**Introduction:**

Hello sir/ma’m,

Greeting of the day,

First of all, thank you for allowing me to introduce myself. My name is Shailesh Suthar. I am from Pali, Rajasthan. I have completed my graduation from MNIT, Jaipur. I have one and half years of full-time work experience, along with 8 months of paid internship experience.

In my current role at Saarthee, I’m working for COMCAST Smart solution team for building their products. This project is divided into three components:

* The first one includes complete end-to-end development of the Sales-Hub CRM application on Dynamic 365 helps to track their lead and conversions.
* The second component includes the development of power BI dashboard from whatever the data comes from sales-hub application on dynamic 365.
* The last one includes the implementation of campaign automation workflow using Knime and Google analytics platform.

Apart from these:

During my academic years, I completed a 6-month internship at the intelligent automation team of JP Morgan, where I have developed a daily risk report that was previously created manually, taking around five to six hours on a daily basis. For that, I designed a Python interface along with Python library which i named it as a PyautoPDF, through which anyone can automate their report in a shorter amount of time.

I also did a 2-month summer internship at Jaipur Clubfoot Pvt Ltd, where I built a machine-learning model to optimize values for minimum dimension variation in SLA printing. The same model has been published in one of the research papers for further enhancements.

I do have some of the achievement which includes:

Recently, in the review meeting I was the top performer in the whole 2023 batch, and received an employee rating of 3.5 out of 5

And during my academic year, me and my team secured first place in a national-level competition organized by SAE India, where we designed and developed an Autonomous 4 Wheeled 4-Seater Electric vehicle for Urban Mobility based on the Indian Traffic Scenario. Which took an about ten months-four stages to complete, and secure 1.5 lac of cash winning price.

In my free time, I enjoy playing chess and volleyball.

------------------------------------------------------------------------------------------------------------------------------------------

**Family:**

In my family, there are five members: my father, who working as a labor at ongoing construction in our village; my mother, who is a housewife; my elder sister, who got married; myself; and my younger sister, who is currently pursuing her final year of a BA degree.

**Reason for leaving the current firm:**

There are a couple of reasons why I want to switch to the firm:

* Once my supervisor left the company, the work environment just wasn’t the same as before. The person who replaced them started micromanaging the team not related to the work but also politics.
* While I was grateful for the opportunities at saarthee, I felt like because I wasn’t using my skills and experience to their full potential. I'm looking for an opportunity that will let me grow, so I can take my career to the next level.
* And I do want to add-up more skills/innovation to my profession which helps me to grow exponentially.

**Strength and Weaknesses:**

* Strength includes:
  + I do have a confident and collaborative nature to work as a team and completing the business objective before deadline.
  + I usually pre-plan everything on the business objective so that I can optimize the task properly to complete it within the given span of time.
  + Innovative nature to think over any problem
  + I have one law which I named it as “Star Law” which is “everything is possible in life.”
* Along with the strengths I do have some of the weaknesses which include:
  + I usually try to think deeply over the given objective which sometimes takes more time than normally. I’m trying to work upon it so that I can reduce my overthinking timing to a certain extent.

**Can you describe yourself in three to five words?**

* Punctual, passionate, innovative, loyal to work

**Can you describe a challenging data analysis project you worked on, the steps you took to tackle it, and the outcome?**

* One of the challenging data analysis projects I have ever been worked on is to develop the 'Sales-Hub' app on Dynamics 365 for COMCAST to track their lead and conversions, which required a deep understanding of the sales process and dynamics 365 within the organization. To tackle this, I started by studying Micro-Soft articles and gathering the requirements from stakeholders and understand pain points. I then designed a data model that could accurately capture and analyze the necessary information. Next, I leveraged my skills in Power Automate to automate processes for email notifications, AI Builder and agreement reports, BPF stage time calculator as so on. Additionally, I enhanced Dynamics 365 with responsive forms and JavaScript to improve data quality and usability. The outcome of the project was highly successful, as the 'Sales-Hub' app streamlined the sales process for COMCAST, leading to increased efficiency and better decision-making based on real-time data insights.

