

Logical Data Model & Data Dictionary

All example company names, trading venues, securities, strategies, and datasets used within this product are entirely fictional and are provided solely for educational and demonstration purposes.

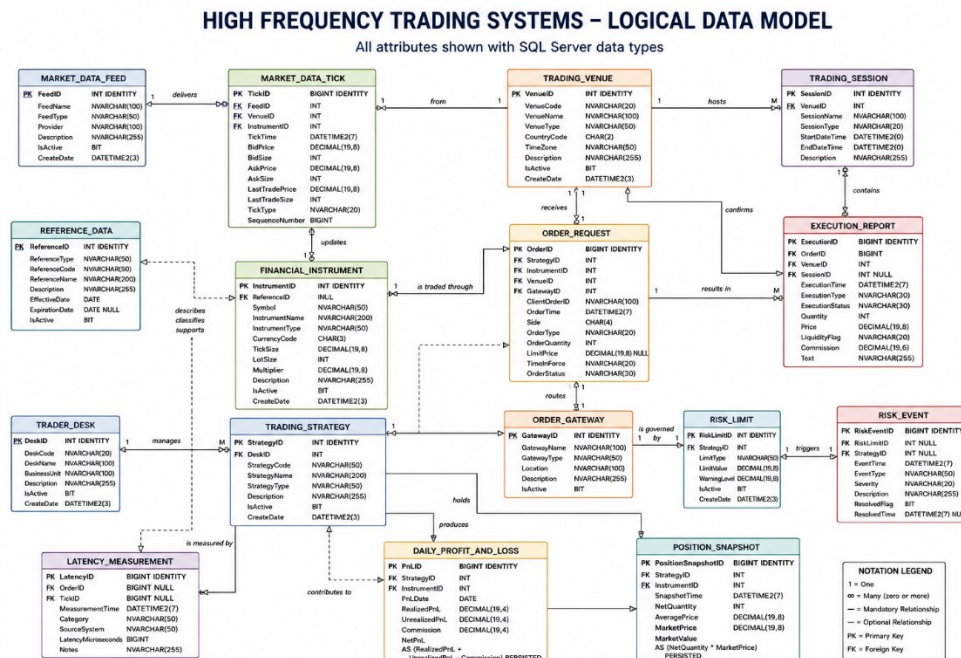
The material was created as original instructional content and is intended to provide a safe, practical, and understandable introduction to data structures and SQL techniques commonly used in high-frequency trading environments.

© Grumpy Old IT Guy Publishers / Angelo R. Bobak 2026.

All references to Microsoft technologies are used strictly for educational, instructional, commentary, and demonstration purposes under the principles of nominative fair use.

Any screenshots, product references, feature descriptions, or code examples referencing Microsoft technologies are intended solely to help users learn database design, SQL development, and related technical concepts.

High Frequency Trading Systems Logical Data Model



The logical High Frequency Trading (HFT) Systems Data Model represents the core operational, analytical, and risk-management components commonly found within modern electronic trading environments. The model was designed to demonstrate how market data, algorithmic trading

activity, order routing, executions, positions, risk controls, and performance measurements can be organized within a structured relational database environment using Microsoft SQL Server.

At the center of the model is the concept of the automated trading strategy. A trading strategy comprises the mathematical rules, algorithms, pricing logic, and execution behaviors that generate buy and sell orders in response to changing market conditions. Each trading strategy is managed by a trader desk or business unit, allowing organizations to group related strategies according to trading function, asset class, or business responsibility.

The model also contains a detailed representation of financial instruments and trading venues. Financial instruments include the securities traded, such as equities, exchange-traded funds, futures contracts, currencies, and derivative products. Trading venues represent the exchanges or electronic marketplaces where orders are routed and executed. By separating instruments from venues, the database can support multiple exchanges trading the same instrument while also allowing analysis of trading activity by market destination.

One of the most important operational components of any HFT environment is the market data feed. The Market Data Feed entity represents the source of incoming pricing information, while the Market Data Tick entity captures the actual stream of bid prices, ask prices, trade prices, and quote sizes arriving from the marketplace. These market data events form the foundation for trading strategies' decisions. Because high-frequency trading systems process enormous volumes of data in extremely short periods of time, the Market Data Tick entity was designed to support high-speed sequential processing and time-based analysis.

The Order Request entity represents orders generated by trading strategies and routed to trading venues through an order gateway. Each order contains information such as side, quantity, order type, limit price, time-in-force instructions, and current order status. The Order Gateway entity represents the infrastructure layer responsible for transmitting orders between internal trading systems and external exchanges or broker networks. This separation of responsibilities reflects the layered architecture commonly found in institutional electronic trading platforms.

