Phase 1: Knowledge & Analysis (100+ Questions)

Part A: Junior Data Analyst (0-2 years of experience)

1. What's the difference between a row and a column in a spreadsheet?
2. How would you count the number of empty cells in a column in Excel?
3. What's a primary key in a database?
4. Can you explain what a pivot table is and when you'd use it?
5. How do you calculate the average of a column in SQL?
6. What does the `GROUP BY` clause do in SQL?
7. How would you import a CSV file into a Pandas DataFrame in Python?
8. What's the difference between a list and a tuple in Python?
9. Can you explain what a data type is? Give a few examples.
10. What's the purpose of data cleaning?
11. What are the three measures of central tendency?
12. How do you find the maximum value in a column in SQL?
13. What is a "null" value in a database?
14. How would you handle a "divide by zero" error in a calculation?
15. What's the difference between `WHERE` and `HAVING` in SQL?
16. What's the `COUNT()` function in SQL used for?
17. How would you check for duplicate records in a table?
18. What's the purpose of a visualization like a bar chart or a line graph?
19. How would you create a new column in a Pandas DataFrame based on a condition?
20. What is a SQL `JOIN`? Name the most common types.
21. What's the difference between an inner join and a left join?
22. How do you filter data in SQL using the `WHERE` clause?
23. What's a common way to deal with missing data?
24. How do you sort data in a SQL query?
25. What's the difference between `SUM` and `COUNT`?
26. What does `SELECT *` do in SQL?
27. How do you find the number of unique values in a column in SQL?
28. What is the `ORDER BY` clause used for?
29. How would you create a simple bar chart in Python using a library like Matplotlib?
30. What's the difference between `SELECT` and `FROM` in a SQL query?
31. How do you find the top 5 rows with the highest sales in a table?
32. What is a data dictionary?
33. How would you merge two dataframes in Pandas?
34. What's the purpose of a `LIMIT` clause in SQL?
35. How would you create a calculated field in a pivot table?

Part B: Mid-Level Data Analyst (2-5 years of experience)

1. What's the difference between a `UNION` and `UNION ALL` in SQL?
2. Can you explain the concept of a **subquery** and provide an example of when you'd use one?
3. How would you handle **outliers** in a dataset? Name a few methods.
4. What's the difference between **correlation** and **causation**?
5. Explain the **normalization** process in a database. Why is it important?
6. What is a **window function** in SQL? Give an example of a common use case.
7. How would you use a **case statement** in a SQL query?
8. Describe a time when you had to deal with a messy dataset. What steps did you take?
9. What's the purpose of creating a **dashboard**? What makes a good one?
10. How would you deal with a **time-series** dataset where you have missing values on some dates?
11. What's the difference between a **stored procedure** and a simple query?
12. Can you explain the difference between **left join** and **full outer join**?
13. How would you calculate the **running total** of a column in SQL?
14. What are some of the common pitfalls of a `self-join`?
15. How would you calculate the **percentile** of a column in SQL?
16. What's the difference between **ETL** and **ELT**?
17. How do you handle a situation where a report you built is showing incorrect data?
18. What's the purpose of a **cross join**? When is it useful?
19. How would you identify the **top N** records for each group in a dataset?
20. Can you explain the concept of **data integrity**?
21. How do you deal with **skewed data** in a visualization?
22. What is the **ACID** property in a database?
23. Describe a situation where you used a **temporary table** or a **CTE (Common Table Expression).**
24. What are the pros and cons of using a **NoSQL database** compared to a SQL database?
25. How would you **automate** a weekly report you create?
26. Explain the concept of a **foreign key.**
27. What are some of the key things to look for during an **exploratory data analysis (EDA)** phase?
28. How would you use a **lag or lead function** in SQL?
29. How do you perform a **sentiment analysis** on text data?
30. How would you calculate the **month-over-month** growth of a metric?
31. What is the difference between a **data warehouse** and a **data lake**?
32. How would you measure the effectiveness of a new marketing campaign?
33. Describe a time when you had to present your findings to a non-technical audience. How did you do it?
34. What's the difference between a **view** and a **table** in a database?
35. What are some common data validation techniques you use?
36. How would you write a query to find the second-highest salary in a table?

