Design Reasoning Document – Futures & Options Database

1. Introduction

This document explains the design, implementation, and optimization decisions made while building a scalable relational database for high-volume Futures & Options (F&O;) trading data.

2. Dataset Overview

The dataset used is the NSE Future and Options Dataset (3 Months) from Kaggle, consisting of approximately 2.53 million rows across 16 columns. It includes instrument details, expiry dates, strike prices, OHLC prices, volume, open interest, and timestamps spanning August to November 2019.

3. Database Schema Design

A normalized schema following Third Normal Form (3NF) was chosen to reduce redundancy and maintain data integrity. Separate tables were created for exchanges, instruments, expiries, and trades, allowing efficient joins and future extensibility for additional exchanges like BSE and MCX.

4. Data Ingestion and Validation

Data ingestion was performed using Python with Pandas and SQLAlchemy. The CSV was first loaded into a staging table using chunked inserts to balance memory usage and performance. Data was then normalized into dimension and fact tables. Row-count validation ensured that the final trades table matched the source dataset exactly.

5. Performance Optimization

To optimize performance on large time-series data, B-tree indexes were created on frequently filtered columns, and a BRIN index was applied on the trade_date column. The trades table was partitioned monthly based on trade_date, with a default partition to handle out-of-range dates safely.

6. Analytical Queries

Advanced analytical queries were implemented, including open interest change using window functions, rolling volatility calculations, option chain summaries by expiry and strike price, and recent-volume analysis optimized using indexes and partitions. All queries were executed successfully on the full dataset.

7. Performance Testing

Query performance was validated using EXPLAIN ANALYZE. The execution plan confirmed effective partition pruning, index-only scans, and fast execution times (~340 ms) for recent-date queries on a dataset exceeding 2.5 million rows.

8. Conclusion

The final solution demonstrates strong database design, SQL analytics, and performance engineering skills using real-world financial data. The system is scalable, optimized, and suitable for quantitative and data engineering use cases.