**Scenario-based Questions:**

**What are the key steps in the demand planning process?**

* The demand planning process typically involves several key steps to forecast demand accurately and align supply with customer needs, such as:
  + **Data Collection:** Gather historical data from sales, inventory, market trends
  + **Demand Forecasting:** Use statistical methods, machine learning models
  + **Adjustments and Assumptions**: Incorporate market intelligence, customer insights, and potential demand shifts.
  + **Inventory Planning**: Determine the optimal inventory levels based on forecasted demand to prevent stockouts or overstocking.
  + **Supply and Capacity Planning**: Align supply capabilities with forecasted demand, adjusting production schedules and resource allocation to meet expected requirements.
  + **Review and Approval**: Present the demand plan to stakeholders for review, incorporating feedback to finalize the forecast.
  + **Continuous Monitoring and Reforecasting**: Track actual demand against the forecast, regularly updating the plan to account for new data and changes in the market.

**How do you handle situations where demand forecasting accuracy is low?**

* **Identify Causes**: Analyze past inaccuracies (e.g., outdated data, market shifts).
* **Improve Data Quality**: Use real-time and external data.
* **Apply Multiple Methods**: Combine statistical models and qualitative insights.
* **Segment Forecasts**: Create separate forecasts for different products or regions.
* **Shorten Forecast Horizon**: Focus on shorter time frames for volatile demand.
* **Increase Flexibility**: Add safety stock, flexible suppliers, and agile production.
* **Use Scenario Planning**: Prepare for best, worst, and likely demand cases.
* **Invest in Technology**: Use AI-driven tools to improve forecasts.
* **Continuous Adjustment**: Review and refine forecasts regularly.

**Which forecasting methods and tools are you familiar with? How would you choose between them?**

* **Forecasting Methods**
  + Time Series Analysis (e.g., Moving Average, Exponential Smoothing): Useful for products with consistent historical data and stable trends.
  + Machine Learning Models (e.g., Random Forest, Neural Networks): Beneficial for complex demand patterns with multiple influencing factors.
  + Qualitative Forecasting (e.g., Delphi Method, Market Surveys): Best for new products or highly uncertain markets without much historical data.
* **Forecasting Tools**
  + Power BI, Excel, Tableau, Python,
* **Choosing b/w Methods and Tools**
  + Scalability
  + Business Needs
  + Forecast Horizon
  + Demand Pattern Complexity
  + Data Availability and Quality

**How would you approach developing a demand forecast for a new product or location?**

* **Choosing b/w Methods and Tools**
  + Market Research and Analysis
  + Analog Forecasting
  + Collaborating with Cross-Functional Teams
  + Using Qualitative Methods
  + Scenario Planning
  + Leverage Test Markets or Pre-Launch Data
  + Monitor and Adjust

**How do you handle outliers in historical data when creating a forecast?**

* **Identify Outliers**: Use statistical methods, such as z-scores or interquartile range (IQR), to detect outliers that significantly deviate from typical demand patterns.
* **Analyze the Cause**: Investigate the root cause of each outlier. Outliers due to one-off events (e.g., promotions, stockouts, seasonal spikes) may need adjustment, while outliers due to sustained changes (e.g., new product introduction) could indicate evolving demand trends.
* **Exclude or Adjust**: For temporary, non-recurring events, either exclude or adjust the outlier values to avoid skewing the forecast. For instance, use a moving average to smooth the impact of extreme values.
* **Use Robust Forecasting Methods**: Choose models less sensitive to outliers, such as median-based forecasts or exponential smoothing, which can handle irregularities without overfitting to outlier data.