Execution activity is captured through the Execution Report entity. Each execution report records the outcome of an order request, including fills, partial fills, cancellations, rejections, commissions, and execution prices. Since a single order may produce multiple execution events over time, the relationship between orders and executions supports detailed trade lifecycle tracking and transaction auditing. Execution reporting is essential for operational monitoring, trade reconciliation, compliance reporting, and profit-and-loss analysis.

The logical model also includes a Position Snapshot entity used to maintain current holdings and exposure information for each strategy and instrument combination. Position snapshots allow traders, risk managers, and analytical systems to monitor net positions, average costs, and market values in near real time. This information supports intraday exposure monitoring and helps ensure that strategies remain within acceptable trading limits.

Risk management is represented through the Risk Limit and Risk Event entities. Risk limits define operational constraints such as maximum order quantity, allowable position size,

acceptable latency thresholds, and maximum daily losses. When trading activity approaches or violates these thresholds, risk events may be generated to alert operators or automated supervisory systems. This portion of the model demonstrates how governance and operational controls can be integrated directly into a trading database environment.

Because speed is critical in high-frequency trading systems, the model also includes a Latency Measurement entity to capture performance metrics for market data processing, order routing, and execution response times. These measurements allow organizations to analyze system bottlenecks, optimize infrastructure performance, and monitor operational efficiency at microsecond-level granularity.

The Daily Profit and Loss Entity supports analytical and financial reporting by tracking realized profit and loss, unrealized profit and loss, commissions, and net profitability for each strategy-instrument combination. This structure allows firms to evaluate trading performance, compare the effectiveness of strategies, and monitor profitability trends over time.

Supporting entities such as Trading Session and Reference Data provide additional operational context and metadata management capabilities. Trading sessions allow activity to be grouped according to market hours, such as pre-market, regular session, or after-hours trading, while reference data provides classification and descriptive information used throughout the environment.

Overall, this logical data model demonstrates how a complex high-frequency trading environment can be decomposed into a structured relational design suitable for implementation within Microsoft SQL Server. The model provides a foundation for educational demonstrations, analytical reporting, performance monitoring, prototype trading platforms, and advanced SQL development exercises. It also illustrates the close relationship between transactional processing, real-time analytics, infrastructure monitoring, and risk management within modern electronic financial systems.

Logical Data Dictionary

High Frequency Trading Systems Data Model

Entity: Market_Data_Feed

Column Name	Data Type	Key	Description
FeedID	INT IDENTITY	PK	Unique identifier for each market data feed.
FeedName	NVARCHAR(100)		Name of the market data feed.
FeedType	NVARCHAR(50)		Type of feed, such as quote, trade, depth, or consolidated.
Provider	NVARCHAR(100)		Fictional provider or source of the feed.

Column Name	Data Type	Key	Description
Description	NVARCHAR(255)		Description of the feed.
IsActive	BIT		Indicates whether the feed is active.
CreateDate	DATETIME2(3)		Date and time the record was created.

Entity: Trading_Venue

Column Name	Data Type	Key	Description
VenueID	INT IDENTITY	PK	Unique identifier for the trading venue.
VenueCode	NVARCHAR(20)		Short code used to identify the venue.
VenueName	NVARCHAR(100)		Name of the exchange or trading venue.
VenueType	NVARCHAR(50)		Type of venue, such as exchange, ECN, or dark pool.
CountryCode	CHAR(2)		Two-character country code.
TimeZone	NVARCHAR(50)		Time zone used by the venue.
Description	NVARCHAR(255)		Description of the venue.
IsActive	BIT		Indicates whether the venue is active.
CreateDate	DATETIME2(3)		Date and time the venue record was created.

Entity: Financial_Instrument

Column Name	Data Type	Key	Description
InstrumentID	INT IDENTITY	PK	Unique identifier for the financial instrument.
ReferenceID	INT	FK	Links the instrument to reference data.
Symbol	NVARCHAR(50)		Trading symbol or ticker.
InstrumentName	NVARCHAR(200)		Full descriptive name of the instrument.
InstrumentType	NVARCHAR(50)		Type of instrument, such as equity, ETF, future, option, or FX pair.
CurrencyCode	CHAR(3)		Trading currency for the instrument.
TickSize	DECIMAL(19,8)		Minimum price movement allowed for the instrument.
LotSize	INT		Standard trading lot size.
Multiplier	DECIMAL(19,8)		Contract or pricing multiplier, if applicable.
Description	NVARCHAR(255)		Description of the instrument.
IsActive	BIT		Indicates whether the instrument is active.
CreateDate	DATETIME2(3)		Date and time the record was created.

