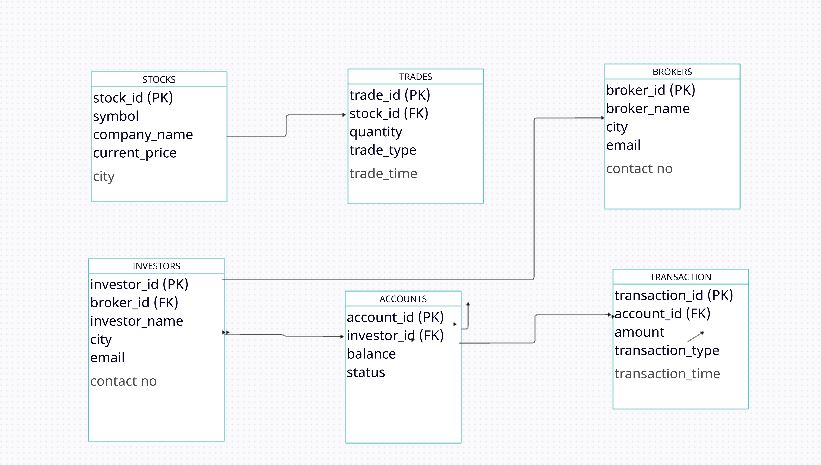
**Distributed Stock Trading Database – Project Report**

