# Why do we need a time-series database

Before we understand why we need to use a time-series database, we need to figure out what exactly is a time-series database .Time-series data is a series of data generated over time, which is simply time-stamped data. A Time Series Database (TSDB) is a Database optimized for ingesting, processing, and storing timestamp data. Such data may include metrics from servers and applications, readings from sensors in the Internet of Things, user interactions on websites or applications, or transaction activity in financial markets. Now,we can try to understand why do we need a time-series database.

## 一、The characteristics of time-series data

**1. Data characteristics:**

Large amount of data, data growth over time, repeated values in the same dimension, smooth changes of indicators (a device of a vehicle uploaded smooth changes of track coordinates).

**2. Write features:**

High concurrent write, and no update (trajectory does not update).

**3. Query features:**

Statistical analysis is conducted on indicators in different dimensions, and there are obvious hot and cold data. Generally, only recent data will be queried (generally, we only care about recent track data).

**4.Once stored, it cannot be modified**

New data will only be added to the system and will not be changed to another value at some future time.

**5.The importance of recent data trumps older data**

**6.The smaller the time interval, the smaller the difference**

The temperature of a place, for example, varies very little if it is monitored in seconds

## 二、compare with a traditional database

**1.MySQL has the following problems in the scenario of massive sequential data：**

**①High storage cost:**

Poor compression of sequential data takes up a lot of machine resources;

**②High maintenance cost:**

single-machine system, need to manually divide the database and table in the upper layer, high maintenance cost;

**③Low write throughput:**

The single write throughput is low, which is difficult to meet the write pressure of tens of millions of sequential data;

**④Poor query performance:**

Applicable to transaction processing, poor performance of mass data aggregation analysis.

**2.In addition, using Hadoop ecology (Hadoop, Spark, etc.) to store sequential data has the following problems:**

**①High data delay:**

offline batch processing system, data from generation to analysis, time-consuming hours or even days;

**②Poor query performance:**

Cannot make good use of indexes, relies on MapReduce tasks, and the query time is usually in minutes.

**3.You can see that the sequential database needs to address the following issues:**

**①Sequential data writing:**

how to support tens of millions of data points written per second.

**②Sequential data reading:**

how to support grouping and aggregation of billions of data at the second level.

**③Cost-sensitive:**

The problem with massive data storage is cost. How to store these data at a lower cost will become the top priority to be solved in timing database.

**4.Advantages of sequential databases**

**①Storage cost:**

Using the characteristics of time increasing, dimension repeating and index smooth change, reasonable selection of coding compression algorithm to improve the data compression ratio;
You can pre-reduce the accuracy to aggregate historical data and save storage space.

**②High concurrent write:**

Write data in batches to reduce network overhead；
Data is first written to the memory and then periodically dumped as immutable files.

**③Low query delay, high query concurrency:**

Optimize common query mode and reduce query delay through index and other technologies;
Improve query concurrency through caching and routing.

**5.The difference between the two**

(1) Structured data is stored. We all know that the traditional big data solution to store

the data contains structured, semi-structured, and unstructured data, thus decided the we can't decide what field and define the field of data types, like hbase is unified storage by byte type, that is to say on the hbase is the data in the byte array, Converting from a normal type to a byte array is something we have to do ourselves, and we don't know how to convert to byte to make it more efficient for storage. However, the data generated by sequential data are all structured data. We can define the fields and types of data in advance, and let the database system select the optimal compression mode according to different field types, thus greatly improving the storage utilization rate.

(2) Analysis aggregates structured data. Since analysis aggregations are structured data, we don't need to use complex computing tools like MapReduce, and generally don't need data warehouses like Hive. Instead, we just need to consolidate the database storage level with computing tools like SUM and AVG, and we can even do some simple streaming calculations. It provides the basis for "super fusion" (super fusion means that multiple components similar to the previous big data processing scheme are fused into one component, mainly because structured data is too simple, collection and calculation are relatively simple, which is also the development trend of sequential database in the future to reduce the system complexity).

## 三、Problems that need to be solved due to large amount of data

**1.Receiving and storing data per second must be fast**

Temporal data has many data points and high update frequency. Currently, LSM technology is widely used for temporal data storage instead of B tree. LSM accumulates data in memory and then writes it to disk in batches, whereas B trees have the advantage of reading rather than storing.

**2.It can effectively compress data and save storage space**

Most temporal databases currently use Facebook's Gorilla algorithm, which simply stores data differences, as temporal data is characterized by high frequency and low variance. If the traditional relational database is directly used to store the sequential data, the storage cost will be extremely high. Compared with the traditional relational database, the sequential database only needs 1/20 or even less storage space.

**3.On the basis of huge amount of existing data, how to achieve fast query**

Many sequential databases are column databases, which have better performance for analyzing data.

**4.Temporal data often requires a retention strategy**

For example, data from a few years ago is processed and retained differently than data from a few months ago. Sequential databases have corresponding processing strategies.

## 四、Why do we need a time-series database

**1.Scale:**

Time series data is accumulated very quickly. (For example, a connected car can collect 25 gigabytes of data per hour.) Conventional databases are not designed to handle data of this size, and relational databases are very poor at handling large data sets; The NoSQ database L handles scale data well, but not as well as a database fine-tuned for time series data. In contrast, time series databases (which can be based on relational or NoSQL databases) treat time as a first-class citizen, processing such large data volumes with increased efficiency and performance gains, including: Higher Ingest Rates, faster large-scale queries (although some support more queries than others), and better data compression.

**2.Availability:**

TSDB also typically includes some common capabilities and operations for analyzing time series data: data retention policies, continuous queries, flexible time aggregation, and so on. Even if you don't care about scale right now (for example, if you're just starting to collect data), these features can still provide a better user experience and make your life easier.

## 五、Application scenario of sequential database

Application scenarios of sequential database Are widely used in Internet of Things, Internet APM and other scenarios. The following lists some application scenarios of sequential database, but not all of them:

① Public safety: Internet access records, call records, individual tracking, interval screening;

② Power industry: centralized monitoring of smart meters, power grids and power generation equipment;

③Internet: server/application monitoring, user access log, AD click log;

④ Internet of things: elevators, boilers, machinery, water meters and other networked equipment;

⑤ Traffic industry: real-time road conditions, intersection flow monitoring, bayonet data;

⑥Financial industry: transaction records, access records, ATM, POS machine monitoring;

## 六、Future and outlook

Timing database is in the stage of rapid development, timing data technology is gradually mature, but this is by no means the end, timing data technology is still facing a variety of new requirements and challenges. As vendors improve the performance of sequential databases, more solutions are being proposed to meet new requirements:

(1) Cloud services. In addition to the stand-alone version, many manufacturers also released the distributed version, cloud service version, especially cloud service, has become an inevitable development trend.

(2) Visual services. With the advent of the Internet of everything, the demand of users for a comprehensive grasp of information is increasing, and the visual display of temporal data has become a major trend, which puts forward higher requirements for the query ability of temporal database.

(3) Edge computing services. Sensors in the Internet era of all things, to bring the huge amount of data is centralized handling difficult to load and this makes the data calculation to the marginalized development, equipment the real-time processing of data by edge equipment analysis feedback after the centralized storage, can improve equipment real-time response ability, increase the value of the timeliness of data, therefore, The support of edge computing in sequential database will become an important function.

In the face of these challenges and opportunities, it is believed that time series database will have a deeper development in the future.