

Design choices

1. The source data is delivered as a single flat file where attributes such as symbol, expiry date, strike price, and option type are repeated across a large number of rows. Storing this data directly in a single table would result in significant redundancy and poor scalability.

To address this, the schema follows Third Normal Form (3NF) by separating the data into reference tables and a fact table:

- EXCHANGES – Exchange-level metadata (NSE, BSE, MCX)
- INSTRUMENTS – Tradable instruments (e.g., NIFTY, BANKNIFTY, GOLD)
- EXPIRIES – Contract-level definitions (expiry date, strike price, option type)
- TRADES – Time-series trading data (OHLC, volume, open interest)

This design ensures that all non-key attributes depend only on the primary key of their respective tables and eliminates transitive dependencies.

2. A dedicated EXPIRIES table is used to store expiry date, strike price, and option type instead of embedding these attributes directly in the trades table.

Becz

- The same strike combination appears across thousands of trade records
- It simplifies analytical queries related to options structures
- Separating contract definitions reduces duplication

3. The source CSV does not include an explicit exchange column, as it contains only NSE data. However, to meet the assignment requirement and ensure future extensibility, exchange information is modeled as a separate reference table (EXCHANGES) and linked to instruments.

Trades exchange information indirectly through the instrument relationship. This avoids storing exchange data redundantly at the trade level and preserves normalization.

Table Structure

EXCHANGES

- Stores a small, static set of exchange identifiers
- Uses a surrogate primary key (EXCHANGE_ID)
- Enforces uniqueness on EXCHANGE_NAME

INSTRUMENTS

- Represents tradable symbols and instrument types
- References the exchange table via a foreign key
- Indexed on symbol and exchange for fast lookups

EXPIRIES

- Stores contract definitions
- Indexed on expiry date and strike price
- Supports efficient option chain queries

TRADES

- Acts as the fact table
- Stores only numeric measures and foreign keys
- Designed for high-volume inserts and time-based filtering
- Uses appropriate numeric data types for prices, volume, and open interest

Optimization Approaches

The TRADES table is range-partitioned by TRADE_TIMESTAMP

BRIN is applied on the timestamp column. BRIN indexes are lightweight and highly effective for append-only, time-ordered data such as market trades.