# Why do we need a time-series database

[Why do we need a time-series database 1](#_Toc108107673)

[What is Time-Series Data? 1](#_Toc108107674)

[Features and functions of time series databases 2](#_Toc108107675)

[why can't I use a "normal" database？ 3](#_Toc108107676)

[TSDB's advantages in scale and availability. 3](#_Toc108107677)

[Time-Series Database Use Cases 4](#_Toc108107678)

### What is Time-Series Data?

Before explaining why we need a time series database, allow me to introduce what a time series database is.

In the past, in the traditional field, relational databases have been widely used due to their simple and easy-to-understand structure, convenient use and operation, flexible query and easy maintenance. But with the rapid development of the Internet of Things and related technologies. Due to the 24-hour uninterrupted production operation in industrial production, such as electricity, energy consumption and gas, a large amount of time series data is generated.

Time series data is time series data, and its biggest feature is that the data are arranged in chronological order. Common time series data include CPU usage, stock prices, and readings of various industrial sensors. Time series data is usually not long in a single piece of data, but the amount of data is large. The number of samples per day exceeds one million, and the capacity required to store the annual sampled data Approaching the terabyte level.

Modeling of time series data includes three important parts: subject, time point, and measurements. Applying this model, you will find that you are in constant contact with this type of data in your daily work and life.

* If you are a stockholder, the stock price of a stock is a type of time series data. It records the stock price of the stock at each time point.
* If you are O&M personnel, the monitoring data is a type of time series data. For example, the monitoring data of the CPU records the actual usage of the CPU at each time point.
* If you are an autonomous vehicle developer, change data is a time series data that records changes in the environment in which the autonomous vehicle continues to collect

Time series data is used to describe the state change information of an object in the historical time dimension. The analysis of time series data is the process of trying to understand and master the rules of the changes. Time series data experiences explosive growth with the development of IoT, big data, and artificial intelligence (AI) technologies. To better support the storage and analysis of such data, a variety of database products have come into being and are available on the market. The invention of this kind of database products is to solve the shortcomings and drawbacks of the conventional relational databases in terms of time series data storage and analysis. These products are uniformly classified as time series databases (TSDBs). Time series database is a database optimized for access to time series data. Time series database is widely used in the Internet, Internet of Things and other fields because of its unique support for time series data, and its application scenarios and scope are still expanding. In view of the characteristics of time series data writing frequency is much higher than query frequency, data update is less, and queries are mostly based on time period, most time series databases provide functions such as monitoring, downsampling, and aggregation, which are convenient for analysis, calculation and visualization of time series data.

### Features and functions of time series databases

Important features of a time series database include the following:

* Data lifecycle management: The process of managing the flow of data through its lifecycle from collection and ingestion to aggregation, processing, and expiration.
* Summarization: The practice of presenting a meaningful summary of your data through flexible queries, transformations, visualizations, and dashboards.
* Large range scans of many records: Scans of millions of time series records is a frequent requirement for many time series use cases. These types of scans require specialized software like time series databases that utilize purpose-built compression, indexing, and spatial generalization algorithms that enable users to quickly write, query, and visualize millions of points.

These features are designed to facilitate large-scale processing of large volumes of time series data. Common tasks of a time series database include the following:

* Write high volumes of data. Whether you’re collecting and writing data at the nanosecond precision for high frequency trading or collecting data from hundreds of thousands of sensors, time series databases are optimized for high ingest rates that other databases simply can’t handle.
* Request a summary of data over a large time period. Collecting summaries of your data over large time periods helps you gain valuable insights into the behavior of the data overall. For example, you might want to look at the mean monthly temperature of various cities for many years before deciding which city you want to move to.
* Automatically downsample or expire old time series that are no longer useful or keep high-precision data around for a short period of time. For example, monitoring the pressure of a pipe in a chemical plant every minute could be critical for upholding safety standards during operation. However, that data doesn’t need to be retained at a high precision forever. A time series database should allow the user to downsample that minute precision data to a daily average.

### why can't I use a "normal" database？

Faced with these time series data, you may ask why can't I use a "normal" (ie non-time series) database to store it?

