The ultimate guide to time series databases

On this page

- **What is time series data?**
- What is a time series database?
- Typical use cases
- How to choose the right time series database
- How KX powers time series applications
- Customer case studies
- Frequently Asked Questions (FAQ)

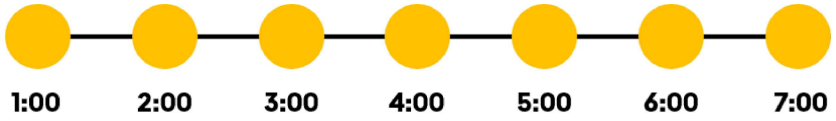
Time series databases are revolutionizing how we handle and analyze timestamped data. Whether working with IoT sensors, financial data, or monitoring system performance in real-time, a time series database provides the speed, scalability, and efficiency you need to maximize your data.

At KX, we offer unparalleled capabilities for managing vast amounts of time series data, ensuring rapid retrieval and insightful analysis to drive informed decision-making.

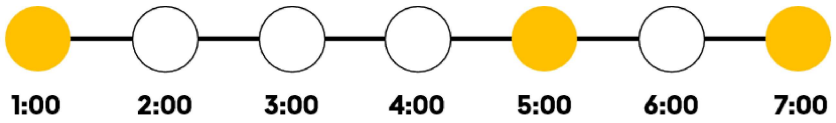
What is time series data?

Time series data is a collection of data points gathered or recorded at successive time intervals. It is used to observe and analyze changes over time, helping to identify trends, patterns, and variations.

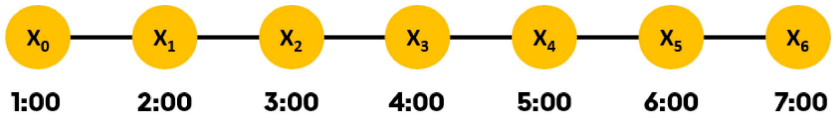
Qomplx LLC, Exhibit 2029
Page 2029 - 1
IPR2026-00184, Microsoft Corp. v. Qomplx LLC



Many modeling and analytics approaches require evenly spaced intervals between timestamped data points; however, real-world data often has irregular intervals, resulting in gaps or missing values, which necessitate imputation methods to address.

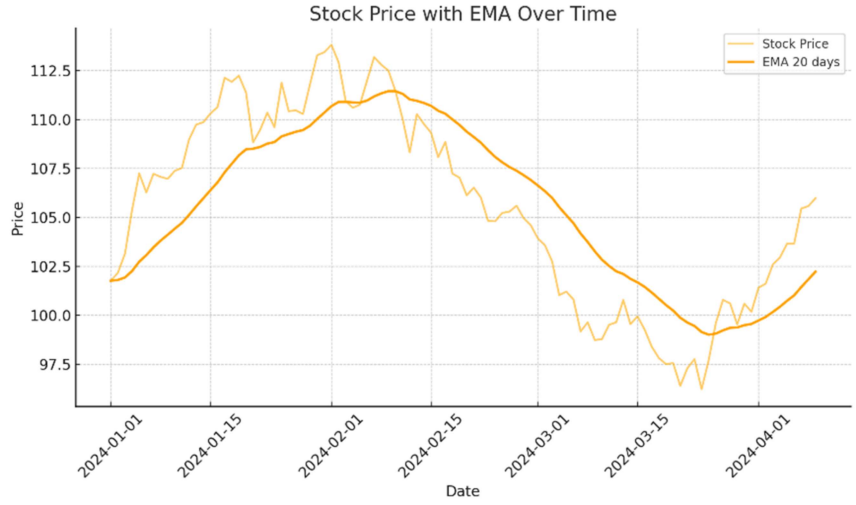


Each timestamped data point is associated with a corresponding data value.



These values can then be visualized in temporal (time-based) charts such as the line chart shown below.

Analytic models can be built from these raw time series data points and used for advanced decision-making. A simple example is a smoothing algorithm for “Exponential Moving Average” (EMA), as shown below.



What is a time series database?

A time series database is optimized to store, retrieve, and manage timestamped data points. These databases are designed to handle high ingestion and query throughput for applications that track changes over time, such as stock market analysis, system monitoring, and IoT data.