Entity: Trader_Desk

Column Name	Data Type	Key	Description
DeskID	INT IDENTITY	PK	Unique identifier for the trading desk.
DeskCode	NVARCHAR(20)		Short desk code.
DeskName	NVARCHAR(100)		Name of the trading desk.
BusinessUnit	NVARCHAR(100)		Business unit responsible for the desk.
Description	NVARCHAR(255)		Description of the desk.
IsActive	BIT		Indicates whether the desk is active.
CreateDate	DATETIME2(3)		Date and time the record was created.

Entity: Trading_Strategy

Column Name	Data Type	Key	Description
StrategyID	INT IDENTITY	PK	Unique identifier for the trading strategy.
DeskID	INT	FK	Identifies the trading desk that manages the strategy.
StrategyCode	NVARCHAR(50)		Short code for the strategy.
StrategyName	NVARCHAR(200)		Full name of the strategy.
StrategyType	NVARCHAR(50)		Type of strategy, such as market making, momentum, or arbitrage.
Description	NVARCHAR(255)		Description of the strategy.
IsActive	BIT		Indicates whether the strategy is active.
CreateDate	DATETIME2(3)		Date and time the strategy record was created.

Entity: Market_Data_Tick

Column Name	Data Type	Key	Description
TickID	BIGINT IDENTITY	PK	Unique identifier for each market data tick.
FeedID	INT	FK	Identifies the market data feed that delivered the tick.
VenueID	INT	FK	Identifies the trading venue associated with the tick.
InstrumentID	INT	FK	Identifies the financial instrument being quoted or traded.
TickTime	DATETIME2(7)		High-precision timestamp for the market data event.
BidPrice	DECIMAL(19,8)		Best bid price at the time of the tick.
BidSize	INT		Quantity available at the bid price.
AskPrice	DECIMAL(19,8)		Best ask price at the time of the tick.
AskSize	INT		Quantity available at the ask price.
LastTradePrice	DECIMAL(19,8)		Last traded price, if available.

Column Name	Data Type	Key	Description
LastTradeSize	INT		Last traded quantity, if available.
TickType	NVARCHAR(20)		Type of tick, such as quote, trade, or book update.
SequenceNumber	BIGINT		Feed sequence number used for ordering and validation.

Entity: Order_Gateway

Column Name	Data Type	Key	Description
GatewayID	INT IDENTITY	PK	Unique identifier for the order gateway.
GatewayName	NVARCHAR(100)		Name of the gateway.
GatewayType	NVARCHAR(50)		Type of gateway, such as internal router or external FIX gateway.
Location	NVARCHAR(100)		Logical or physical location of the gateway.
Description	NVARCHAR(255)		Description of the gateway.
IsActive	BIT		Indicates whether the gateway is active.

Entity: Order_Request

Column Name	Data Type	Key	Description
OrderID	BIGINT IDENTITY	PK	Unique identifier for the order request.
StrategyID	INT	FK	Identifies the strategy that generated the order.
InstrumentID	INT	FK	Identifies the instrument being traded.
VenueID	INT	FK	Identifies the venue receiving the order.
GatewayID	INT	FK	Identifies the gateway used to route the order.
ClientOrderID	NVARCHAR(100)		External or internal client order identifier.
OrderTime	DATETIME2(7)		High-precision timestamp when the order was created.
Side	CHAR(4)		Buy or sell indicator.
OrderType	NVARCHAR(20)		Type of order, such as market, limit, or stop.
OrderQuantity	INT		Number of shares, contracts, or units ordered.
LimitPrice	DECIMAL(19,8)		Limit price for limit orders.
TimeInForce	NVARCHAR(20)		Instruction for how long the order remains active.
OrderStatus	NVARCHAR(30)		Current status of the order.

Entity: Execution_Report

Column Name	Data Type	Key	Description
ExecutionID	BIGINT IDENTITY	PK	Unique identifier for the execution report.
OrderID	BIGINT	FK	Identifies the order associated with the execution.
VenueID	INT	FK	Identifies the venue where the execution occurred.
SessionID	INT NULL	FK	Identifies the trading session, if available.
ExecutionTime	DATETIME2(7)		High-precision timestamp of the execution event.
ExecutionType	NVARCHAR(30)		Type of execution event, such as fill, partial fill, cancel, or reject.
ExecutionStatus	NVARCHAR(30)		Final or current execution status.
Quantity	INT		Executed quantity.
Price	DECIMAL(19,8)		Execution price.
LiquidityFlag	NVARCHAR(20)		Indicates whether liquidity was added or removed.
Commission	DECIMAL(19,6)		Commission charged for the execution.
Text	NVARCHAR(255)		Optional notes or status message.

