

Conceptual Data Model & Data Dictionary

Grumpy Old IT Guy Publishers and **Angelo R. Bobak** are independent educational content providers and are not affiliated with, endorsed by, sponsored by, or associated with Microsoft Corporation.

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.

The High Frequency Trading Systems Data Model and SQL Server Toolkit is a professionally structured educational package designed for database developers, data architects, financial technologists, quantitative analysts, and students interested in the architecture of modern electronic trading systems. The product introduces the foundational concepts behind high-frequency trading environments and demonstrates how complex trading activity can be represented using structured relational database models implemented in Microsoft SQL Server.

High Frequency Trading, often referred to as HFT, is a specialized form of electronic trading in which automated systems process market data, generate trading decisions, route orders, and capture executions at extremely high speeds. These systems are designed to react to rapidly changing market conditions in fractions of a second while continuously monitoring prices, liquidity, latency, risk exposure, and trading profitability. Behind every successful HFT platform is a carefully engineered database structure that efficiently and accurately stores and analyzes massive amounts of transactional and market data.

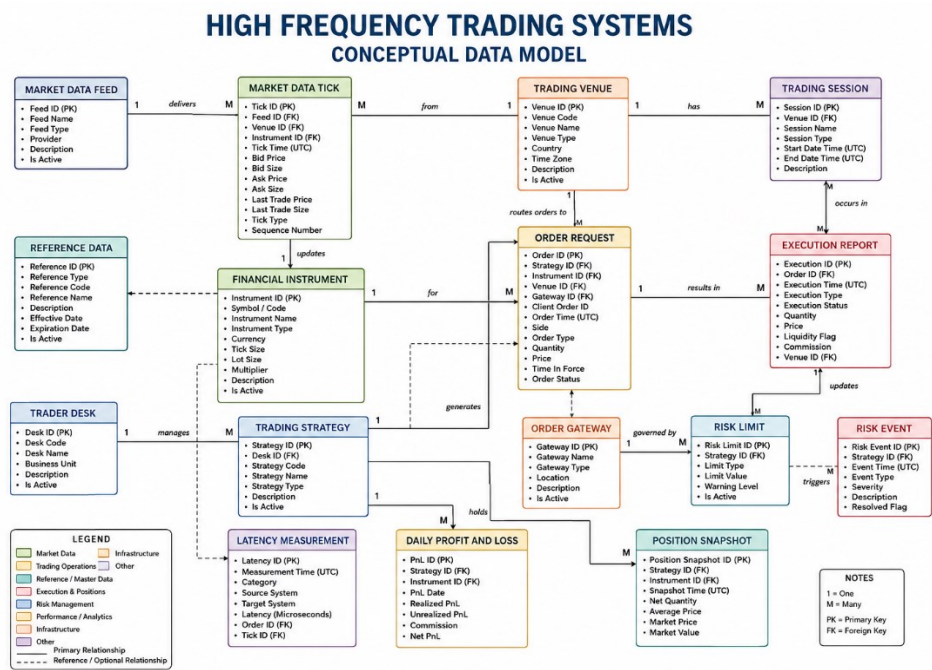
This product provides a complete educational framework that demonstrates how such a trading environment can be modeled using standard relational database techniques and Transact-SQL programming practices. The material has been designed for instructional purposes and is suitable for classroom demonstrations, technical workshops, independent learning, prototype development, and proof-of-concept implementations.

The package includes a fully documented conceptual business model describing the major functional components commonly found in electronic trading environments, including trading venues, financial instruments, market data feeds, trading strategies, order management, executions, positions, latency monitoring, risk management, and profit-and-loss reporting. These business concepts are then translated into detailed logical and physical data models that illustrate how relationships between entities can be implemented within a normalized SQL Server database design.

In addition to the conceptual and logical models, the product contains detailed data dictionaries that document each entity, attribute, relationship, and business rule used throughout the system. These dictionaries help readers understand the purpose of each table and column while also serving as examples of professional metadata documentation standards commonly used in enterprise data management initiatives.

Below is the conceptual data model design for this database.

Conceptual Data Model



Entity Data Dictionary

Below are the descriptions for each entity.

Entity	Description
Trading Venue	Represents an exchange, electronic communication network, alternative trading system, or other market destination where securities may be quoted, traded, or routed.
Financial Instrument	Represents a tradable security or asset, such as an equity, ETF, futures contract, option, currency pair, or other market product.
Trader Desk	Represents the business unit, desk, or trading group responsible for one or more trading strategies.
Trading Strategy	Represents an automated or semi-automated trading approach that generates orders based on market data, pricing signals, statistical rules, or risk controls.

Entity	Description
Market Data Tick	Represents a single market data event, such as a bid update, ask update, trade print, quote change, or price/size change received from a market data feed.
Order Request	Represents an order instruction generated by a trading strategy and submitted to a trading venue or order gateway.
Execution Report	Represents the outcome or status update of an order, including fills, partial fills, cancellations, rejections, or acknowledgments.
Latency Measurement	Represents timing measurements used to monitor system performance, including market data latency, order routing latency, and order-to-execution latency.
Position Snapshot	Represents the quantity and market value of holdings for a trading strategy and financial instrument at a specific point in time.
Risk Limit	Represents predefined trading controls used to restrict order size, net position, daily loss, latency tolerance, or other operational limits.
Daily Profit and Loss	Represents daily financial performance for a strategy and instrument, including realized profit/loss, unrealized profit/loss, commissions, and net profit/loss.
Market Data Feed	Represents the source or channel through which real-time or delayed pricing, quote, and trade information is received.
Order Gateway	Represents the system component responsible for accepting orders from strategies and routing them to the appropriate trading venue.
Reference Data	Represents descriptive and classification information used to identify instruments, venues, currencies, calendars, and other supporting trading data.
Trading Session	Represents a defined market period during which trading activity occurs, such as pre-market, regular session, after-hours, or overnight trading.
Risk Event	Represents an exception, breach, warning, or alert generated when trading activity approaches or violates a defined risk threshold.

Entity Relationship Data Dictionary

Below are the basic verb phrases that describe the business relationships between entities:

Parent Entity	Association Rule	Child Entity
Market Data Feed	delivers	Market Data Tick
Trading Venue	publishes or receives	Market Data Tick
Financial Instrument	is quoted by	Market Data Tick
Trading Venue	hosts	Trading Session
Trading Session	contains	Execution Report
Trader Desk	manages	Trading Strategy
Trading Strategy	generates	Order Request

Parent Entity	Association Rule	Child Entity
Financial Instrument	is traded through	Order Request
Trading Venue	receives	Order Request
Order Gateway	routes	Order Request
Order Request	results in	Execution Report
Trading Venue	confirms	Execution Report
Trading Strategy	is governed by	Risk Limit
Risk Limit	triggers	Risk Event
Trading Strategy	generates	Risk Event
Trading Strategy	holds	Position Snapshot
Financial Instrument	is held in	Position Snapshot
Trading Strategy	produces	Daily Profit and Loss
Financial Instrument	contributes to	Daily Profit and Loss
Order Request	is measured by	Latency Measurement
Market Data Tick	is measured by	Latency Measurement
Reference Data	describes	Financial Instrument
Reference Data	classifies	Trading Venue
Reference Data	supports	Trading Strategy

Disclaimer

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.