**ETL Pipeline for Consumer Insights**

This ETL pipeline processes customer and transaction data to identify top customers and analyze spending patterns. It includes data cleaning, transformation, and loading into a SQLite database, with support for incremental updates. Visualizations and unit tests are also provided.

**Prerequisites**

* Python 3.6+
* SQLite3
* Pandas
* SQLAlchemy
* pytest

**Installation**

1. Clone the repository or download the file directly in git:
   * git clone <https://github.com/yourusername/etl-code-test.git>
   * cd etl-pipeline-consumer-insights
2. Install the required packages:
   * pip install -r requirements.txt

**Project Structure**

* etl\_pipeline.py: Contains the ETLProcessor class and the main function.
* test\_etl\_pipeline.py: Contains unit tests for the ETL pipeline.
* data/: Directory containing raw data files (users-1.csv, transactions-1.csv, pricing-1.csv).
* output/: Directory for output files.
* database/: Directory for output database.
* requirements.txt: List of required Python packages.

**Data Files**

* users-1.csv: Contains user data with columns user\_id, name, email, date\_joined.
* transactions-1.csv: Contains transaction data with columns trans\_id, user\_id, product, amount, trans\_date.
* pricing-1.csv: Contains pricing data with columns puk, product, price.

**Running the ETL Pipeline**

1. Ensure the raw data files are in the data/ directory.
2. Run the ETL pipeline:
   * python etl\_pipeline.py

**ETLProcessor Class**

The ETLProcessor class handles the entire ETL process, including loading data, cleaning and transforming data, and loading it into a SQLite database.

**sUnit Tests**

Unit tests are provided to ensure the reliability of the ETL pipeline.

1. Ensure you have pytest installed:
   * pip install pytest
2. Run the tests:
   * pytest test\_etl\_pipeline.py

**Visualizations**

The ETL pipeline includes data exploration and visualization to provide insights into customer spending patterns.

* **Top Customers**: Bar charts showing total spending by customer.
* **Spending Patterns**: Line graphs illustrating spending trends over time.
* **Customer Segments**: Pie charts or histograms depicting customer distribution by spending brackets.

**Using Looker Studio for Visualization**

You can access this site for the sample dashboard I crafted looker studio

*https://lookerstudio.google.com/reporting/d937d299-de28-43d2-976d-c388a1406657*

1. **Export Data**: Export the cleaned and transformed data to CSV files.
   * **­located at output folder**
2. **Upload CSV to Google Sheets**:
   * Open Google Sheets.
   * Create a new spreadsheet for users.csv and transactions.csv.
   * Use the "File" -> "Import" -> "Upload" option to upload each CSV file.
3. **Connect Google Sheets to Looker Studio**:
   * Go to [Looker Studio](https://lookerstudio.google.com/).
   * Create a new report and select "Google Sheets" as the data source.
   * Select the Google Sheets file you uploaded and configure the connection.
4. **Create Visualizations**:
   * Use Looker Studio's tools to create bar charts, line graphs, pie charts, etc.
   * Example visualizations:
     + **Top Customers**: Bar chart showing total spending by customer.
     + **Spending Patterns**: Line graph illustrating spending trends over time.
     + **Customer Segments**: Pie chart depicting customer distribution by spending brackets.

**Production Setup**

For a production environment, consider the following enhancements:

1. **Database**: Migrate from SQLite to PostgreSQL or MySQL for better performance and scalability.
2. **ETL Orchestration**: Use Apache Airflow or similar tools for scheduling and monitoring ETL jobs.
3. **Data Storage**: Store raw and processed data in a data warehouse like Amazon Redshift or Google BigQuery.
4. **Reporting**: Integrate with BI tools like Tableau or Power BI for advanced reporting and dashboarding.