Entity: Position_Snapshot

Column Name	Data Type	Key	Description
PositionSnapshotID	BIGINT IDENTITY	PK	Unique identifier for the position snapshot.
StrategyID	INT	FK	Identifies the strategy holding the position.
InstrumentID	INT	FK	Identifies the instrument held.
SnapshotTime	DATETIME2(7)		Timestamp when the position was captured.
NetQuantity	INT		Net quantity held by the strategy.
AveragePrice	DECIMAL(19,8)		Average price of the position.
MarketPrice	DECIMAL(19,8)		Current market price used for valuation.
MarketValue	DECIMAL(19,4)		Calculated market value of the position.

Entity: Risk_Limit

Column Name	Data Type	Key	Description
RiskLimitID	INT IDENTITY	PK	Unique identifier for the risk limit.
StrategyID	INT	FK	Identifies the strategy governed by the limit.
LimitType	NVARCHAR(50)		Type of limit, such as order quantity, position, loss, or latency.
LimitValue	DECIMAL(19,8)		Numeric value of the limit.
WarningLevel	DECIMAL(19,8)		Threshold used to generate a warning before breach.
IsActive	BIT		Indicates whether the risk limit is active.
CreateDate	DATETIME2(3)		Date and time the limit record was created.

Entity: Risk_Event

Column Name	Data Type	Key	Description
RiskEventID	BIGINT IDENTITY	PK	Unique identifier for the risk event.
RiskLimitID	INT NULL	FK	Identifies the risk limit that triggered the event.
StrategyID	INT NULL	FK	Identifies the strategy associated with the event.
EventTime	DATETIME2(7)		Timestamp when the risk event occurred.
EventType	NVARCHAR(50)		Type of event, such as warning, breach, or system alert.
Severity	NVARCHAR(20)		Severity level of the event.
Description	NVARCHAR(255)		Description of the risk event.
ResolvedFlag	BIT		Indicates whether the event has been resolved.
ResolvedTime	DATETIME2(7) NULL		Timestamp when the event was resolved.

Entity: Latency_Measurement

Column Name	Data Type	Key	Description
LatencyID	BIGINT IDENTITY	PK	Unique identifier for the latency measurement.
OrderID	BIGINT NULL	FK	Identifies the order associated with the measurement, if applicable.
TickID	BIGINT NULL	FK	Identifies the market data tick associated with the measurement, if applicable.
MeasurementTime	DATETIME2(7)		Timestamp when latency was measured.
Category	NVARCHAR(50)		Type of latency being measured.
SourceSystem	NVARCHAR(50)		System where the measurement originated.
LatencyMicroseconds	BIGINT		Latency value measured in microseconds.
Notes	NVARCHAR(255)		Optional measurement notes.

Entity: Daily_Profit_And_Loss

Column Name	Data Type	Key	Description
PnLID	BIGINT IDENTITY	PK	Unique identifier for the profit and loss record.
StrategyID	INT	FK	Identifies the strategy associated with the PnL.
InstrumentID	INT	FK	Identifies the instrument associated with the PnL.
PnLDate	DATE		Business date for the PnL record.
RealizedPnL	DECIMAL(19,4)		Profit or loss from closed trades.
UnrealizedPnL	DECIMAL(19,4)		Profit or loss from open positions.

Column Name	Data Type	Key	Description
Commission	DECIMAL(19,4)		Trading commission amount.
NetPnL	DECIMAL(19,4)		Net profit or loss after commissions.

Entity: Trading_Session

Column Name	Data Type	Key	Description
SessionID	INT IDENTITY	PK	Unique identifier for the trading session.
VenueID	INT	FK	Identifies the venue where the session occurs.
SessionName	NVARCHAR(100)		Name of the session.
SessionType	NVARCHAR(20)		Type of session, such as pre-market, regular, or after-hours.
StartDateTime	DATETIME2(0)		Start date and time of the trading session.
EndDateTime	DATETIME2(0)		End date and time of the trading session.
Description	NVARCHAR(255)		Description of the session.

Entity: Reference_Data

Column Name	Data Type	Key	Description
ReferenceID	INT IDENTITY	PK	Unique identifier for the reference data record.
ReferenceType	NVARCHAR(50)		Category of reference data.
ReferenceCode	NVARCHAR(50)		Code value used by the system.
ReferenceName	NVARCHAR(200)		Descriptive name of the reference value.
Description	NVARCHAR(255)		Description of the reference data value.
EffectiveDate	DATE		Date the reference value becomes effective.
ExpirationDate	DATE NULL		Date the reference value expires, if applicable.
IsActive	BIT		Indicates whether the reference data value is active.