Part C: Senior Data Analyst (5+ years of experience)

1. What's the difference between a **star schema** and a **snowflake schema**? When would you use each?
2. How would you approach a project to **optimize a slow-running SQL query**?
3. Describe your process for building a **data model** from scratch.
4. How would you set up an **A/B test** to measure the impact of a new website feature?
5. Can you explain the concept of **data governance** and why it's crucial for a company?
6. How would you use **recursive CTEs** in SQL? Give a real-world example.
7. What's the difference between **online analytical processing (OLAP)** and **online transaction processing (OLTP)**?
8. How would you identify and communicate the **business impact** of your analysis?
9. Explain the concept of **surrogate keys** and **natural keys**.
10. Describe a time when you had to **resolve a data inconsistency issue** across multiple data sources.
11. What's the difference between a **data pipeline** and an **ETL process**?
12. How would you handle **schema evolution** in a data warehouse?
13. What is **data lineage**, and why is it important?
14. Can you describe the difference between **descriptive, diagnostic, predictive, and prescriptive analytics**?
15. How would you implement a **slowly changing dimension (SCD)** Type 2 in a data warehouse?
16. What's your experience with **data visualization tools** like Tableau, Power BI, or Looker?
17. How would you use **change data capture (CDC)** in a data pipeline?
18. What are some of the key considerations for **data security and privacy** in your analysis?
19. How would you deal with a **large dataset** that doesn't fit into memory?
20. Describe your experience with **cloud platforms** like AWS, GCP, or Azure.
21. How do you **manage a backlog** of data requests and prioritize them?
22. What's the difference between a **view** and a **materialized view**?
23. How do you ensure the **quality and accuracy** of your data and analyses?
24. How would you design a **reporting system** for a new business unit?
25. What's the difference between **statistical significance** and **business significance**?
26. How would you use **indexes** to improve query performance?
27. Describe a time when you had to **mentor a junior analyst.** What did you teach them?
28. What are some of the **best practices for version control** in a data analysis project?
29. How would you approach a project that requires a **real-time dashboard**?
30. What's your experience with **data modeling techniques** beyond simple schemas?
31. How would you explain **data reliability and scalability** to a CEO?
32. What's the difference between **relational and non-relational databases** from a data analyst's perspective?
33. How would you conduct a **root cause analysis** for a sudden drop in a key business metric?
34. Describe a time you had to **advocate for a new tool or technology** for your team.
35. What is the role of **metadata** in a data analytics ecosystem?
36. What are the key components of a robust **data strategy**?

37. How do you stay up-to-date with the latest trends and tools in the data analytics field?

Phase 2: Coding & System Design (200+ Questions)

Part A: Fundamentals & Algorithms

Data Structures (50 questions)

1. What's the difference between an **array** and a **list**?
2. How would you implement a **stack** using a list?
3. What's the principle of a **queue**? (FIFO or LIFO?)
4. Can you explain the concept of a **hash table** and a **hash function**?
5. How do you handle a **collision** in a hash table?
6. What are the time complexities for searching, insertion, and deletion in a **singly linked list**?
7. What's a **doubly linked list**? What's its main advantage over a singly linked list?
8. How would you find the middle element of a linked list in a single pass?
9. What's a **tree** data structure? What's the difference between a tree and a graph?
10. What's a **binary search tree (BST)**? What is its main property?
11. How do you perform a **pre-order, in-order, and post-order traversal** of a binary tree?
12. What's the difference between a **complete binary tree** and a **full binary tree**?
13. What is a **heap**? What's the difference between a max-heap and a min-heap?
14. How would you implement a **priority queue**?
15. What's the purpose of a **trie** data structure?
16. Can you explain what a **graph** is? (Nodes/Vertices and Edges)
17. What's the difference between a **directed graph** and an **undirected graph**?
18. How do you represent a graph in code? (Adjacency Matrix vs. Adjacency List)
19. What are the time and space complexities for each graph representation?
20. What is a **tree traversal**? Name a few methods.
21. What is a **linked list cycle** and how would you detect it?
22. How would you reverse a linked list?
23. What is a **dictionary** or a **hash map**?
24. How would you count the frequency of each word in a text file?
25. What's the purpose of a **set** data structure?
26. How do you check if a set is a subset of another?
27. What are the time complexities for insertion and searching in a hash set?
28. What's the difference between a **static array** and a **dynamic array**?
29. How would you find a duplicate element in an array?
30. What's the difference between a **queue** and a **deque**?
31. How would you sort a stack using a temporary stack?
32. What's the time complexity of pushing an element to a stack and a queue?
33. How do you search for an element in a binary search tree?
34. What's the worst-case scenario for searching in a BST?

