Instructions for Data Engineering Test

Duration: Approximately [1 Hour]

Instructions:

* You have access to a relational database management system (RDBMS) such as Postgres, MySQL, or SQLite to perform SQL queries.
* Utilize Python or any scripting language of your choice to load data from Excel files into the database.
* Ensure that your SQL queries are correctly structured and follow best practices for readability and efficiency.
* Document any assumptions or limitations in your approach.
* Verify the correctness of your results and provide explanations for your solutions where necessary.
* Complete the tasks within the allotted time frame.
* You may use any relevant resources, such as documentation or online references, to assist you during the test. However, ensure that all solutions and code are your own work.

# TEST

1. Database Creation and Data Loading:

Question: Create a relational database schema to accommodate the provided data tables (transaction, account, GL, loan, and credit scoring). Define appropriate data types, primary keys, and foreign keys to establish relationships between tables.

Task: Use SQL to create the necessary tables in a relational database management system (RDBMS) such as Postgres, MySQL or SQLite. Once the tables are created, load the data from the provided Excel files into their respective tables using Python or a scripting language/in any way deemed appropriate.

1. Data Joining and Analysis:

Question: Write a SQL query to join the transaction, account, and credit scoring tables to retrieve the following information for each customer: Customer ID, Transaction ID, Account Number, Transaction Amount, Account Balance, Credit Score, and Credit Rating.

Task: Use SQL to perform a join operation between the transaction, account, and credit scoring tables based on the shared Customer ID. Execute the query and examine the results to ensure that each customer's transaction details, account information, and credit scoring data are appropriately combined.

1. Data Engineering Challenges:

Question: Calculate the average transaction amount for customers with a credit score above 700, grouped by their credit rating. Additionally, determine the total number of transactions made by customers in each credit rating category.

Task: Write a SQL query that filters customers based on their credit score, calculates the average transaction amount for each credit rating category, and counts the total number of transactions made by customers in each category. Execute the query to obtain the desired metrics.

1. Data Integrity and Validation:

Question: Identify any discrepancies between the GL data and the transaction data. Specifically, find transactions that are not recorded in the GL data and GL entries that do not have corresponding transactions.

Task: Write SQL queries to compare the transaction data with the GL data based on the shared Transaction ID. Identify any discrepancies between the two datasets, such as missing transactions or GL entries. Investigate the root cause of these discrepancies and propose potential solutions.

1. Performance Optimization:

Question: Evaluate the performance of the database queries used in the previous questions. Identify any potential bottlenecks and propose optimizations to improve query execution time, considering factors such as indexing, query structure, and database configuration.

Task: Analyze the execution plans and query performance metrics generated by the database management system. Identify any inefficient query patterns or resource-intensive operations. Propose optimizations, such as creating indexes on frequently queried columns, rewriting complex queries to improve efficiency, or tuning database configuration parameters to better utilize hardware resources.