iCustomer

Coding Test for Senior Data Engineer

Objective:

To evaluate the candidate's proficiency in data engineering, including data pipelines, ETL processes, data storage solutions, and distributed systems.

Duration:

2-3 hours

Submission:

Candidates should submit their code via a GitHub repository.

Task 1:Data Ingestion and ETL

Scenario:

You are working for a Data Product company that needs to process user interaction data. The data is in CSV format and needs to be cleaned, transformed, and loaded into a data warehouse for analysis.

Steps:

1. Data Ingestion:

    - Write a Python script to ingest the interaction data from a given CSV file.

- The CSV file will have columns: `interaction_id`, `user_id`, `product_id`, `action`, `timestamp`.

2. Data Cleaning:

- Handle missing values by either filling with a default value or removing the rows.

- Ensure that the data types are correct (e.g., `timestamp` should be a datetime type).

3. Data Transformation:

- Calculate the number of interactions per user and per product.

- Add a new column `interaction_count` to the data for each user and product.

4. Data Loading:

- Load the cleaned and transformed data into a SQL database (e.g., SQLite or PostgreSQL).

5. Documentation:

- Document your ETL process clearly in a README file.

Deliverables:

- Python script(s) for data ingestion, cleaning, transformation, and loading.

- SQL database file with the loaded data.

- README file with instructions on how to run the scripts and any dependencies required.

iCustomer

Task 2: Data Pipeline with Apache Airflow

Scenario: Automate the ETL process from Task 1 using Apache Airflow.

Steps:

1. Set Up Airflow:

   - Install Apache Airflow and set up a local environment.

2. Create an Airflow DAG:

   - Define a DAG that includes tasks for data ingestion, cleaning, transformation, and loading as defined in Task 1.

3. Task Scheduling:

   - Schedule the DAG to run daily.

4. Error Handling:

   - Implement error handling and logging within the DAG to manage and record any issues that occur during the ETL process.

Deliverables:

- Airflow DAG file(s).

- Instructions on how to set up and run the Airflow environment.

iCustomer

Task 3: Data Storage and Retrieval

Scenario: Implement a solution to store and retrieve user interaction data efficiently.

Steps:

1. Data Storage:

- Design a schema for storing user interaction data in a SQL database (e.g., PostgreSQL).

- Implement the schema in the chosen database.

2. Data Retrieval:

- Write SQL queries to answer the following questions:

  1. Total number of interactions per day.

  2. Top 5 users by the number of interactions.

  3. Most interacted products based on the number of interactions.

3. Optimization:

- Discuss any optimization techniques you used to improve the performance of your queries.

Deliverables:

- SQL scripts for schema creation and data insertion.

- SQL queries for data retrieval.

- A document explaining the optimization techniques used.

Instructions for Submission:

1. Create a GitHub repository and commit all your code, scripts, and documentation.

2. Ensure that your repository is well-structured and includes a README file with clear instructions on how to set up and run your solutions.

3. Share the GitHub repository link for review.

Evaluation Criteria:

1. Code Quality:Readability, organization, and adherence to best practices.

2. Functionality:Correctness and completeness of the ETL process, Airflow DAG, and SQL queries.

3. Documentation:Clarity and thoroughness of the provided documentation.

4. Efficiency:Performance and optimization of the data retrieval queries.

5. Error Handling:Robustness of the ETL process and error handling in Airflow.