Unlike traditional relational databases, time series databases offer:

1. Time-based indexing for fast access to historical and real-time data.
2. Optimized compression for efficient storage.
3. Scalable performance for handling vast datasets.

This makes them the ideal choice for time-series data in industries that require frequent, time-based queries, such as finance, healthcare, and infrastructure monitoring. Conversely, traditional relational databases such as MySQL and PostgreSQL can handle structured data well but struggle to manage timestamped data efficiently.

Key differences:

Feature	Time series databases	Relational databases
Data model	Time-based, optimized for timestamped data	Structured, rows, & columns
Write throughput	High, optimized for time-series data	Moderate, optimized for transaction-based data
Query performance	Fast for time-based queries	Slower, complex queries for time data

Storage efficiency	Compressed, supports downsampling	Traditional, less optimized for historical data

A columnar time series database, such as kdb+, is designed to optimize the storage, retrieval, and analysis of timestamped data. Columnar databases became popular in the 2000s as a solution to the limitations of row-based databases in handling large amounts of data. Unlike traditional row-based databases, where entire rows of data are stored together, columnar databases store data by columns. This storage approach is particularly beneficial for time-series data, as it enables highly efficient querying and data compression.

Here's how it works:

1. **Columnar storage model**: In a traditional row-based database, a row represents a complete record (e.g. a single transaction or event), and all its fields are stored together. In contrast, a columnar database stores each data field (or column) separately.

   For example, consider a dataset with time, temperature, and humidity readings. A columnar database would store all time values in one column, all temperature values in another column, and all humidity values in yet another column. This layout allows for more efficient storage and retrieval, especially when you frequently query specific columns rather than the entire dataset.

2. **Efficient querying**: In time series analysis, most queries target specific columns (e.g. "Show me the temperature readings for the past hour"). In a columnar database, only the relevant column (temperature) needs to be read, which reduces I/O operations and increases query speed. In a row-based system, the database must read entire rows even if only one field is needed.

   Columnar storage also enables faster aggregations and filtering because the data is stored sequentially within each column, allowing for highly efficient scan operations.

3. **Data compression**: Columnar databases typically achieve high compression rates because data within each column is often similar in structure and value. For example, temperature readings over time may only vary slightly from one entry to the next. This

Qomplx LLC, Exhibit 2029
Page 2029 - 4
IPR2026-00184, Microsoft Corp. v. Qomplx LLC

makes compression techniques like run-length encoding (RLE), delta encoding, or dictionary encoding highly effective.

4. **Time series optimization**: In a time series database, the time column (or timestamp) is usually the primary index, and other values (like sensor readings) are stored alongside the time index.

   Time series data is inherently ordered by time; columnar databases use this order to optimize data storage and retrieval. For example, downsampling older data (storing fewer details for older time periods) can help manage the volume of data over time without losing the overall trends.

   Many time series databases also implement retention policies to automatically remove or archive data after a specified period, optimizing storage for high-frequency, time-sensitive applications.

5. **In-memory processing**: Some columnar time series databases, such as kdb+, utilize in-memory storage for real-time data, enabling high-speed processing. This enables real-time analytics on streaming data, which is essential in applications such as high-frequency trading or IoT monitoring, where latency is critical.
6. **Efficient aggregations and window functions**: Time series analysis often involves aggregating data over time windows (e.g. computing the average temperature over the past 10 minutes). Columnar databases are optimized for these aggregations because they can scan large blocks of similar data within each column and directly apply the aggregation function (like sum, average, or max) on compressed data to accelerate calculations.

Typical use cases

Time series databases are crucial in industries that rely on time-based data.

This includes:

1. **Financial data and stock market analysis:** Financial institutions utilize time series databases, such as kdb+, to store and analyze high-frequency data from stock markets. The ability to store massive datasets and quickly retrieve real-time data helps traders and analysts gain insights and make split-second decisions.
2. **IoT data collection**: The Internet of Things (IoT) collects timestamped data from thousands of sensors, devices, and systems. Time series databases like kdb+ efficiently handle large-scale data ingestion, providing real-time analytics and long-term storage.
3. **Industrial monitoring**: Manufacturers and industrial operations rely on time series databases to track equipment performance and optimize production lines. With time-based queries, companies can analyze historical trends and predict potential equipment failures, enabling preventive maintenance.

