

Steps to build

1. Identify entities
2. Find attributes (columns)
3. Define entity relationship [eg: 1-1 , 1:N, M:N]
4. Visualize schema (Conceptual)
5. Convert to logical Schema (Database)
6. Check design with real problems , analyse fix and reiterate

Entity Relationship

Entity set value (n) -> 9

1. { Global Environmental variable }
2. { Brokerage info }
3. { identifying different accounts at each brokerage }
4. { stock demographics }
5. { stock price history }
6. { priority Archive price }
7. { priority Current price }
8. { Orders , showing asset }
9. { Transactions log }

Schema -> A map of your data - how entity (things) -> n = 9 in your system (algorithmic trading application) relate to each other.

Step 1 : Clarifying and identifying entities(n=9 entity-table)

N o	Entity	Description	Type
1	Global Environment Variables	Holds app-wide configurations like market hours, trading mode (live/simulated), etc.	Configuration
2	Brokerage Info	Contains metadata about supported brokers (Alpaca, Robinhood, etc.).	Static reference
3	Accounts	Each brokerage account (identified by user and broker).	User/Trading
4	Stock Demographics	Basic company details (sector, market cap, etc.).	Reference data
5	Stock Price History	Historical OHLCV data for each stock.	Time-series
6	Priority Archive Price	Archived “priority” calculated prices (e.g., derived analytics snapshots).	Derived

7	Priority Current Price	Current real-time computed metrics (live “priority” value per stock).	Live data
8	Orders	Buy/sell requests placed by accounts.	Trading
9	Transaction Log	Execution details of each order (partial fills, executions, etc.).	Audit/Record

Step 2: Entity Relationship

From	To	Relationship	Meaning
Global Environment Variables	(Referenced by all)	1 → Many	Used globally (not directly linked to users)
Brokerage Info	Accounts	1 → Many	One broker can have many user accounts
Accounts	Orders	1 → Many	Each account places many orders
Orders	Transactions	1 → Many	Each order results in one or more transactions
Accounts	Transactions	1 → Many	Transactions are logged under the account
Stock Demographics	Stock Price History	1 → Many	Each stock has many daily prices
Stock Demographics	Priority Archive Price	1 → Many	Each stock can have archived analytics
Stock Demographics	Priority Current Price	1 → 1	One current “priority” per stock
Orders	Stock Demographics	Many → 1	Each order targets one stock
Transactions	Stock Demographics	Many → 1	Each trade involves one stock

Step 3: Schema

(Top to Bottom)

GlobalEnv

↓

Brokerages —< **Accounts** —< **Orders** —< **Transactions**



1.The environment defines system-level config.

2.Brokerages contain accounts.