**Describe your experience with inventory optimization in a high-demand, fast-paced environment.**

* **Data-Driven Decision Making**: By analyzing historical sales data, I identified trends and seasonality, enabling proactive inventory adjustments.
* **Implementing Just-In-Time (JIT)**: This approach allowed for rapid replenishment based on actual sales data.
* **Safety Stock Calculations**: I carefully calculated safety stock levels based on lead times and variability in demand. This ensured that we had enough buffer stock to prevent stockouts during peak demand periods.
* **Regular Inventory Reviews**: Conducting weekly inventory reviews helped identify slow-moving or obsolete items.
* **Scenario Planning**: scenario planning to prepare for unexpected spikes in demand, such as during holiday seasons or product launches. This helped us develop contingency plans for rapid inventory adjustments.
* **Continuous Improvement**: I actively sought feedback from team members and analyzed performance metrics to identify areas for improvement.

**What KPIs are most critical for demand planning and why?**

* Forecast Accuracy:
  + Measures how closely actual demand matches forecasts. High accuracy leads to better inventory management and reduced costs.
* Customer Service Level:
  + Indicates the percentage of customer demand met without stockouts. High service levels enhance customer satisfaction and loyalty.
* Stockout Rate:
  + Measures the frequency of stockouts. A low stockout rate indicates effective inventory management and demand fulfillment.
* Inventory Turnover Ratio:
  + Reflects how quickly inventory is sold and replaced. High turnover indicates efficient inventory management, while low turnover can suggest overstocking or poor demand forecasting.
* Lead Time:
  + Measures the time taken from placing an order to receiving it. Short lead times improve responsiveness to demand changes and reduce the risk of stockouts.
* Safety Stock Level:
  + Indicates the amount of extra inventory held to prevent stockouts. Optimizing safety stock levels can balance the risk of stockouts with holding costs.
* Backorder Rate:
  + Measures the percentage of orders that cannot be fulfilled at the time of purchase. A high backorder rate signals issues with demand planning or inventory management.
* Order Fulfillment Cycle Time:
  + Measures the time taken to process and deliver customer orders. Shorter cycle times improve customer satisfaction and indicate efficient operations.
* Sales Growth:
  + Monitors increases in sales over time, providing insights into demand trends and the effectiveness of demand planning strategies.

**Describe how you track and report demand planning performance to stakeholders.**

* Establish KPIs
* Utilize Reporting Tools
* Regular Reporting Schedule
* Contextualize Data
* Collaborative Review Meetings
* Actionable Insights
* Track Progress Over Time
* Tailored Reporting

**How would you handle a sudden and unexpected drop in demand for a key product?**

* Analyze the Cause
* Communicate with Stakeholders
* Adjust Forecasts
* Optimize Inventory Levels
* Implement Cost-Control Measures
* Review Marketing and Sales Strategies
* Monitor Performance
* Develop Contingency Plans

**How do you prioritize tasks and make decisions when handling multiple projects?**

* Assess Project Urgency and Importance
* Set Clear Goals
* Create a Task List
* Evaluate Resources and Constraints
* Prioritize Based on Impact
* Utilize Project Management Tools
* Regularly Review and Adjust Priorities
* Communicate Effectively
* Delegate When Possible

------------------------------------------------------------------------------------------------------------------------------------------