4. **Aerospace & defense**: Essential for tracking real-time sensor information, such as speed, altitude, and temperature, across aircraft and defense systems. It supports predictive maintenance by monitoring performance trends, reducing downtime, and enhancing safety. Additionally, it aids in flight data analysis and mission planning, improving operational efficiency and decision-making.
5. **DevOps and system monitoring**: kdb+ is a leading choice for system monitoring in cloud and containerized environments. It collects real-time metrics from applications and infrastructure, providing insight into system health and performance.

How to choose the right time series database

Choosing the right time series database depends on your specific use case, data requirements, and the scale of your application.

Here are some considerations:

1. **Performance & scalability**: Performance and scalability are critical because they determine how well a system handles increasing workloads and data demands. High performance ensures that queries and operations are executed quickly, even with large datasets, leading to efficient application functionality. Scalability allows the system to grow seamlessly as data volume and user demand increase, ensuring that the system remains responsive and reliable without significant infrastructure changes or degradation in performance.
2. **Data volume**: Data volume directly impacts a system's scalability, performance, and storage requirements. Large volumes of data demand efficient processing, querying, and storage solutions to ensure smooth operation and fast response times. Managing high data volumes effectively enables businesses to gain deeper insights, make data-driven decisions, and maintain a competitive edge even as data volumes continue to grow.
3. **Cost**: Considerations should be made for maintenance, scalability, and performance. While some open-source databases advertise low upfront costs, they often have higher support and maintenance costs. Additionally, one should consider the level of performance they are getting for a set cost and how many tools are required in the overall stack.
4. **Developer experience**: A good developer experience in a time series database streamlines adoption, allowing developers to manage, query, and analyze complex data quickly. With intuitive APIs, clear documentation, and strong tooling, developers can focus on building solutions quickly and efficiently. This fosters faster innovation, reduces friction, and enhances the overall quality and performance of applications using the database.

How KX powers time series applications

kdb+ is a high-performance time series database that combines temporal data (time series) with metadata (relational) for contextual insights. The columnar design of kdb+ offers greater speed and efficiency than typical relational databases, making it ideal for real-time analysis of streaming and historical data

Insights Enterprise is a fully integrated, scalable analytics platform for time-series data analysis. It builds upon the capabilities of Insights SDK by offering a turnkey solution that includes a management console, low/no-code query experience, and real-time dashboard visualizations

Insights SDK builds upon kdb+, providing developers with a modular toolkit for creating custom time-series analytics applications. It provides the highest level of flexibility and control over the data/analytics pipeline, enabling developers to architect, implement, deploy, and manage distributed analytics systems

KX is independently benchmarked as the world's fastest time series database and analytics engine. Its columnar design, in-memory architecture, 'peach' parallel processing, and multithreaded data loading ensure fast analytics on large-scale datasets both in motion and at rest.

Advantages:

1. **Performance and efficiency**: Our solutions are designed specifically for time series analytics at the data source, eliminating data movement. This is particularly beneficial for applications requiring immediate insights, such as financial trading and risk management.
2. **Integration and extensibility**: Our solutions offer a cloud-first, multi-vertical streaming analytics platform that combines temporal data (time series) with metadata (relational) for contextual insights. It is extensible with other languages like Python and SQL, making it versatile for developers and data engineers.
3. **Industry validation**: Our solutions are independently validated by industry benchmarks, such as STAC, which recognized kdb+ as the clear winner in 15 out of 17 performance tests for financial time-series data.
4. **Scalability and flexibility**: Our solutions are deployable on the cloud, at the edge, or on-premises. Full support for Docker and Kubernetes enables businesses to scale effortlessly and integrate with existing infrastructure.
5. **Comprehensive portfolio**: KX offers a range of solutions built on the architecture of kdb+ and q, including Insights Portfolio, PyKX,

and KDB.AI. These cater to various needs, from real-time data management and analytics to machine learning operations and AI applications.