35. How would you find the height of a binary tree?
36. What is a **balanced tree**? Why is it important?
37. How would you insert a new node into a min-heap?
38. How do you remove the root from a max-heap?
39. What's the difference between a **breadth-first search (BFS)** and a **depth-first search (DFS)**?
40. When would you use BFS over DFS?
41. How would you find the shortest path between two nodes in a graph?
42. What is a **cycle** in a graph?
43. How would you detect a cycle in a directed graph?
44. What is a **singly linked list**?
45. How would you merge two sorted linked lists?
46. What's the difference between a **circular linked list** and a doubly linked list?
47. What is a **skip list**?
48. How would you find the second largest element in a BST?
49. What is the memory layout of an array vs. a linked list?
50. How would you implement a **priority queue** from scratch?

Algorithms (50 questions)

1. What's the difference between **time complexity** and **space complexity**?
2. Can you explain **Big O notation**? Give examples of O(1), O(n), and O(n^2).
3. Describe the **bubble sort** algorithm. What's its time complexity?
4. Explain **insertion sort**. What kind of data is it most efficient for?
5. What's the difference between **merge sort** and **quick sort**? Which one is generally faster?
6. How does **binary search** work? What's its time complexity?
7. What's the main requirement for using binary search?
8. Can you explain the **selection sort** algorithm?
9. What's the time complexity of the most efficient sorting algorithms?
10. How would you implement a recursive function?
11. What's the difference between **recursion** and **iteration**?
12. Explain the concept of **dynamic programming**.
13. What is **memoization**? How does it differ from a simple lookup?
14. Can you describe the **Fibonacci sequence** using recursion and then using iteration?
15. What's the purpose of a **greedy algorithm**?
16. How would you find the factorial of a number?
17. What's a **brute-force algorithm**?
18. Can you explain the **two pointers** technique?
19. What's the difference between a **linear search** and a **binary search**?
20. How would you check if a string is a **palindrome**?
21. What is the **Euclidean algorithm** for finding the greatest common divisor?
22. How would you find the **power** of a number efficiently?
23. Explain the **Sieve of Eratosthenes** algorithm for finding prime numbers.
24. What is a **bit manipulation** algorithm?

25. How would you count the number of set bits in an integer?
26. What's the purpose of a **backtracking** algorithm?
27. How does the **knapsack problem** work? (0/1 Knapsack)
28. What is the difference between a **greedy approach** and **dynamic programming** for the knapsack problem?
29. Explain **Dijkstra's algorithm** for finding the shortest path in a graph.
30. What's the purpose of a **topological sort**?
31. How would you find all permutations of a string?
32. What is the **traveling salesman problem**?
33. How would you find the maximum subarray sum in an array?
34. Explain the concept of a **sliding window** algorithm.
35. How would you find the longest substring without repeating characters?
36. What is a **divide and conquer** algorithm?
37. Can you explain the **Master Theorem** for analyzing the time complexity of recursive algorithms?
38. What is a **suffix array** or a **suffix tree**?
39. How would you find the **longest common subsequence** of two strings?
40. What's the difference between a **hash table** and a **hash set**?
41. How would you perform a **depth-first search** on a graph?
42. Explain how **breadth-first search** works on a graph.
43. What's the **Rabin-Karp** string matching algorithm?
44. How would you reverse the words in a string?
45. What is **amortized analysis**?
46. Explain the concept of a **minimum spanning tree**.
47. How does **Prim's algorithm** differ from **Kruskal's algorithm**?
48. What's the purpose of a **disjoint set union (DSU)** data structure?
49. How would you check for a cycle in an undirected graph using DSU?
50. Explain the concept of **NP-hard** and **NP-complete** problems.

Object-Oriented Programming (OOP) and Fundamentals (50 questions)

1. What are the four pillars of **Object-Oriented Programming**?
2. What is an **object**? What is a **class**?
3. Can you explain the concept of **encapsulation**?
4. How do you achieve **encapsulation** in your code?
5. What is **inheritance**? Why is it useful?
6. What's the difference between a **superclass** and a **subclass**?
7. Can you explain **polymorphism**? Give a real-world example.
8. What is **method overriding** vs. **method overloading**?
9. What is **abstraction**? What's the difference between an **abstract class** and an **interface**?
10. When would you use an abstract class over an interface?
11. What is a **constructor**?
12. What's the difference between a **default constructor** and a **parameterized constructor**?

