Assignment - Senior Data Engineer

Task Overview

You are required to design and implement a data pipeline that extracts data from a SQL database, transforms the data, and loads it into a NoSQL database. The pipeline should be orchestrated using either Airflow or Prefect and should run every 3 hours. The orchestration should handle fault tolerance and retries. You will also need to provide a brief explanation of the data structures used and their time complexities. Additionally, generate four different insights from the data using Polars or Pandas and store these insights in the MongoDB along with the aggregated data.

Assignment Details

Data Source Design

You will be working with an e-commerce dataset consisting of six tables: `customers`, `orders`, `order_items`, `products`, `categories`, and `reviews`. Your task is to extract data from these tables, perform necessary cleaning and transformations, and load the aggregated data and insights into a NoSQL database.

Data source table structure

```
-- Table: customers

CREATE TABLE customers (

   customer_id INT PRIMARY KEY,

   name VARCHAR(100) NOT NULL,

   email VARCHAR(100) UNIQUE NOT NULL,

   country VARCHAR(50) NOT NULL

);


-- Table: orders
```

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT NOT NULL,
    order_date DATE NOT NULL,
    total_amount DECIMAL(10, 2) NOT NULL,
    status VARCHAR(20) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- Table: order_items
CREATE TABLE order_items (
    item_id INT PRIMARY KEY,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

-- Table: products
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100) NOT NULL,
    category_id INT NOT NULL,

    FOREIGN KEY (category_id) REFERENCES categories(category_id)

);


-- Table: categories

CREATE TABLE categories (

    category_id INT PRIMARY KEY,

    category_name VARCHAR(100) NOT NULL

);


-- Table: reviews

CREATE TABLE reviews (

    review_id INT PRIMARY KEY,

    product_id INT NOT NULL,

    customer_id INT NOT NULL,

    rating INT CHECK (rating BETWEEN 1 AND 5),

    review_date DATE NOT NULL,

    FOREIGN KEY (product_id) REFERENCES products(product_id),

    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);
```

Assignment Details

1. Data Extraction

- Extract data from the `customers`, `orders`, `order_items`, `products`, `categories`, and `reviews` tables in a SQL database.

2. Data Cleaning

- Handle missing values, duplicates, and any inconsistencies in the data.

- Ensure data types are correct and consistent across all tables.

3. Data Transformation

- Join the `customers`, `orders`, `order_items`, `products`, `categories`, and `reviews` tables on relevant keys.

- Aggregate the data to calculate the total amount spent, the number of orders, and the average order value by each customer, along with the number of products ordered and average rating received.

- The result should include `customer_id`, `name`, `email`, `country`, `total_amount_spent`, `total_orders`, `average_order_value`, `total_products_ordered`, and `average_rating`.

4. Insights Generation

- Generate four different insights from the data using Polars or Pandas. These could include:

1. Top 5 customers by total amount spent.

2. Top 5 products by number of orders.

3. Average rating of products by category.

4. Monthly sales trend.

5. Data Loading

- Load the aggregated data and the generated insights into a NoSQL database, preferably MongoDB. Data modeling will be your choice.

6. Orchestration

- Use Airflow or Prefect to orchestrate the ETL pipeline.

  - Create a DAG (Directed Acyclic Graph) or flow that includes tasks for data extraction, cleaning, transformation, and loading.

  - Schedule the pipeline to run every 3 hours.

  - Implement fault tolerance and retries. Ensure that the pipeline can recover from failures and retry tasks where necessary.

  - Include error handling and logging.

7. Documentation

- Provide a README file with:
  - Setup instructions for the SQL and NoSQL databases.
  - Instructions on how to run the pipeline.
  - Explanations of the data structures used and their time complexities (Big O notations).
  - Discussion of any challenges faced and solutions implemented.

Submission Requirements

1. **Code Repository:** Provide a link to a public GitHub repository containing your code.

2. **README File:** Include a detailed README file with setup instructions, explanations, and any other relevant information.

3. **Orchestration Script:** Ensure that the Airflow DAG or Prefect flow is included and well-documented.

4. **Data Samples:** Provide sample data for the SQL and NoSQL databases to demonstrate the pipeline.

Deadline

Please complete the assignment within three days from the date of receiving it. If you have any questions or need clarifications, feel free to reach out.