6. **Proven track record**: KX has a long history of powering some of the world's most data-intensive organizations across industries, including finance, energy, telecommunications, medicine, and Formula One.
7. **Developer friendly**: kdb+ utilizes the q language, SQL, and PyKX for efficient querying, data manipulation, and integration with external systems. Each offers unique benefits that enhance the performance and usability of kdb+ for time series data and high-performance analytics.

Customer case studies

Bank of America

Bank of America faced significant challenges with its existing data systems, primarily a lack of flexibility in processing and analyzing large volumes of structured data. The financial institution needed a solution that could not only scale with its data growth but also deliver faster queries and enhanced analytics.

The lack of flexibility in their existing infrastructure limited the bank's ability to perform real-time analytics, anomaly detection, and predictive analytics on their vast data sets. They needed a solution that could handle large data volumes and deliver high-performance analytics to support decision-making across various business units.

Bank of America turned to KX for its powerful real-time data platform, capable of ingesting over 1TB of data daily and accessing up to 500TB of historical data. By leveraging its time series capabilities, low-latency queries, and q language, they achieved enhanced anomaly detection, predictive analytics, algorithmic automation, asset monitoring, and more.

The bank chose KX over competitors like Tigerdata (Timescale) and MongoDB due to its unique combination of speed, flexibility, and scalability, achieving ROI within 12-18 months.

Internet software and services company

A large enterprise-scale internet and services company faced significant difficulty scaling its existing data infrastructure to meet growing business demands. Their previous systems struggled to handle the increasing data volumes and processing requirements, creating bottlenecks that threatened operational efficiency. With daily data ingestion reaching up to 10 GB and historical data stores ranging from 500 TB to 1 PB, the company needed a solution that could

manage both real-time streaming data and big data processing workloads simultaneously.

Before selecting KX, the company conducted a thorough evaluation of alternative solutions, including MongoDB and Oracle. These platforms failed to meet the company's stringent requirements for low-latency queries and scalable data processing. The KX deployment delivered substantial business value, generating "millions of dollars" in returns for the organization and a return on investment within 6-12 months, demonstrating the platform's immediate impact on operational efficiency and business performance.

Fortune 500 Bank

Before implementing KX, a Fortune 500 bank faced three critical challenges that threatened its operational efficiency and market position.

- The company's existing data infrastructure lacked the flexibility needed to adapt to rapidly changing market conditions and diverse data requirements
- Scaling their systems to handle increasing data volumes proved increasingly difficult, creating bottlenecks that impacted business operations
- Slow development cycles hindered the company's ability to deploy new analytical capabilities and respond quickly to market opportunities.

By implementing KX, the bank was able to address all three pain points, achieving a return on investment within 6-12 months. With daily data ingestion exceeding 1 TB and historical data stores ranging from 1 PB to 10 PB, the implementation was described as "Priceless," reflecting the transformative impact on operations.

Benefits of q language

q is the native language of kdb+, designed specifically for handling time series data. It is a concise, expressive, and highly efficient array-based programming language that works seamlessly with the underlying architecture of kdb+.

Here are the key benefits:

1. **High performance**: q is designed for speed, particularly when working with large time series datasets. It leverages kdb+'s in-memory capabilities and columnar storage to enable real-time querying and data manipulation.q excels in environments where fast data ingestion, querying, and real-time analytics are critical, such as high-frequency trading.

Qomplx LLC, Exhibit 2029
Page 2029 - 9
IPR2026-00184, Microsoft Corp. v. Qomplx LLC

2. **Concise syntax**: q's syntax is minimalistic and expressive, allowing developers to write complex queries and operations in a few lines of code. Its terseness also makes it easier to work with large-scale data and perform rapid analytics without long, verbose code.
3. **Optimized for time series data**: q is particularly well-suited for operations involving timestamped data, such as analyzing and segmenting time windows, aggregating data over time, and performing analytics on time series trends. It has built-in functions for date, time, and timestamp manipulation, making it ideal for applications in finance, IoT, and system monitoring.
4. Built-In functions for complex analytics: q offers a rich set of built-in functions, such as statistical operations, joins, aggregations, and window functions, optimized for working with arrays and time series data, eliminating the need for external libraries or complex query patterns, providing a streamlined environment for conducting complex data analysis.
5. **Real-time analytics**: q is designed for high-frequency, low-latency environments, making it ideal for real-time analytics where fast response times are essential. This is particularly beneficial in scenarios like trading, where data needs to be processed in milliseconds.

