**Zerodha DB Schema & Architecture Overview**

**✅ Objective**

Design a scalable trading platform like Zerodha that supports:

* Real-time trading (Buy/Sell)
* Watchlist & portfolio management
* Reporting (P&L, Holdings, etc.)
* Historical market data visualization
* Asynchronous heavy computation handling

**🧱 SQL DB Schemas**

**📁 Users**

CREATE TABLE Users (

user\_id VARCHAR PRIMARY KEY,

name VARCHAR,

email VARCHAR UNIQUE,

phone VARCHAR,

created\_at TIMESTAMP

);

**💰 Wallet**

CREATE TABLE Wallet (

user\_id VARCHAR REFERENCES Users(user\_id),

balance DECIMAL(12,2),

updated\_at TIMESTAMP

);

**📈 Orders**

CREATE TABLE Orders (

order\_id VARCHAR PRIMARY KEY,

user\_id VARCHAR REFERENCES Users(user\_id),

stock\_symbol VARCHAR,

type ENUM('BUY', 'SELL'),

price DECIMAL(10,2),

quantity INT,

status ENUM('PENDING', 'EXECUTED', 'FAILED'),

placed\_at TIMESTAMP

);

**🧾 PnLReports**

CREATE TABLE PnLReports (

report\_id SERIAL PRIMARY KEY,

user\_id VARCHAR,

period VARCHAR,

total\_profit DECIMAL,

generated\_at TIMESTAMP

);

**🟣 NoSQL Schema (MongoDB-like)**

**📄 UserWatchlists**

{

"userId": "U123",

"watchlist": [

{ "symbol": "TCS", "exchange": "NSE" },

{ "symbol": "INFY", "exchange": "NSE" }

],

"lastUpdated": "2024-12-10T10:45:00Z"

}

**🕐 TimeSeries Data (ClickHouse)**

| **Column** | **Type** |
| --- | --- |
| timestamp | DateTime |
| symbol | String |
| price | Float64 |
| volume | Int64 |
| open | Float64 |
| close | Float64 |
| high | Float64 |
| low | Float64 |

**🔐 ResultCache (Redis/Mongo)**

{

"userId": "U123",

"reportType": "P&L",

"period": "Q1-2024",

"data": {

"totalProfit": 12000,

"summary": { "INFY": 7000, "TCS": 5000 }

},

"cachedAt": "2024-12-20T11:30:00Z"

}

**✅ Summary Table**

| **Type** | **Entity** | **Fields** |
| --- | --- | --- |
| SQL | Users | user\_id, name, email, phone, created\_at |
| SQL | Wallet | user\_id, balance, updated\_at |
| SQL | Orders | order\_id, user\_id, stock\_symbol, type, price, quantity, status |
| SQL | PnLReports | report\_id, user\_id, period, total\_profit, generated\_at |
| NoSQL | Watchlists | userId, watchlist[], lastUpdated |
| TimeDB | MarketData | timestamp, symbol, price, volume, open, close, high, low |
| Cache | ResultCache | userId, reportType, period, data(JSON), cachedAt |