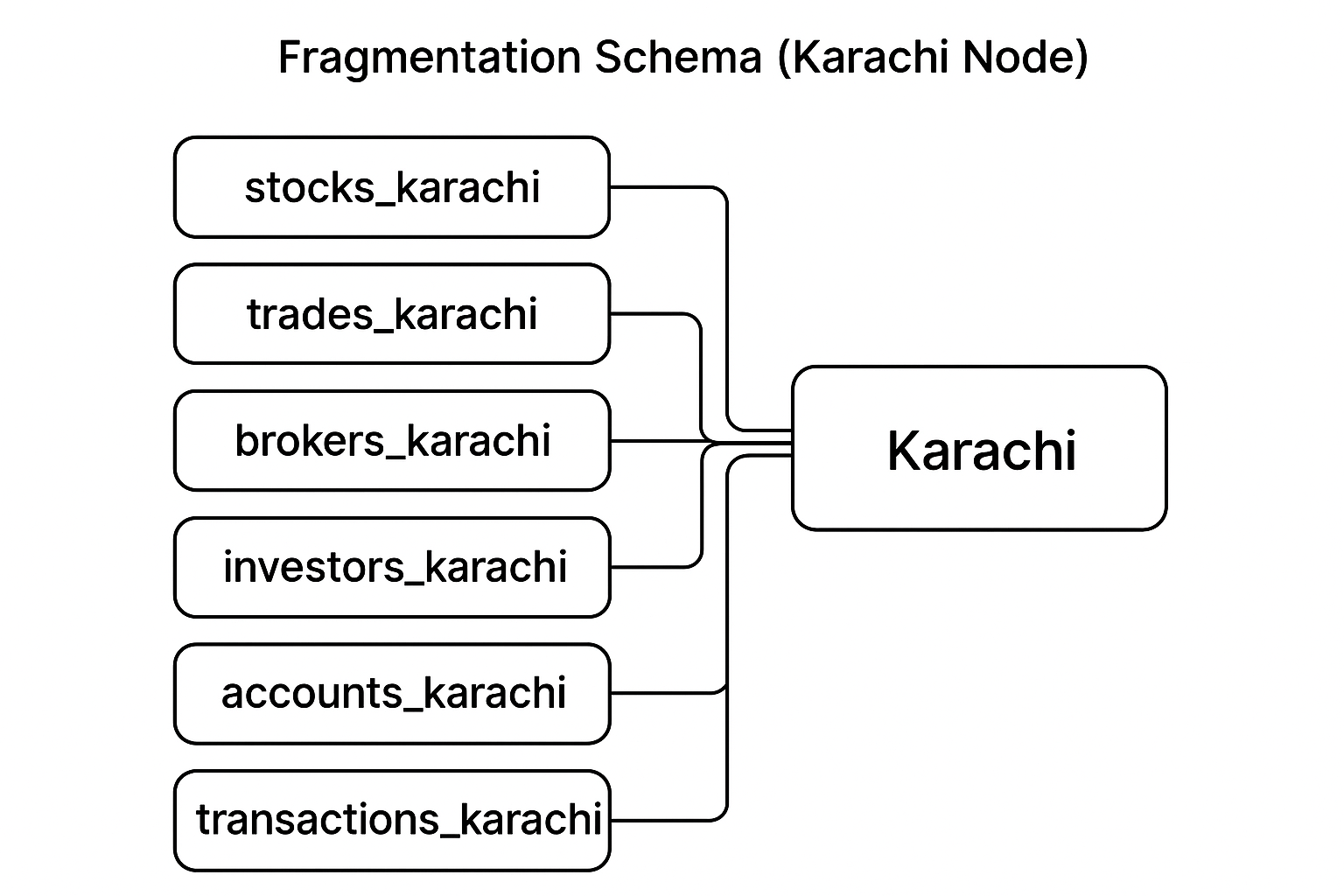
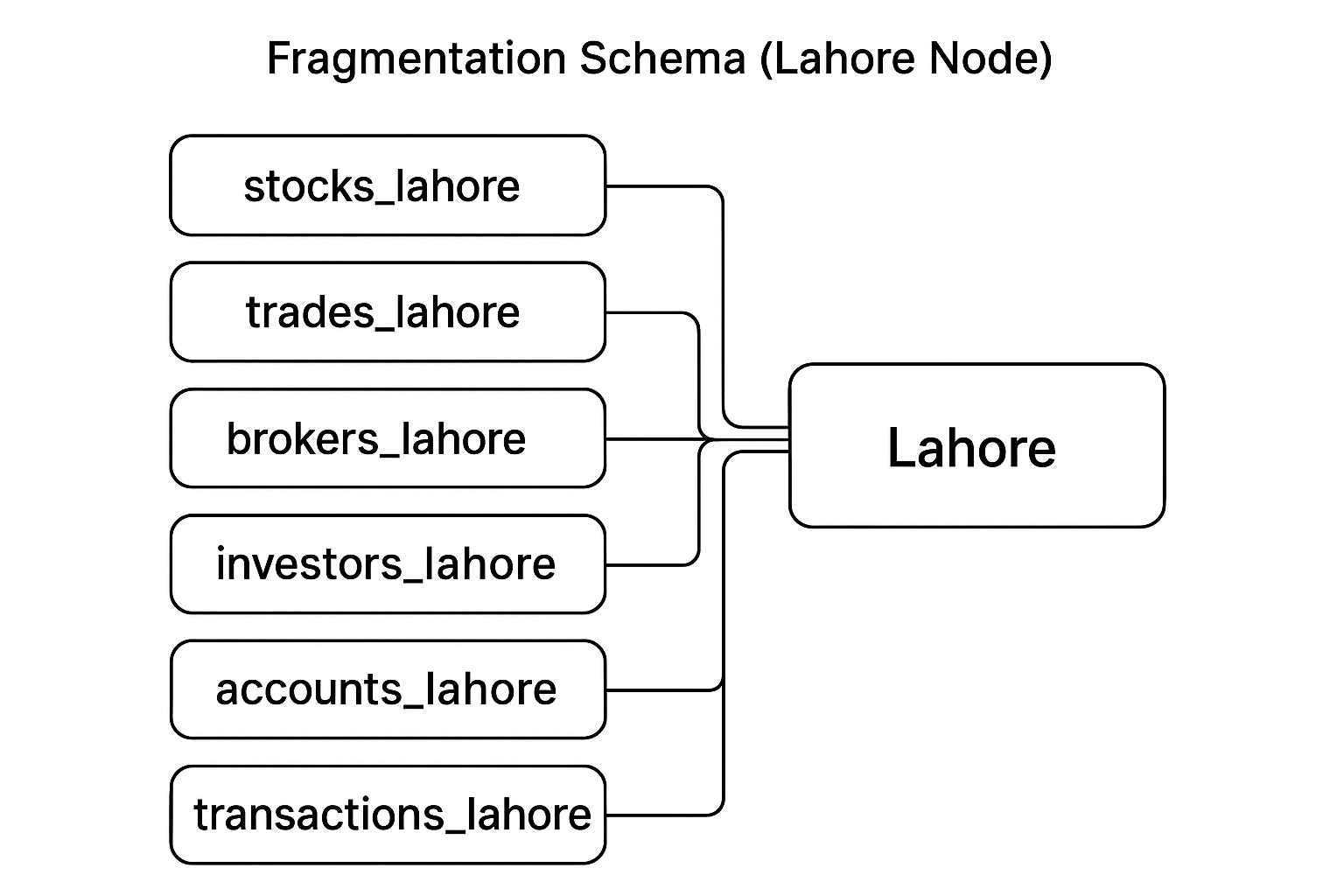