13. What is **garbage collection**?
14. What are **static methods** and **static variables**?
15. What's the purpose of the `this` keyword?
16. What is a **singleton class**?
17. What are some of the benefits of using OOP?
18. What's the difference between a **procedural programming** paradigm and an **object-oriented** one?
19. What is **composition**? How does it differ from inheritance?
20. What is a **final class** or a **sealed class**?
21. What is an **exception**? How do you handle them?
22. What's the difference between a **checked exception** and an **unchecked exception**?
23. What's the purpose of a **try-catch-finally** block?
24. What is **concurrency**? How does it differ from **parallelism**?
25. Can you explain the concept of a **thread**?
26. What is a **race condition**? How do you prevent it?
27. What is a **deadlock**?
28. What is a **process**? What's the difference between a process and a thread?
29. How would you communicate between two processes?
30. What's the difference between **compile-time** and **run-time**?
31. What is a **virtual method**?
32. What is a **pure virtual function**?
33. What is an **assertion**?
34. What's the difference between a **value type** and a **reference type**?
35. How would you implement a simple **observer design pattern**?
36. What is **serialization**? Why is it useful?
37. What is **reflection** in a programming language?
38. What is the **dependency injection** pattern?
39. How do you manage **memory** in C++? (Stack vs. Heap)
40. What is a **pointer**?
41. What is a **lambda function** or an **anonymous function**?
42. What is a **closure**?
43. What's the purpose of a **decorator** in Python?
44. How does a **context manager** work in Python?
45. What is the difference between an **iterator** and a **generator**?
46. What's the purpose of a **list comprehension**?
47. How would you implement a **custom exception** class?
48. What is a **design pattern**?
49. Can you name a few common design patterns and explain their purpose?
50. How would you design a simple **banking system** using OOP principles?

Part B: Enterprise-Level & Domain-Specific

Business & Finance Domain (100 questions)

1. How would you design a database schema for a **Customer Relationship Management (CRM)** system?
2. What are the key tables needed for a **e-commerce platform**'s order management system?
3. How would you handle a **high volume of concurrent transactions** in a financial application?
4. Explain the concept of **idempotency** and why it's crucial for payment gateways.
5. How would you design a system to detect **fraudulent transactions**?
6. What is **ETL (Extract, Transform, Load)** and how is it used in a business intelligence context?
7. How would you create a system for **real-time stock price tracking**?
8. What is a **data warehouse** and what's its purpose in a business?
9. How would you handle **data encryption** for sensitive customer information?
10. Describe a robust system for **user authentication and authorization**.
11. How would you design a solution for **subscription billing**?
12. What are the key considerations for **data privacy regulations** like GDPR or CCPA?
13. How do you ensure **data integrity** in a distributed system?
14. How would you implement a system to generate **monthly financial reports**?
15. What is a **microservices architecture** and why would a company use it?
16. How would you design a system to manage a company's **payroll**?
17. How do you handle **currency conversion** in a global e-commerce application?
18. What is a **CAP theorem**? How does it apply to business systems?
19. How would you design a system to manage a company's **inventory**?
20. What are the pros and cons of using a **serverless architecture** for a financial app?
21. How would you handle **auditing and logging** for financial transactions?
22. What is a **distributed transaction** and how would you manage it?
23. How would you design a system to manage **product reviews and ratings** for an e-commerce site?
24. What are the main challenges when dealing with **legacy systems**?
25. How would you build a system to manage **loyalty points or rewards** for customers?
26. What is the difference between a **relational database** and a **document database**? When would you use each?
27. How would you design a system for **user session management**?
28. What are some strategies for **database sharding**?
29. How do you ensure **high availability** for a critical business application?
30. What is a **message queue** and how would you use it to decouple services?
31. How would you design a system to manage **employee records**?
32. How do you handle **data migration** from an old system to a new one?
33. What is a **load balancer** and what's its role in a scalable system?
34. How would you design a system for **product recommendations**?
35. What's the purpose of **data replication**? What are the different types?
36. How do you ensure **scalability** for a system that's expected to grow exponentially?
37. How would you handle **chargebacks and refunds** in a payment system?
38. What is **version control**? Why is it essential for enterprise-level coding?

39. How would you design a system to manage a company's **marketing campaigns**?
40. What are the key metrics you'd track for the health of a business application?
41. How would you build a system for **real-time dashboarding**?
42. What are the main challenges in a **multi-tenant application**?
43. How would you design a system to manage **customer support tickets**?
44. What's the purpose of a **content delivery network (CDN)**?
45. How would you design a system for **dynamic pricing**?
46. What is a **distributed cache** and why is it used?
47. How would you design a system to manage a company's **supply chain logistics**?
48. What are the main components of a **RESTful API**?
49. How do you handle **API versioning**?
50. What is **GraphQL** and how does it differ from a REST API?