A relational database is a collection of data items organized in tables from which data can be accessed or reorganized in many different ways. Each row in the table contains a unique data instance for the corresponding data category. When creating a relational database, the constraints and relationships between data make it a "relational" table. Most relational databases use Structured Query Language (SQL) to query and modify data stored in the database. The relational database can reasonably reduce the redundancy of the database through the paradigm design during design. However, the paradigm processing also makes a slightly more complex query need to use a large number of join table queries (ie join operations), which in turn leads to poor query performance. decline. With the explosive growth of time series data, the amount of data has also reached tens of millions or even higher. In the face of these time-series big data, the performance problems exposed by traditional relational databases are becoming more and more serious, and it is difficult to meet the low-latency requirements of users when reading and writing data. In a production environment, it often takes several 10s to query a specific time series data from a relational database, and this time will further increase as the amount of data increases.

### TSDB's advantages in scale and availability.

**Scale**: Time series data accumulates very quickly, and normal databases are not designed to handle this scale (at least not in an automated way). Traditionally, relational databases have not performed well with very large datasets In contrast, time-series databases - whether they’re relational or NoSQL-based - introduce efficiencies that are only possible when you treat time as a first-class citizen. These efficiencies allow them to offer massive scale, from performance improvements, including higher ingest rates and faster queries at scale (although some support more queries than others) to better data compression.

**Usability**: TSDBs also typically include built-in functions and operations common to time-series data analysis, such as data retention policies, continuous queries, flexible time aggregations, etc. Even if you’re just starting to collect this type of data and scale is not a concern at the moment, these features can still provide a better user experience and make data analysis tasks easier. Having built-in functions and features to analyze trends readily available at the data-layer often leads you to discover opportunities you didn’t know existed, no matter how big or small your dataset.

### Time-Series Database Use Cases

#### Accessing IoT Data

Most IoT implementations, such as connected water, energy, and temperature meters, necessitate regular data collecting and reporting. Seasonal patterns, average consumption, and inefficiencies can all be identified via time-series analysis, which provides time-stamped data points. A connected pH meter attached to a TSDB, for example, can alert a technician in charge of maintaining a certain pH level that a specific vat of water is becoming too acidic. Massive volumes of data are collected by IoT endpoints which require highly scalable time-series databases.

#### Forecasting Financial Trends

Accurately predicting financial trends using just time-series data is extremely difficult. A TSDB, on the other hand, can give a lot of contextual data to aid analysts. Consider the stock market: a significant surge in airline stock could coincide with holiday travel. Alternatively, a change in corporate leadership may frighten investors, leading the stock to drop briefly. Cross-referencing data is simple with time-series databases which results in a richer, clearer picture.

#### Monitoring Web Services

Time-series databases can be used by businesses to assess the success of their applications and web properties. The open-source monitoring system Prometheus, for example, is a time-series database that allows engineers to track performance patterns across time. This helps them to quickly notice when problems arise, allowing them to schedule maintenance and respond to occurrences to maintain an optimal user experience. Some web and mobile apps use a TSDB to keep track of certain events such as a button click, playing a video, or sharing some content. They can use these events to map a user's path, highlight challenges or performance bottlenecks, and streamline more sophisticated activities.

#### Sales Forecasting

Retail shops are obligated to constantly estimate future sales in order to appropriately stock their shelves with merchandise. Thanks to time-series databases, retailers can use statistical models based on historical data and cross-reference the data with customer behavior trends to predict future patterns and make informed decisions about which products to keep in stock and when.

#### Anomaly Detection

Anomaly detection aids in the detection of out-of-the-ordinary aberrations in time-series data. When a system change occurs, time-series data captures a value. Organizations can use these values to track changes, uncover how changes occurred in the past, keep track of what's going on now, and use the accumulated data to forecast future occurrences.

A major aspect of detecting anomalies is virtualization. A time-series graphic, for example, provides the visual aid that many people want while looking for outliers. Another option is to use automated anomaly detection, which can speed up the process by providing real-time information. This makes it possible to swiftly connect outliers.