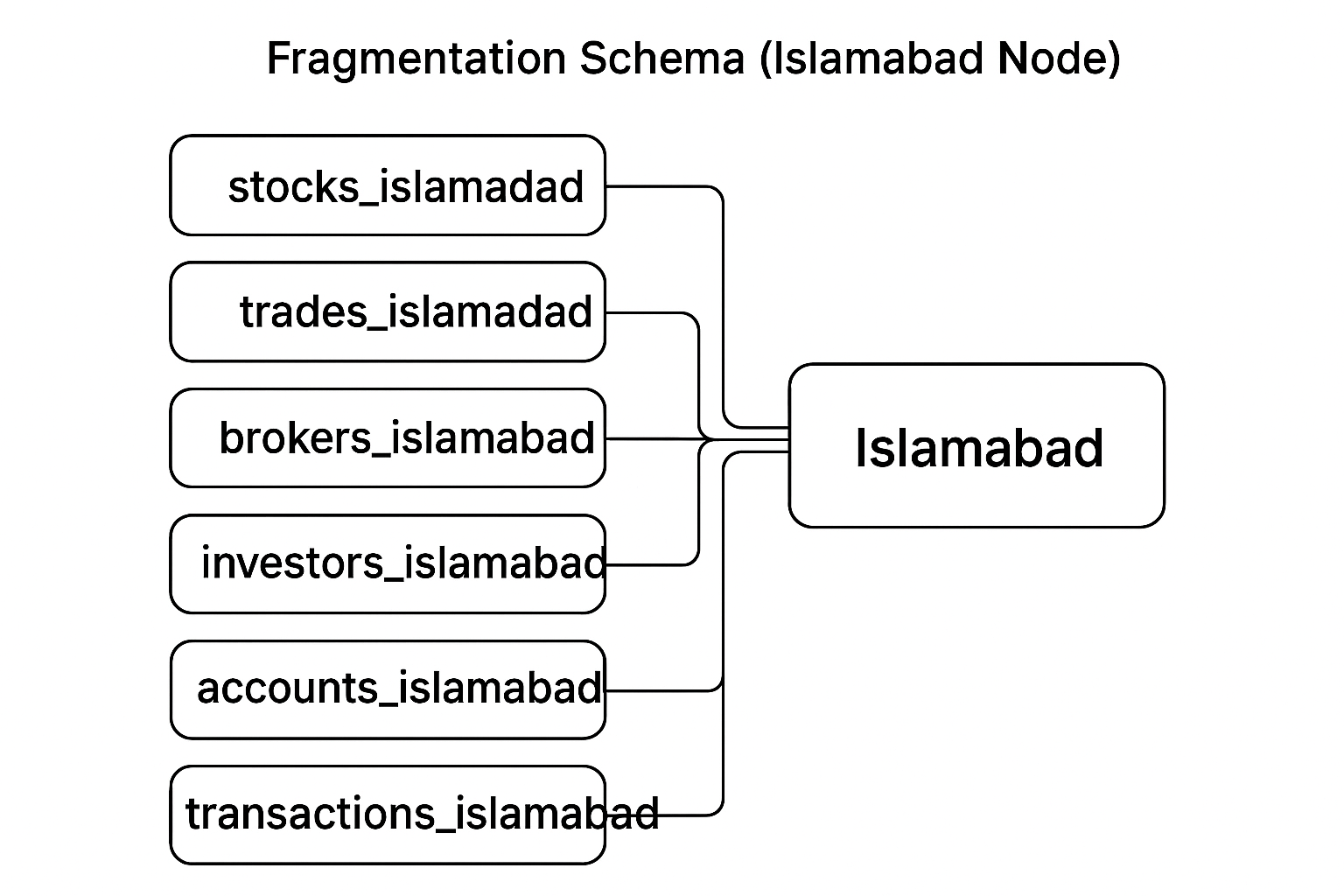
**Introduction**