Medical & Healthcare Domain (100 questions)

1. How would you design a database for an **Electronic Health Records (EHR)** system?
2. What are the key security and privacy regulations like **HIPAA** that must be followed in healthcare?
3. How would you handle **patient consent** and data access in a medical system?
4. What's the purpose of **medical imaging data** and how would you store and manage it?
5. How would you design a system to manage **patient appointments and scheduling**?
6. What are the challenges of **interoperability** between different hospital systems?
7. How would you design a system for **remote patient monitoring**?
8. What is the **HL7 standard** and why is it important in healthcare data?
9. How would you handle **audit trails** for every access to a patient's medical record?
10. What are some of the key metrics to track for a hospital's efficiency?
11. How would you design a system to manage **prescriptions and medication dispensing**?
12. What is the difference between **structured and unstructured medical data**?
13. How would you ensure **data quality and accuracy** for clinical research data?
14. How do you handle **secure data transfer** between a patient and a doctor's office?
15. How would you design a system for **telemedicine consultations**?
16. What are the security risks of using **mobile health (mHealth) apps**?
17. How would you design a system for **lab results management**?
18. What is the **ICD-10 or SNOMED CT** coding system?
19. How would you handle **data anonymization and de-identification** for research purposes?
20. What is a **DICOM file** and how would you handle it?
21. How would you design a system to manage a hospital's **inventory of medical supplies**?
22. What are the challenges of using **cloud computing** in the healthcare sector?
23. How would you build a system to predict **patient readmission rates**?
24. What are some of the ethical considerations of using **AI and machine learning** in healthcare?
25. How would you design a system for **clinical trial management**?

26. What is the purpose of a **blockchain** in healthcare?
27. How do you ensure **system uptime and reliability** for a life-critical application?
28. How would you design a system to track a **patient's wearable health data**?
29. What are the key components of a system for **electronic prescribing (e-prescribing)**?
30. How would you handle **emergency data access** for a first responder?
31. What are the challenges of integrating with **legacy EMR systems**?
32. How would you design a system for **vaccine management and distribution**?
33. How do you handle **real-time alerts** for critical patient conditions?
34. What's the purpose of a **biometric authentication system** in a hospital setting?
35. How would you build a system to manage a hospital's **billing and insurance claims**?
36. What is **interoperability** and how can it be achieved using standards like **FHIR**?
37. How would you design a system for a **personalized medicine platform**?
38. What are the main challenges of using **big data analytics** in healthcare?
39. How would you design a system for **remote surgery or telesurgery**?
40. How do you ensure the **traceability of medical devices**?
41. What is the role of a **health information exchange (HIE)**?
42. How would you design a system to manage **patient care plans**?
43. What are the ethical implications of using a **patient's genetic data**?
44. How would you handle **secure video conferencing** for remote patient visits?
45. What are the key components of a system for **hospital asset tracking**?
46. How would you design a system for a **public health disease surveillance**?
47. What are the main challenges of **data standardization** in healthcare?
48. How do you handle **system-wide outages** in a hospital?
49. How would you design a system for a **drug discovery platform**?
50. What is the **value-based care** model, and how does technology support it?

Phase 3: Practical Coding Questions

Part A: Junior Data Analyst (0-2 years)

Python/Pandas

1. Given a list of integers, write a function that returns a new list with only the unique elements.
2. How do you read a CSV file into a Pandas DataFrame?
3. Write a Python function to find the sum of all even numbers in a list.
4. Given a DataFrame `df` with a column 'Sales', how do you calculate the total sales?
5. How would you filter a DataFrame to show only rows where the 'Category' column is equal to 'Electronics'?
6. Write a Python script to count the number of occurrences of each word in a given string.
7. Given a DataFrame, how do you drop rows that contain any missing values?
8. How would you create a new column 'TotalPrice' in a DataFrame, which is the product of 'Quantity' and 'UnitPrice'?

9. Write a Python function to check if a string is a palindrome.
10. How do you sort a Pandas DataFrame by a specific column in descending order?