**SQL Questions:**

* What is SQL?
* What is a database?
* Differentiation between Power BI and Tableau
* Does SQL support programming language features?
* What is the difference between CHAR and VARCHAR2 datatype in SQL?
* What do you mean by data definition language?
* What do you mean by data manipulation language?
  + INSERT; UPDATE; DELETE
* What is the view in SQL?
* What do you mean by foreign key?
* What are table and Field?
* What is the primary key?
* What is a Default constraint?
* What is normalization?
  + Normalization in SQL is the process of organizing data within a database to reduce redundancy and improve data integrity. It involves dividing large tables into smaller, related tables and defining relationships between them to ensure efficient data storage and retrieval. This process helps minimize data anomalies and promotes consistency by following a set of rules called "normal forms."
  + The common normal forms include:
    - Ensuring each column contains atomic (indivisible) values, with each row unique.
    - Ensure that all non-key columns are fully dependent on the primary key.
  + Normalization aims to:
    - Eliminate redundant data.
    - Enhance data consistency and integrity.
    - Optimize the efficiency of database storage.
* What is Denormalization?
  + Denormalization is the process of intentionally adding redundancy to a database to improve read performance by reducing the need for complex joints and queries across multiple tables. It essentially reverses some aspects of normalization by combining tables, which can speed up data retrieval in applications with heavy reading requirements, such as reporting systems or analytics databases.
  + Key Points of Denormalization:
    - Improves Query Performance: It reduces the number of joins needed to retrieve data, which can be beneficial in read-heavy environments.
    - Increases Redundancy: Denormalization often duplicates data across tables, which can lead to storage inefficiencies and increase the risk of data anomalies.
    - Use Cases: Denormalization is typically used in OLAP (Online Analytical Processing) systems and for building data warehouses, where faster data retrieval outweighs the need for strict data integrity.
    - Example: Storing customer and order information in a single table instead of separate tables to avoid joining during retrieval.
  + Trade-offs
    - While denormalization can enhance performance, it also increases the need for data maintenance and can lead to data inconsistency if updates are not managed carefully.
* What is a query?
* What is a subquery?
* What are the different operators available in SQL?
  + Arithmetic Operators
  + Logical Operators
  + Comparison Operators
* What is a Constraint?
* What is MySQL collation?
  + MySQL collation is a set of rules that determines how string comparisons are made within the database. It defines the character set and the sorting rules (including case sensitivity) for storing and retrieving text. Collations are used to manage the sorting and comparison behavior of textual data based on specific languages, alphabets, and cultural norms.
  + Key Concepts of MySQL Collation:
    - Character Set: A character set defines the available characters in a collation. For example, UTF-8 or Latin1.
    - Sorting and Comparison Rules: Collation specifies how characters are ordered and compared. For instance, it can dictate whether comparisons are case-sensitive or case-insensitive.
    - Collation Names: A collation name usually follows the format: charset\_collation\_type. For example, utf8\_general\_ci (UTF-8 character set with general collation and case insensitivity).
    - Case Sensitivity:
    - Case-Insensitive (ci): ci in a collation name indicates case-insensitive comparisons (e.g., utf8\_general\_ci).
    - Case-Sensitive (cs): cs in a collation name indicates case-sensitive comparisons (e.g., utf8\_bin).
    - Default Collation: Each MySQL character set has a default collation that can be overridden at the table or column level.
  + Example:
    - If a table is set to use utf8\_general\_ci, a WHERE clause comparing "ABC" and "abc" would consider them equal. If set to utf8\_bin, the same comparison would be case-sensitive, and "ABC" and "abc" would be treated as distinct.
  + Changing Collation:
    - Collation can be changed for a database, table, or column, which affects how MySQL treats and sorts text data.
* What is user-defined functions (UDF)?
* What are all types of user-defined functions?
  + User-defined functions (UDFs) in SQL are custom functions created by users to perform specific operations that may not be covered by SQL’s built-in functions. They allow for modular, reusable code blocks that can be called within SQL queries to perform calculations, transformations, or other operations. UDFs can return a single value or a table, depending on the function type.
  + Types of User-Defined Functions
    - Scalar Functions: Return a single value, such as a numeric, text, or date value. These are used in calculations and transformations.
      * Example: A function to calculate the age based on a birth date.
    - Table-Valued Functions: Return a table result set, which can be queried like a regular table. Useful for generating a subset of data.
      * Example: A function that returns a list of employees within a specific department.
    - Inline Table-Valued Functions: Like table-valued functions, but they contain a single SELECT statement and don’t include BEGIN/END blocks. These are efficient and often faster than multi-statement functions.
      * Example: A function that retrieves customer data based on specific criteria.
  + Benefits of User-Defined Functions
    - Modularity: Encapsulate logic in one place, promoting code reusability.
    - Simplified Queries: Abstract complex operations, making queries easier to read and maintain.
    - Error Reduction: Reduces code duplication, which can help prevent errors in logic.
  + Example of a Scalar UDF
    - Here's a simple user-defined function to calculate the square of a number:
    - CREATE FUNCTION dbo.Square (@Number INT)
    - RETURNS INT
    - AS
    - BEGIN
    - RETURN @Number \* @Number;
    - END;
* What is a stored procedure?
  + A stored procedure in SQL is a precompiled set of one or more SQL statements that can be stored in the database and executed as a single unit. Stored procedures are used to perform complex operations like data manipulation, business logic, or administrative tasks on the database in a structured and reusable way. They help improve performance, reduce redundancy, and enhance security.
  + Key Benefits of Stored Procedures
    - Performance Improvement: Since stored procedures are precompiled, they execute faster than separate SQL queries.
    - Reusability: Once created, stored procedures can be called multiple times, which reduces code duplication.
    - Security: Permissions can be set to allow specific users to execute the procedure without giving them direct access to underlying tables.
    - Reduced Network Traffic: Sending a stored procedure call from an application is generally more efficient than sending multiple SQL queries.
  + Syntax for Creating a Stored Procedure
    - Here’s a basic structure for creating a stored procedure:
      * CREATE PROCEDURE GetEmployeesByDepartment
      * @DepartmentID INT
      * AS
      * BEGIN
      * SELECT EmployeeID, EmployeeName, JobTitle
      * FROM Employees
      * WHERE DepartmentID = @DepartmentID;
      * END;
      * EXEC GetEmployeesByDepartment @DepartmentID = 1;
* What are aggregate and scalar functions?
  + Aggregate Functions
    - Aggregate functions operate on a set of values and return a single summary value. They are commonly used with the GROUP BY clause to perform calculations on groups of data.
    - SUM(): COUNT(): AVG(): MAX(): MIN():
  + Scalar Functions
    - Scalar functions perform operations on individual values and return a single value for each input. They are often used to modify data or perform calculations at the row level.
    - UCASE() / UPPER(): LCASE() / LOWER(): LEN(): ROUND(): NOW(): CONCAT():
  + Key Differences Between Aggregate and Scalar Functions
