**Data Processing and Analytics System**

This repository provides a comprehensive data processing and analytics system, designed to handle transactional data efficiently. The system incorporates data ingestion, transformation, and analytics features, while exposing endpoints through FastAPI for real-time data retrieval.

**Table of Contents**

* [Features](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#features)
* [Getting Started](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#getting-started)
  + [Environment Setup](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#1-environment-setup)
  + [Database Configuration](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#2-database-configuration)
  + [Run Data Processing](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#3-run-data-processing)
  + [Run FastAPI Server](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#4-run-fastapi-server)
  + [Example Usage](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#5-example-usage)
* [FastAPI Endpoints](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#fastapi-endpoints)
* [Analytics](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#analytics)
* [Contributing](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#contributing)
* [License](https://chat.openai.com/c/a7a30757-b3af-4970-8056-88ca1b8091bd#license)

**Features**

* **Data Ingestion:** Ingests transactional data from CSV files into a PostgreSQL database.
* **Data Transformation:** Transforms the raw data into silver and gold layers, enhancing its structure and usability.
* **Dimension Tables:** Creates dimension tables for customer information and expenditure types.
* **FastAPI Endpoints:**
  + **send\_transaction**: Accepts transaction data via POST requests for real-time ingestion.
  + **get\_total\_amount**: Retrieves the total amount for a given transaction ID (**trans\_id**).
  + **number\_of\_transactions**: Retrieves the total number of distinct transactions for a given customer ID (**cust\_id**).
  + **data\_movement\_time**: Provides statistics on data movement time, including average, maximum, and minimum.
  + **check\_rows**: Compares the total number of rows in the database with the number of rows logged.

**Getting Started**

**1. Environment Setup**

* Install the required Python packages using **pip install -r requirements.txt**.
* Ensure PostgreSQL is installed and running.

**2. Database Configuration**

Modify the **connection\_params** in the scripts to match your PostgreSQL database configuration.

**3. Run Data Processing**

* Execute the data processing scripts in the following order:
  1. **data\_processing.py**: Creates tables, ingests CSV data, and performs data transformations.
  2. **data\_analytics.py**: Implements analytics on the processed data.

**4. Run FastAPI Server**

Execute **uvicorn api.py** to start the FastAPI server.

**FastAPI Endpoints**

* **/send\_transaction**: POST endpoint to ingest transaction data.
* **/get\_total\_amount/{trans\_id}**: GET endpoint to retrieve total amount by transaction ID.
* **/get\_total\_transactions/{cust\_id}**: GET endpoint to retrieve the total number of distinct transactions by customer ID.
* **/data\_movement\_time/**: GET endpoint to get statistics on data movement time.
* **/check\_rows/**: GET endpoint to compare total rows in the database with rows logged.

**Analytics**

* The analytics script (**data\_analytics.py**) performs various analytical queries on the processed data, including total amounts, transaction counts, and data movement time statistics.

**Contributing**

Feel free to contribute by opening issues, proposing features, or submitting pull requests. Your feedback is highly appreciated!