This project focuses on developing a distributed database for a real-time stock trading system using PostgreSQL. The main goal was to efficiently manage stock and trade data across multiple nodes, ensuring high performance, security, and scalability. Instead of replication, we used **Foreign Data Wrappers (FDW)** to integrate and access data from multiple servers, allowing seamless querying across distributed nodes.

**Project Objectives**

The project aimed to implement a distributed database architecture that supports data fragmentation, query optimization, and secure access control. Key objectives included designing distributed database schemas for stocks and trades, implementing horizontal, vertical, and mixed fragmentation strategies, optimizing queries for efficient data retrieval, and ensuring robust security through user roles and permissions. The project also evaluated database performance and provided insights for future enhancements.





**1. Add Table Summary Section**

**Tables Overview**  
The system consists of six main tables across nodes:

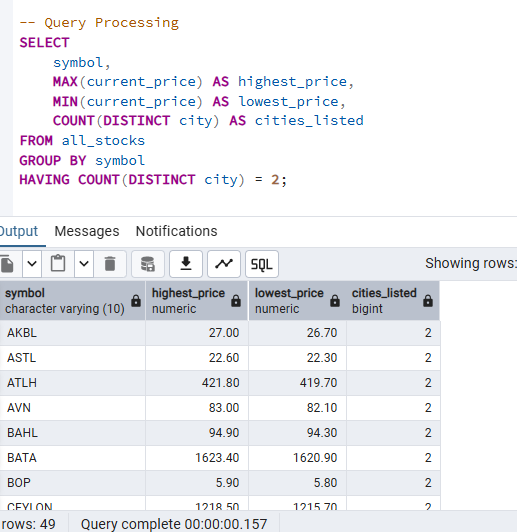
* Stocks: Holds stock information and current prices.
* Trades: Records each stock trade transaction.
* Brokers: Contains broker profiles and trade handling details.
* Investors: Stores investor identities and account links.
* Accounts: Maintains investor financial balances.
* Transactions: Tracks deposits and withdrawals.

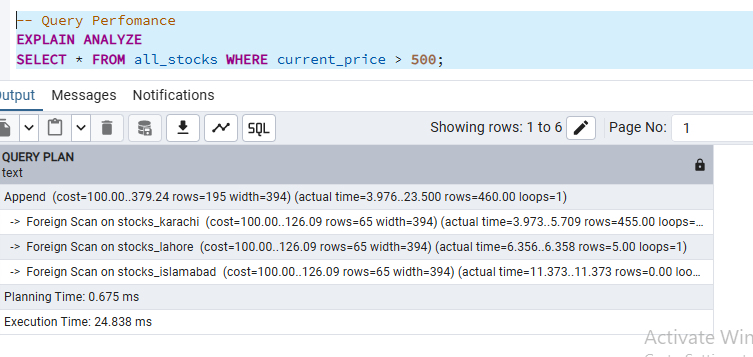
**2. Schema Design and Fragmentation Section**

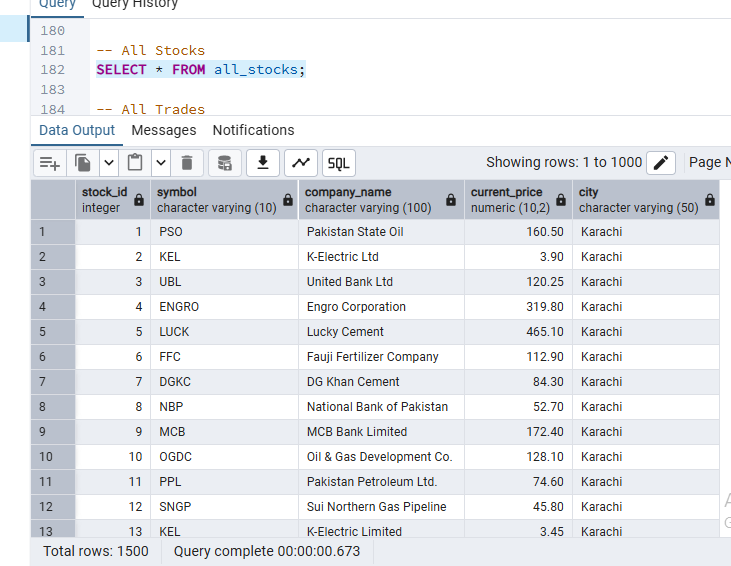
Horizontal fragmentation was applied based on stock location (Karachi, Lahore, Islamabad).  
Each node maintains local data relevant to its city — e.g., *stocks* , *trades , brokers , accounts , transactions and investor* on the Karachi node.  
Vertical fragmentation separated large transaction details into the *transactions* table for scalability.s

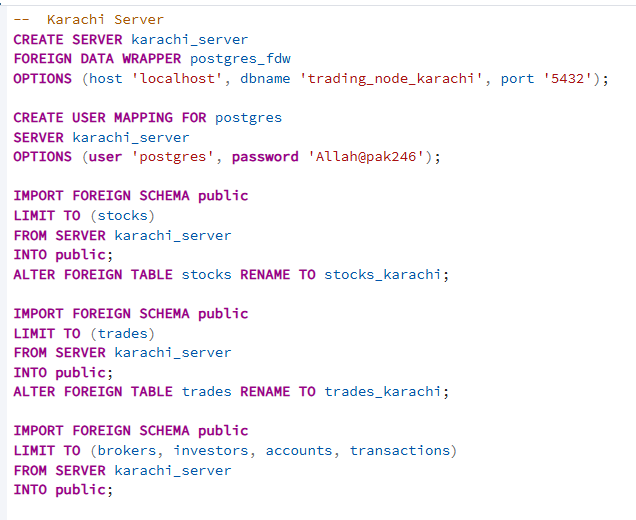
**3. “Query Processing and Optimization”**