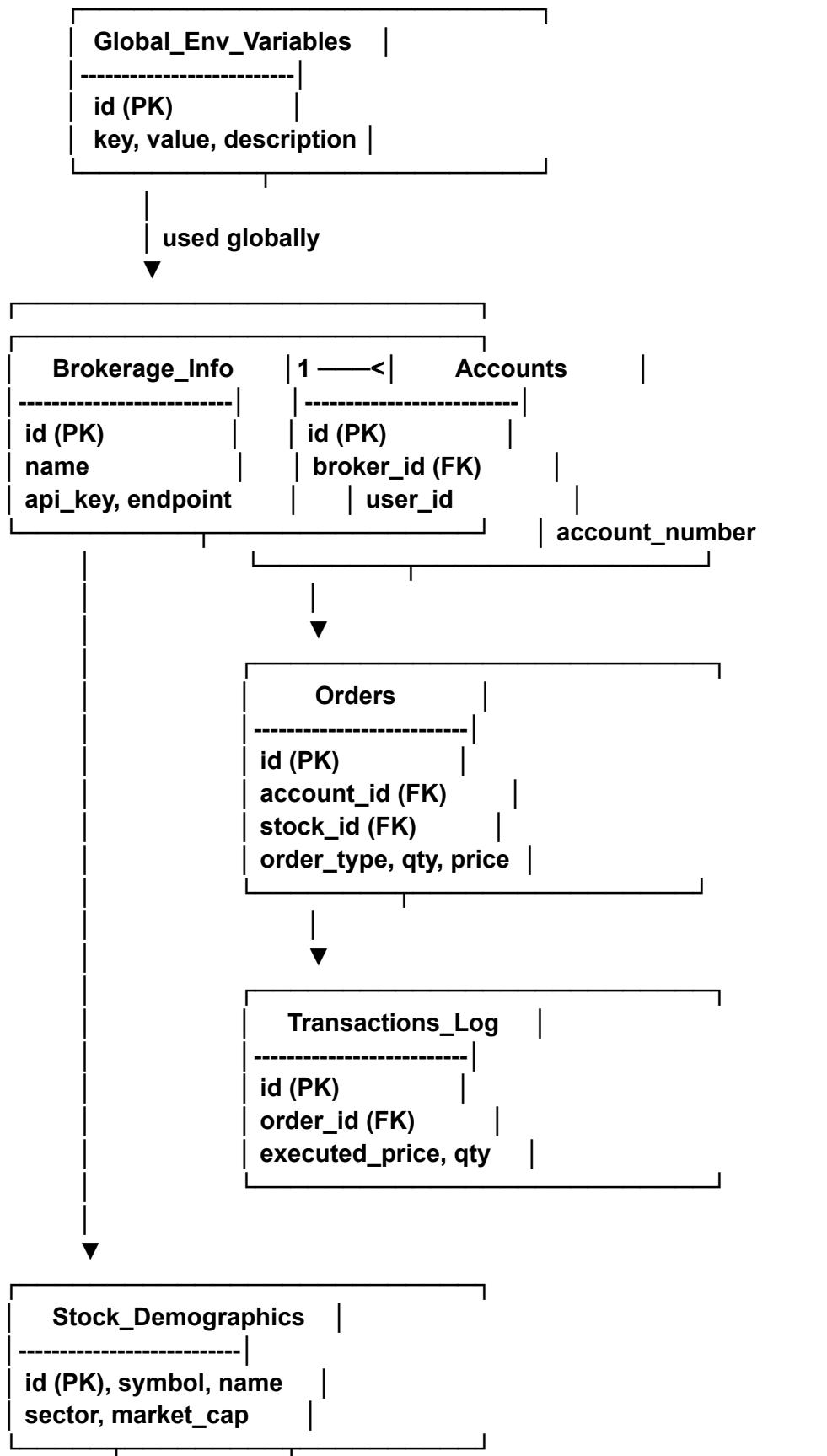
3.Each account places orders.

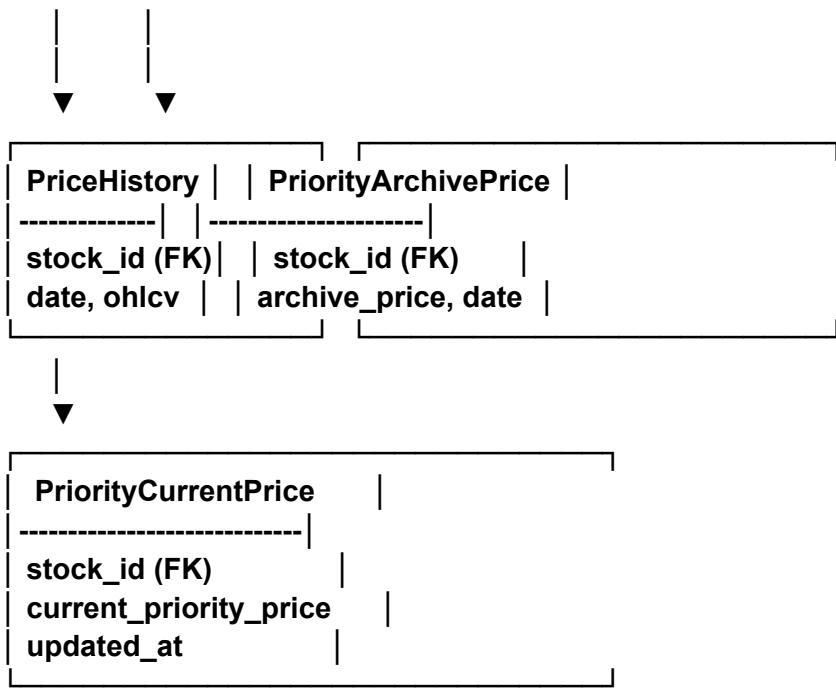
4.Each order results in transactions.

5.Each order targets a stock.

6.Stocks have demographics, historical, and priority data.

Step 4 : Entity Relationship Diagram





System-wide configs in `GlobalEnv` affect how brokers and prices are processed.

Brokerage_Info defines external APIs (Robinhood, Alpaca, etc.).

Each Account belongs to one broker and one user.

Orders originate from an account and target a Stock.

Transactions are the detailed execution log for each order.

Stock_Demographics is your static “master stock list.”

PriceHistory tracks daily prices.

PriorityArchivePrice stores old computed metrics (for analytics).

PriorityCurrentPrice stores live computed data (for dashboards)