Populating a PostgreSQL Database with Financial Data

Overview

This report provides an overview of the system developed to populate a PostgreSQL database with financial data. The system is designed to efficiently gather, process, and store large volumes of financial information. In the second phase of our project, we migrated our tables from the relational database to the MongoDb NoSQL database, which uses document-store.

Approach

The approach taken to populate the PostgreSQL database the following steps:

1. **Identifying Data Sources**: We identified multiple APIs that provide comprehensive financial datasets. These included stock prices, market news, and company information.
2. **API Integration**: We integrated these APIs into our system using custom scripts that automate data retrieval at regular intervals.
3. **Data Cleaning and Transformation**: Raw data from the APIs was cleaned to remove any inconsistencies or errors. It was then transformed into a normalized format that aligns with our database schema.
4. **Database Population**: The cleaned and transformed data was inserted into the PostgreSQL database using batch processing techniques to optimize performance.

Challenges Faced

During the development and implementation of this system, we encountered several challenges:

- **Data Volume Limitations**: Initially, we struggled to obtain sufficient data to populate our database with 300MB of financial information. This limitation was primarily due to restrictions on free API usage, which often impose limits on the amount of data that can be accessed.
- **Solution**: To overcome this challenge, we opted to purchase premium API subscriptions, which provided us with access to larger datasets. This investment allowed us to meet our data volume requirements effectively.
- **Data Quality Assurance**: Ensuring the quality and consistency of incoming data was another challenge. Variations in data formats across different APIs required us to implement robust validation and transformation processes.

Conclusion

The system successfully populates a PostgreSQL database with financial data by leveraging paid API services. Despite initial challenges related to data volume limitations and quality assurance, strategic investments in premium API access and rigorous data processing protocols enabled us to achieve our objectives.