Sinch RTCx Data Engineer Assignment

Objective: Create a data **warehousing** solution that reflects real-world data engineering challenges, showcasing your skills in data modelling, SQL, and Python.

Introduction to the Business Domain: The Sinch Grill

The Sinch Grill, fictional business - which encompasses three primary teams:

1. **Members Team:** Manages member data, including membership types and preferences.
2. **Marketing Team:** Oversees marketing campaigns to boost member engagement and sales.
3. **Orders Team:** Handles customer orders and tracks their fulfilment status.

Each team maintains independent microservices and databases. While current communication between services occurs via point-to-point JSON with shared primary domain keys, plans include transitioning to a data bus architecture.

Domain Details

Members Domain:

- **Overview**: The Sinch Grill offers a membership-based service with different tiers that members can access for a defined time. Memberships are free but require regular renewals and spending thresholds to maintain or upgrade levels. Members have preferences that are automatically applied to their orders, like requesting extra onions or no pickles.

Marketing Domain:

- **Overview**: Marketing campaigns are tailored to increase member engagement and boost sales. Campaigns can be specific to individual stores and are currently aimed at total sales, with plans to target specific items in the future.

Order Domain:

- **Overview**: The order domain manages data related to customer orders, tracking items within each order, customizations based on preferences, and the status of the orders (e.g., submitted, in progress, delivered). Nested items (e.g., a combo meal including a burger, fries, and shake) are supported. Sales are final, with no voids or refunds. The subtotal reflects the base price of the items, while the total includes taxes and discounts.

Provided Schema Context

To design your data warehouse schema, you can rely on the following structure:

Members Domain

- **Members**: Contains details like member ID, name, membership type, join date, and expiration date.
- **Preferences**: Stores member preferences (e.g., extra onions, no pickles) linked by member ID.

Marketing Domain

- **Campaigns**: Includes campaign ID, target audience, and store-specific details.

Order Domain

- **Orders**: Contains order ID, member ID, order date, subtotal, and total after taxes and discounts.
- **Order Items**: Details the items within each order, supporting nested structures.
- **Order Status**: Tracks the status changes of orders.

Assignment Requirements

1. Data Warehousing & Modelling

Task: Design a data warehouse schema based on the provided domain details. The schema should support the following use cases:

- Analyse gross revenue by campaign, store, customer, and membership type.
- Identify the most and least popular items.
- Assess average order processing and delivery times.

Deliverable:

- A detailed ERD (Entity-Relationship Diagram) representing the schema.
- Justification for your design choices, including table structures, keys, relationships, and performance optimizations (e.g., partitions, indexing).
- SQL Scripts for the creation of the database, tables, indices and any other related database objects. These scripts should be compatible with PostgreSQL.

2. ETL Scripts

Task: Develop Python scripts to load provided CSV data into the database designed above.

Deliverables:

- Well-structured Python scripts functioning as ETL jobs.
- Clear comments explaining any assumptions made in the scripts.

3. SQL Queries

Task: Write SQL queries to address the following use cases:

- Report totals gross revenue by store and campaign.
- Report on highest and lowest spending members for each month.
- Retrieve the top 2 most and least popular items for each month.
- Calculate average order processing and delivery times per store.

Deliverable:

- SQL scripts with comments explaining the logic behind each query.

Docker Setup for Development

A `docker-compose.yml` file is provided in the docker folder of this assignment to help you set up the development environment. This will spin up a PostgreSQL container along with pgAdmin to assist you in writing the required SQL and Python scripts.

Instructions for Setup:

1. Open a terminal and navigate to the docker folder.
2. Run the following command to start the containers:
   ```
   docker-compose up -d
   ```
3. Once the containers are up, access pgAdmin at: `http://localhost:5050`
4. Once you are done with the assignment run the following command to clean up the containers:
   ```
   docker-compose down -v
   ```

Database Connection Details:

- **Host (from Python or other scripts running locally):** `localhost`
- **Port:** `5432`
- **Username:** `postgres`
- **Password:** `mysecretpassword`

Use these credentials for connecting to the database while working on your Python ETL scripts or other SQL queries.

Evaluation Criteria

- **Completeness**: Does the schema address all require use cases? Are SQL queries effective?
- **Code Quality**: Is the code clean, well-structured, and adherent to best practices?
- **Performance**: Are data processing and queries optimized?
- **Documentation**: Are justifications and explanations clear and professional?
- **Bonus (Optional)**: Additional features, such as visualizations or advanced optimizations, will be considered positively.

Submission Guidelines

- **Format**: Submit your work as a GitHub repository.
- **Components**: Include the data warehouse schema, SQL scripts, Python ETL scripts, and documentation.
- **Deadline**: Complete the assignment within one week of receipt.