    - Aggregate functions work on a group of values and return a single result, while scalar functions operate on a single value and return a single value.
    - Aggregate functions are often used with GROUP BY, whereas scalar functions are usually applied directly in SELECT queries or in conditions.
* What is an ALIAS command?
* What are Union, minus, and Interact commands?
* What is a T-SQL?
  + T-SQL, or Transact-SQL, is an extension of SQL (Structured Query Language) used primarily with Microsoft SQL Server. It adds programming constructs like variables, control-of-flow statements, and error handling to standard SQL, enabling more complex database operations and management tasks. Here are some key features of T-SQL:
    - **Procedural Programming Capabilities**: T-SQL includes functions such as IF...ELSE, WHILE, and error handling through TRY...CATCH, enabling procedural logic.
    - **Variables and Functions**: It allows the use of variables and offers built-in functions for string manipulation, date handling, and mathematical calculations.
    - **Transactions**: T-SQL supports explicit transactions, allowing users to commit or roll back changes as a unit of work.
    - **Advanced Querying**: With T-SQL, you can perform complex queries using window functions (ROW\_NUMBER(), RANK()), common table expressions (CTEs), and recursive queries.
    - **Triggers and Stored Procedures**: It supports triggers for automated responses to data changes and stored procedures for reusable code, optimizing performance and ensuring consistency.
* What is ETL in SQL?
  + **ETL is a process in Data Warehousing, and it stands for Extract, Transform, and Load. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system. These are three database functions that are incorporated into one tool to pull data out from one database and put data into another database.**
* How to copy tables in SQL?
  + CREATE TABLE new\_table AS
  + SELECT \* FROM original\_table;
* What is SQL injection?
  + **SQL Injection is a code injection technique where an attacker exploits vulnerabilities in an application’s SQL query handling to manipulate or access the database in unintended ways. By inserting or "injecting" malicious SQL code into a query, attackers can bypass authentication, retrieve sensitive data, modify or delete records, and even gain administrative control over the database.**
  + **How SQL Injection Works**
    - **SQL Injection typically occurs when user inputs (like login credentials or search terms) are not properly validated or sanitized. If a SQL query directly incorporates user inputs without validation, attackers can alter the query’s structure and influence database operations.**
  + **Example of SQL Injection**
    - SELECT \* FROM Users WHERE username = 'input\_username' AND password = 'input\_password';
    - SELECT \* FROM Users WHERE username = 'admin' -- ' AND password = 'anything';
    - **The -- starts a comment, which ignores the password condition, allowing access without a valid password.**
  + **Types of SQL Injection**
    - **Classic SQL Injection: Manipulating queries to gain unauthorized access.**
    - **Blind SQL Injection: Exploiting errors to retrieve data without direct feedback.**
    - **Union-Based Injection: Using UNION to retrieve data from other tables.**
    - **Error-Based Injection: Leveraging database error messages to gain information.**
  + **Preventing SQL Injection**
    - **Parameterized Queries/Prepared Statements: Ensure that SQL queries treat user inputs as data rather than executable code.**
    - **Input Validation and Sanitization: Verify and sanitize all user inputs to prevent malicious characters.**
    - **Use of ORM (Object-Relational Mapping): Helps generate safe SQL code automatically.**
    - **Limit Database Permissions: Restrict user privileges to minimize potential damage.**
    - **Regular Security Audits: Review code and conduct penetration testing to catch vulnerabilities.**
  + **SQL Injection can cause serious data breaches, so securing input handling is essential in web application development.**
* Can we disable a trigger? If yes, how?
* What are the differences between SQL and PL/SQL?