Benefits of PyKX

PyKX enables developers to work with kdb+ using Python. It bridges the gap between the performance and power of kdb+ and the flexibility and popularity of Python.

Here are the key benefits of using PyKX:

1. **Python integration**: PyKX enables Python developers to access the power of kdb+ without needing to learn the q language. This makes kdb+ more accessible to a broader audience, especially for those already familiar with Python's rich data science and machine learning libraries.
2. **Access to Python ecosystem**: By using PyKX, kdb+ users can leverage Python's extensive libraries for data analysis, visualization, and machine learning. For example, you can use pandas for data manipulation, matplotlib for data visualization, or scikit-learn for machine learning while still taking advantage of kdb+'s high-performance backend.
3. **Efficient data transfer**: PyKX ensures efficient data transfer between Python and kdb+. Large datasets can be queried in q and then brought into Python for further analysis without significant performance overhead, which is critical when dealing with time-sensitive data.This efficiency makes it suitable for real-time analytics use cases where the results of kdb+ queries need to be processed or visualized in Python.

Qomplx LLC, Exhibit 2029
Page 2029 - 10
IPR2026-00184, Microsoft Corp. v. Qomplx LLC

4. **Data science and machine learning integration**: By enabling kdb+ to integrate with popular Python-based data science tools, PyKX creates new possibilities for machine learning and advanced analytics on time series data stored in kdb+.
5. **Rapid prototyping**: Python's versatility allows for rapid prototyping of analytical applications. PyKX users can quickly build prototypes and proof-of-concept applications using kdb+ as a powerful backend for real-time data processing.

Frequently Asked Questions (FAQ)

What is time series data

Time series data is a collection of data points gathered or recorded at successive time intervals. It is used to observe and analyze changes over time, helping to identify trends, patterns, and variations.

What is a time series database?

A time series database is optimized to store and query timestamped data. This is crucial for applications that require tracking changes over time, such as financial markets, IoT, and system performance monitoring.

How does a time series database differ from a relational database?

Time series databases are optimized for timestamped data and provide better performance for time-based queries. They also offer features like data compression and downsampling, unlike traditional relational databases.

How do columnar time series databases work?

Columnar time series databases store data by columns rather than rows, allowing for highly efficient querying and data compression. This storage approach is particularly beneficial for time series data because it allows for more efficient storage and retrieval.

How do time series databases handle data retention and optimization?

Time series databases often implement retention policies to automatically move data after a specified period. For example, in kdb+, data is moved from the real-time database (RDB) to the intraday database (IDB) and finally the historical database (HDB).

What are the best use cases for time series databases?

Common use cases include financial data analysis, IoT sensor data, system monitoring in DevOps, and industrial equipment monitoring.

Demo the world's fastest database for vector, time-series, and real-time analytics

Start your journey to becoming an AI-first enterprise with 100x* more performant data and MLOps pipelines.

Process data at unmatched speed and scale

Build high-performance data-driven applications

Turbocharge analytics tools in the cloud, on premise, or at the edge

*Based on time-series queries running in real-world use cases on customer environments.

Book a demo with an expert

First Name *

Last Name *

Company *

Job Title *

Business Telephone *

(201) 555-0123

Business Email *

Industry *

Select Industry

Country *

Select Country

How can KX help you? *

Tell us about your needs or goals–we're here to help!

0 of 255 max characters

How did you hear about us? *

Eg. Google Search, Word of Mouth, Friend/

0 of 255 max characters

By submitting this form, you will also receive sales and/or marketing communications on KX products, services, news and events. You can unsubscribe from receiving communications by visiting our Privacy Policy. You can find further information on how we collect and use your personal data in our Privacy Policy.

Book a demo

©2026 KX. All Rights Reserved. KX® and kdb+ are registered trademarks of KX Systems, Inc., a subsidiary of KX Software Limited.