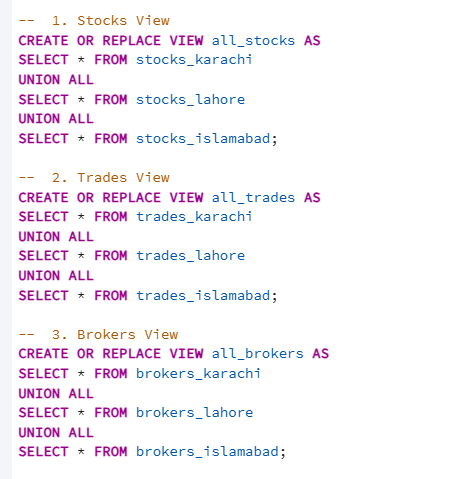
“While FDW enables cross-node query execution, performance depends on network latency and foreign scan efficiency. Query optimization was achieved by using ANALYZE and EXPLAIN plans to reduce remote lookups.”

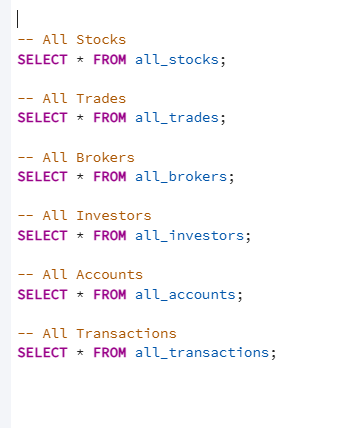
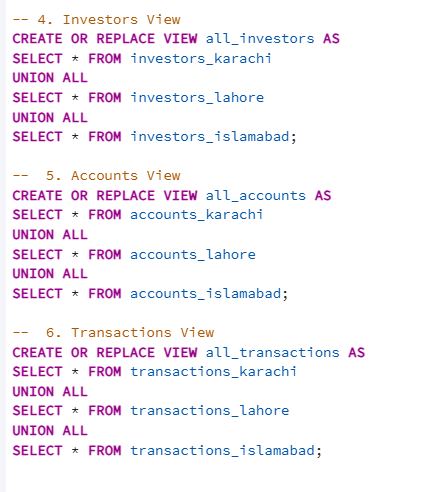




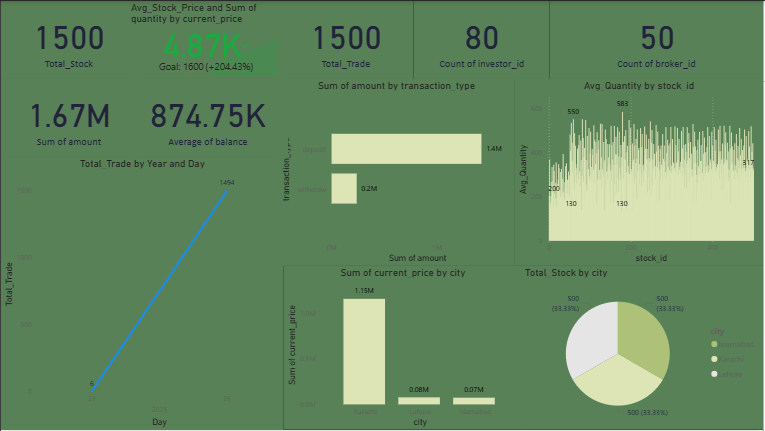








**Visualization of Stock Trading Dashboard**



**Conclusion**

This project successfully demonstrates the design and implementation of a Distributed Stock Trading Database System using PostgreSQL and Foreign Data Wrappers (FDW). By applying horizontal, vertical, and mixed fragmentation, the database achieves efficient data distribution, scalability, and optimized query performance across multiple city-based nodes (Karachi, Lahore, and Islamabad). The integration of FDW enables seamless access to remote data without replication, ensuring real-time availability and consistency across the distributed environment. Security mechanisms, user role management, and query optimization further enhance the system’s reliability and efficiency. Overall, this project provides a robust, scalable, and secure foundation for real-time stock trading analytics, showcasing how distributed database architecture can effectively handle geographically dispersed financial data in a modern, performance-driven environment.