|  |  |
| --- | --- |
| SQL | PL/SQL (Procedural Language extensions to the Structured Query Language) |
| **We can embed SQL in PL/SQL** | **We cannot embed PL/SQL in SQL** |
| **SQL tells databases, what to do?** | **PL/SQL tells databases how to do.** |
| **We can execute one statement at a time in SQL** | **We can execute blocks of statements in PL/SQL** |
| **It is used for manipulating data.** | **It is used for creating applications.** |
| **SQL is very declarative in nature.** | **PL/SQL has a procedural nature.** |
| **SQL is a data-oriented language.** | **PL/SQL is a procedural language** |
| **SQL is a query execution or commanding language** | **PL/SQL is a complete programming language** |

* What is the difference between BETWEEN and IN operators in SQL?
* Write an SQL query to find the names of employees starting with 'A'
* Difference between Primary Key and Unique Constraints
* What is a Join in SQL? What are the Types of Joins?
* What is an Index?
* What is the On Delete Cascade Constraint?
* Explain WITH Clause in SQL.
* Different Attributes of Indexes
* What is a Cursor?
  + **A cursor in SQL is a database object that allows for the retrieval and manipulation of a result set row by row. It provides a way to iterate through a set of records returned by a query, enabling developers to process individual rows in a controlled manner.**
* Various Types of Relationships in SQL
  + **One-to-One Relationship.**
  + **One to Many Relationships.**
  + **Many to One Relationship.**
  + **Self-Referencing Relationship.**
* What is a Trigger?
* Difference between SQL DELETE and SQL TRUNCATE Commands
* Difference between Cluster and Non-Cluster Index
* What is a Live Lock?
* What is Case WHEN in SQL?
* Name different types of case manipulation functions available in SQL.
* What are local and global variables and their differences?
* Name the function which is used to remove spaces at the end of a string?
* What is the difference between TRUNCATE and DROP statements?
* Which operator is used in queries for matching patterns?
* Define SQL Order by the statement?
* Explain SQL Having statement?
* Explain SQL AND OR statement with an example?
* Define BETWEEN statements in SQL?
* Why do we use Commit and Rollback commands?
* What are ACID properties?
  + **ACID properties are a set of principles that guarantee reliable processing of database transactions. ACID stands for Atomicity, Consistency, Isolation, and Durability. These properties ensure that database transactions are processed reliably and help maintain the integrity of the database.**
    - **Atomicity**: Ensures that all parts of a transaction succeed or none do.
    - **Consistency**: Guarantees that a transaction takes the database from one valid state to another.
    - **Isolation**: Keeps transactions independent from one another, avoiding interference.
    - **Durability**: Guarantees that committed transactions are permanently recorded.
  + **Importance of ACID Properties**
    - **Maintaining data integrity in applications.**
    - **Ensuring reliability in multi-user environments.**
    - **Supporting complex transactions where consistency and correctness are paramount.**
* Are NULL values the same as zero or a blank space?
* What is the need for group functions in SQL?
* What is the need for a MERGE statement?
  + **The MERGE statement in SQL, also known as an "upsert," is used to perform conditional INSERT, UPDATE, or DELETE operations on a target table based on the results of a join with a source table. It allows for efficient handling of situations where you need to synchronize data between two tables, typically for scenarios involving data integration, ETL processes, or maintaining a data warehouse.**
  + **Key Benefits of the MERGE Statement:**
    - **Combines Multiple Operations: MERGE allows you to execute INSERT, UPDATE, and DELETE operations in a single statement, reducing the need for multiple SQL queries.**
    - **Improves Performance: By consolidating operations, MERGE can enhance performance, especially when dealing with large datasets, since it reduces the overhead associated with executing multiple queries.**
    - **Simplicity and Readability: The MERGE statement simplifies the logic required to perform complex operations, making the SQL code easier to read and maintain.**
    - **Data Synchronization: It's useful for synchronizing data between tables, ensuring that the target table reflects the latest state of the source table.**
    - **Conditional Logic: MERGE allows for complex conditional logic, enabling you to specify different actions based on the existence or absence of matching rows.**
      * MERGE INTO Customers AS c
      * USING NewCustomers AS nc
      * ON c.CustomerID = nc.CustomerID
      * WHEN MATCHED THEN
      * UPDATE SET c.CustomerName = nc.CustomerName
      * WHEN NOT MATCHED THEN
      * INSERT (CustomerID, CustomerName) VALUES (nc.CustomerID, nc.CustomerName)
      * WHEN NOT MATCHED BY SOURCE THEN
      * DELETE;
* How can you fetch common records from two tables?
* What are the advantages of PL/SQL functions?
* What is the SQL query to display the current date?
* What are Nested Triggers?
* How to find the available constraint information in the table?
* How do we avoid getting duplicate entries in a query without using the distinct keyword?
* The difference between NVL and NVL2 functions?
* What is the difference between COALESCE() & ISNULL()?
* Name the operator which is used in the query for appending two strings?
* Common SQL commands:
  + **DDL: CREATE, ALTER TABLE, DROP, TRUNCATE, and ADD COLUMN**
  + **DML: UPDATE, DELETE, and INSERT**
  + **DCL: GRANT and REVOKE**
  + **TCL: COMMIT, SET TRANSACTION, ROLLBACK, and SAVEPOINT**
  + **DQL: SELECT**

Link:

* [Top 70+ SQL Interview Questions and Answers for 2024](https://www.geeksforgeeks.org/sql-interview-questions/)