SQL 📊

1. Write a query to select all columns from a table named `customers`.
2. How would you count the total number of customers in a table?
3. Given a table `orders`, write a query to find the average `order_amount`.
4. Write a query to find all employees with a salary greater than 50,000.
5. How do you join two tables, `employees` and `departments`, on a common column `department_id`?
6. Write a query to find the total sales for each product in a `sales` table.
7. How would you find the top 10 most expensive products from a `products` table?
8. Write a query to find all unique product categories.
9. How do you find the minimum `order_date` from an `orders` table?
10. Write a query to find all customers who live in 'New York' or 'California'.

Part B: Mid-Level Data Analyst (2-5 years)

Python/Pandas 📈

1. Given a DataFrame with a 'TransactionDate' column, how do you convert it to a datetime object and then extract the year and month?
2. Write a function to identify and remove duplicate rows based on a specific set of columns.
3. How would you perform an inner join on two DataFrames using a common column?
4. Write a Python function that uses the `groupby()` method to calculate the mean 'Sales' for each 'Region'.
5. Given a DataFrame with 'Price' and 'Sales' columns, how do you calculate the correlation between them?
6. How do you create a pivot table in Pandas to show the total sales per region for each product category?
7. Write a function that handles missing values in a DataFrame by filling them with the mean of their respective columns.
8. How would you find and remove outliers from a numerical column using the Interquartile Range (IQR) method?
9. Write a Python script to read multiple CSV files from a directory and concatenate them into a single DataFrame.
10. How do you apply a custom function to a specific column in a DataFrame using `.apply()`?

SQL 🚀

1. What's the difference between `UNION` and `UNION ALL`? Provide an example of each.

2. Write a query to find the second-highest salary from an `employees` table without using `LIMIT` or `TOP`.
3. How would you find the running total of sales over time from an `orders` table?
4. Write a query to find the number of employees in each department, but only show departments with more than 50 employees.
5. How do you use a `subquery` to find all orders placed by customers from 'California'?
6. Write a query that uses a `CASE` statement to create a new column 'Sales_Tier' based on the 'Sales' amount (e.g., 'Low', 'Medium', 'High').
7. How do you calculate the year-over-year growth of a metric (e.g., `total_revenue`)?
8. Write a query to find customers who have not placed an order in the last 6 months.
9. How would you find the top 5 products with the highest sales in each product category?
10. Write a query to pivot data from a `sales` table to show sales for each month as a separate column.

Part C: Senior Data Analyst (5+ years)

Python/Pandas ⚙️

1. How do you optimize memory usage when working with a large dataset in Pandas? (e.g., using `astype('category')`, `Dask`, etc.)
2. Given a time-series dataset, how would you perform a rolling average calculation and then visualize the results using Matplotlib?
3. Write a class that represents a **data cleaning pipeline**, with methods for handling missing values, standardizing columns, and removing duplicates.
4. How do you handle a scenario where you have multiple files with slightly different schemas and you need to combine them?
5. Write a Python function to perform a multi-level `groupby()` and aggregate different columns with different functions (e.g., sum on one column, mean on another).
6. How would you implement a simple **A/B testing analysis** in Python, including calculating a p-value and determining statistical significance?
7. Write a script to read from a large database table in chunks to avoid memory errors.
8. How would you serialize a Pandas DataFrame into a different format like **Parquet** or **HDF5** and what are the advantages of doing so?
9. Write a function that uses a **decorator** to log the execution time of a data processing function.
10. How would you design a data validation script that checks for inconsistencies and data quality issues in a new dataset before it's used for analysis?

SQL 🏛️

1. Write a query to calculate the **percentile rank** of each employee's salary within their department.
2. Explain the concept of **window functions**. Write a query to find the `rank` of each product by sales within its category.

3. How would you use a **Common Table Expression (CTE)** to simplify a complex query that involves multiple joins and aggregations?
4. Write a query to perform a **self-join** to find pairs of employees who work in the same department.
5. How do you handle **slowly changing dimensions (SCDs)** in a data warehouse? Write a query to show a customer's address history.
6. Write a query to find the `first` and `last` order dates for each customer.
7. How do you optimize a query that is running very slowly on a large table? Mention at least three techniques.
8. Write a query that uses a **recursive CTE** to find all sub-categories and their children in a hierarchical product catalog.
9. Explain the difference between `ROW_NUMBER()`, `RANK()`, and `DENSE_RANK()`. Provide a scenario where each would be most appropriate.
10. Write a query to perform a **pivot** on data that is not in a simple group-by format, but where you have multiple entries for the same row.