: You’re tasked with analyzing the performance of a recent marketing campaign. Write a SQL query to compare the campaign's sales against a control group.
9. **Retention Analysis**: Your manager wants to analyze customer retention rates over the last 6 months. Write a SQL query to calculate the percentage of returning customers each month.
10. **Basket Analysis**: The product team wants to understand which products are frequently bought together. Write a SQL query to perform a market basket analysis on transaction data.

**Data Integrity and Security SQL Scenario Questions**

1. **Data Masking**: To comply with data privacy regulations, you’ve been asked to mask sensitive information like credit card numbers in the customers table. Write a SQL query to mask this data while still allowing analysis.
2. **User Access Levels**: You’ve been tasked with ensuring that certain users can only view specific data based on their roles. Write a SQL query to filter data based on the user’s role in the system.
3. **Audit Log**: The security team needs to track any changes made to the orders table. Write a SQL query to implement an audit trail that captures who made changes and when.
4. **Data Encryption**: You’ve been asked to store sensitive information, such as passwords, in an encrypted format in the users table. How would you implement encryption using SQL?
5. **Data Validation**: To ensure data integrity, your team wants to add validation checks when inserting new records into the customers table. Write a SQL query that adds constraints to validate email and phone number formats.
6. **Handling Missing Data**: The analytics team has noticed gaps in the sales data. How would you write a SQL query to fill missing values or handle incomplete data entries?
7. **User Permissions Auditing**: You’ve been asked to review the database for any inappropriate user access levels. Write a SQL query to audit user permissions and report any anomalies.
8. **Data Breach Investigation**: After a suspected data breach, you’ve been asked to identify all changes made to the users table in the past 24 hours. Write a SQL query to track modifications.
9. **Row-Level Security**: Your company wants to implement row-level security so that certain users can only view their own records in the sales table. Write a SQL query to enforce this restriction.
10. **Database Hardening**: You’ve been asked to ensure that all sensitive data is secured according to best practices. Write SQL queries that implement security best practices, such as restricting access and enforcing encryption.

**Real-World SQL Challenges**

1. **Multi-Tenant Application**: You are designing a database for a SaaS application that serves multiple clients. Write a SQL query to retrieve data for a specific client while ensuring data isolation for other clients.
2. **Handling Time Zones**: Your company operates globally, and you need to ensure that all timestamps are stored and retrieved accurately across time zones. Write a SQL query to convert and standardize timestamps.
3. **Transactional Data Processing**: You are building a system that processes millions of transactions daily. Write a SQL query to ensure that transactions are processed atomically and in a fail-safe manner.
4. **Real-Time Data Ingestion**: You are building a real-time analytics dashboard for the marketing team. How would you write SQL queries that continuously process and update new data as it arrives?
5. **ETL Process**: You are tasked with building an ETL (Extract, Transform, Load) pipeline to move data from the production system to the analytics data warehouse. Write SQL queries for each step of the process.
6. **Data Archival Strategy**: Your company wants to archive old customer records to save on storage costs. Write a SQL query to move records older than 5 years from the active customers table to an archive table.
7. **Handling Outliers in Data**: The analytics team has noticed outliers in the sales data. Write a SQL query to identify and remove these outliers based on statistical measures like standard deviation.
8. **Data Aggregation for Reporting**: Your finance team needs a monthly report summarizing total sales, costs, and profits. Write a SQL query to aggregate and summarize this data for reporting purposes.
9. **Customer Feedback Analysis**: Your company recently launched a new product, and you’ve been asked to analyze customer feedback. Write a SQL query to categorize feedback and provide insights into customer sentiment.
10. **Dynamic Reporting**: Your company wants to implement a system that allows users to generate custom reports based on various filters. Write SQL queries that dynamically generate reports based on user inputs (e.g., date range, product category, region).