**Power BI Questions:**

* What is Power Bi?
* Features of Power BI.
* Differentiation between Power BI and Tableau
* Explain the difference between Power BI Desktop and Power BI Service.
* How do you import data into Power BI?
* What is the purpose of the Query Editor in Power BI?
* What is the role of visuals in a Power BI report?
* What is a filter in Power BI, and why is it useful?
* Major components of Power BI.
* Common chart in Power BI.
* DAX and benefits from using variables in DAX.
* Three fundamental concepts of DAX.
  + Syntax
  + Function
  + Context
* Most common DAX functions.
  + Aggregation Functions: SUM, MIN, MAX, AVG, COUNTROWS, DISTINCTCOUNT
  + Information Functions: ISBLANK, ISFILTERED, ISCROSSFILTERED
  + Statistical Functions: GEOMEAN, MEDIAN
  + Logical Functions: IF, AND, OR, SWITCH
  + Date & Time Functions: DATEDIFF, DATEVALUE
  + Filter Functions: VALUES, ALL, FILTER, CALCULATE, TOPN
  + Other Functions: UNION, INTERSECT, EXCEPT, NATURALINNERJOIN, NATURALLEFTEROUTERJOIN,
  + SUMMARIZECOLUMNS, ISEMPTY, VAR
* Calculate function in DAX
* Custom visualization.
* Different connectivity modes in power bi.
* Data source that we can connect to power bi.
* Views in power bi.
* Type of filters.
* Can you explain the basic steps for data cleaning and transformation in Power Query?
  + Data load > removed unwanted columns and rows > filter rows > rename the column and tables > correct the data type > handle missing data > standardize data format > merge and append the data > apply transformation (create calculated columns, pivot/unpivot data, split columns or performing some aggregation) > remove duplicates > group the data > sort it > and final review and then load it.
* What is a slicer in Power BI, and how is it used for data filtering?
* What are the benefits of using Power BI for data visualization and analysis?
  + Easy Integration with Data Sources
  + Interactive Visualizations
  + User-Friendly Interface
  + Powerful Data Modeling Capabilities
  + Advanced Analytics with DAX
  + Real-Time Dashboards and Monitoring
  + Collaboration and Sharing
  + Mobile Accessibility
  + Security and Compliance
  + Scalability for Large Data Volumes
  + AI and Machine Learning Integration
  + Cost-Effectiveness
* Explain various types of users who can use Power BI.
  + Business Analysts
  + Data Analysts and Data Scientists
  + Report Consumers (End Users)
  + Managers and Executives
  + Data Engineers
  + Developers and IT Professionals
  + Sales and Marketing Teams
  + Finance and Operations Teams
  + Customer Support Teams
  + Power BI Administrators
* Star Schema and its benefits.

Links:  
- [Top 30 Power BI Interview Questions and Answers](https://www.geeksforgeeks.org/power-bi-interview-questions-and-answers/)

* [25 Must-Know Power BI Interview Questions and Answers (2023)](https://www.dataquest.io/blog/power-bi-interview-questions-and-answers/)

**Tableau Questions:**

* What is Tableau?
* What are the different data connection options available in Tableau?
* How can you create a calculated field in Tableau?
* What is the difference between a dimension and a measure in Tableau?
* What is data blending in Tableau?
* What is the purpose of a parameter in Tableau?
* How can you perform data aggregation in Tableau?
* What Are the Data Types Supported in Tableau?
* What is Meant by ‘discrete’ and ‘continuous’ in Tableau?
* \*\* What Are the Filters? Name the Different Filters in Tableau.
  + Extract filters
  + Context filters
  + Data source filters
  + Filters on measures
  + Filters on dimensions
  + Table calculation filter
* What Are the Different Joins in Tableau?
* What is the Difference Between Joining and Blending?
* What is the Difference Between a Live Connection and an Extract?
* Is There a Difference Between Sets and Groups in Tableau?
* What is the Difference Between Treemaps and Heat Maps?
* What is the Difference Between .twbx And .twb?
* What Do You Understand by the Blended Axis?
* What is the Use of Dual-axis? How Do You Create One?
* What is the Level of Detail (LOD) Expression?
* How Can You Optimize the Performance of a Dashboard?
  + Maximize the number of fields and records. You can exclude unused fields from your visualization or use extract filters.
  + Limit the number of filters used, by avoiding quick filters and using action and parameter filters instead. These filters reduce query loads.
  + use Min/Max instead of Average because average functions require more processing time than Min/Max
  + Use boolean or numerical calculations more than string calculations. Computers can process integers and boolean much faster than strings.
* In Tableau, what are aggregation and disaggregation?
* What is KPI in Tableau?
* What are Bins in Tableau?
* What is a Quick Filter in Tableau?
* How do you calculate percentage in Tableau?
* What are the limitations of Tableau?
* What are cascading filters in Tableau?

**Python Questions:**