Why do we need a time-series database?

To understand why we need timing databases, we first need to get a detailed understanding of the timing database and its related applications.

The so-called database, strictly speaking, should be called a relational database.The characteristic of relational database is that it stores the data. the data will be stored in rows and columns, which is convenient for reading and query.A relational database has been used in many occasions and has proved universal due to clear and understandable design principles such as naming normalization, data consistency, completeness, redundancy avoidance, and a well-defined paradigm.

But the relational databases have had trouble in industrial production.When needing to record the time series data, the various operations of the relational database appear inefficient and cumbersome.We first need to understand, in industry, time series data refers to the electric power industry, chemical industry, meteorological industry, geographic information and other types of real-time monitoring, inspection and analysis equipment collection, data, they are characterized by fast frequency (each monitoring within a second can produce more data), seriously dependent on the collection time (each data requires the only time), more information (conventional real-time monitoring system has thousands of monitoring, monitoring data every second, produce dozens of GB data every day).Intuitively, recording such data requires a lot of writing operations, and it requires the database with the ability to allow for large-scale changes.According to the relevant expertise, we can simply analyze the disadvantages of relational databases: more literally, relational databases are static in the face of such large writing and change requirements.

According to the principles of the relational database design already understood, we can know that the properties and attribute capacity of the relational database have been decided at the beginning of the design. If the data changes in the subsequent record and the attribute capacity is just small, this will have serious consequences and the data cannot be recorded; if all the data attribute capacity is defined to be large, it will be wasteful and unprofessional.Moreover, relational databases are stored in the form of tables, and data is stored in rows and columns, which makes reading and query convenient to write difficult, let alone change the data.As it happens, time series data requires a lot of writing and changes.Therefore, in the face of the storage requirements of time series data, due to the limitations of its own characteristics, relational databases appear inadequate, and we are also in urgent need of a database that can effectively store time series data.

Next, we will lead to the focus of this paper, that is, the introduction of the timing database and its related characteristics.Simply put, the timing database is fully called the time series database, and the time series database is a database dedicated to storing and processing the time series data.The timing database has some basic concepts, which are different from the rows and columns in the relational database, and they need to be distinguished:

1. Metric: Metric are similar to tables in a relational database, representing a series of similar timing data.For example, when storing monitoring data for a temperature sensor, we need to create a metric to store the timing data;

2. Tags: Tags are used to describe the characteristics of the data sources, and usually do not change over time.Still using temperature sensors, the fixed information in the area of the sensor device is considered tags.The timing database is indexed for labels to support relevant retrieval operations about tags.Tags consists of tag keys and tag values, both String types;

3. Timestamp: The timestamp represents the time point of data generation, which can be specified when written or automatically generated by the system;

4. Field: The field describes the measurement index of the data source, usually constantly changing with time, such as the temperature sensor includes the field like temperature;

5. Data point: A certain field value generated by a data source at a certain time is called a data point, which is when query and written by the number of data points in the database;

6. Time series: an indicator of the data source changes with time, forming a time series, and determining a time series of measurement + label + measurement value combination; the calculation of timing data includes downsampling, aggregation (sum, count, max, min, etc.), and interpolation are based on the time series dimension.

Time sequence database is also significantly different from the traditional relational database.First of all, we can consider the data writing aspect. In this respect, we can summarize the following writing characteristics according to the characteristics of the timing data:

1. Large data volume: Take the monitoring data as an example. If the time interval of the monitoring data collected is 1s, one monitoring item will produce 86400 data points per day, and if there are 10000 monitoring items, there will be 864000000 data points in a day.In the Internet of Things scenario, the number will be even bigger.The size of the whole data is TB or even PB level;

2. Cold and hot clarity: The timing data has a very typical hot and hot characteristics. The more historical the data is, the lower the probability of being queried and analyzed;

3. Timeliness: Time sequence data has timeliness. The data usually has a saving period. The data beyond this saving period can be considered invalid and can be recovered.On the one hand, the more historical the data is, the lower the available value, and the low-value data can be cleaned up to save on storage costs；

4. Multi-precision data storage: In the characteristics of the query, mentioned timing data for storage cost and query efficiency consideration, will need multi-precision query, but also need multi-precision data storage.

Because the timing database wants to achieve the above characteristics, this requires that it must have some characteristics, or the ability to meet some characteristics.Time series database is not a relational database with a time stamp, it needs to be used in the scenario of massive data, so it must have different characteristics from the traditional relational database to solve a large number of time series data problems.

First, the timing database must be able to support the writing of the timing data, which is usually tens of millions of data points per second; and the timing database must support the reading of the timing data.Similarly, such reads are a packet aggregation operation of hundreds of millions of data in seconds; otherwise, timing databases must address cost-sensitive issues.This is the cost of having huge amounts of data storage.How to store massive amounts of data at low cost will become the top priority among the problems of timing databases.

Different timing databases have different solutions to these thorny problems.There is a database called RRD (Round Robin Database), which is a circular database composed of a fixed size data file to store the data.This database does not be as traditional as for the size of files as data grows.RRD is created after its file size is fixed, you can imagine it as a circle, the many diameter of the circle divides the circle into a sector, each sector is the slot can save data, each slot is marked with a time stamp.There is in the center a pointer that, over time, after retrieving the data, the pointer is responsible for filling the data in the corresponding slot position.When the pointer turns to 360 degrees, the initial data is overwritten, so that the RRD cycle fills with the data.

The other solution is given by InfluxDB.InfluxDB built a wheel called Time Structured Merge Tree, simply TSM.The name of the TSM is very misleading, it is not a tree structure, but not a log, but a Merge.When TSM writes, it writes data to memory and then to the hard drive.During the read operation, the TSM reads both memory and the hard disk and merges the results.During the deletion, the TSM writes a deletion tag, so that the marked data is not returned when read; the background merges small blocks into large blocks, and then the marked and deleted data is really deleted.Compared with ordinary data, regular time series data can greatly improve the compression ratio during merging.So it seems that the TSM created by InfluxDB itself really has a very good effect in the application of the timing database.

At present, it seems that timing database is still an emerging industry, and many companies are also trying to develop their own timing databases.In the current Internet industry, the timing database has a high heat, there are a lot of timing database available for use.Specialized timing databases include TimescaleDB, KairosDB, and InfluxDB, as mentioned above.Certain generic databases also support timing databases, such as CrateDB and Kudu.Due to space constraints, I won't introduce it all here.I will pick up TimescaleDB, CrateDB, and InfluxDB for a brief introduction.

TimescaleDB is a timing database built based on the PostgreSQL database and supports SQL.TimescaleDB has a plug-in form and upgrades with PostgreSQL upgrades, without trouble with another branch.Due to the original architectural features, the TimescaleDB data is automatically sharded over time and space.Timescale DB can also be automatically sized.In sum, TimescaleDB is a timing database that can be automatically optimized and provides rich SQL functions.Perhaps TimescaleDB can be called the insufficient place is only suitable for the small amount of data.

CrateDB is based on Elastic Search, but it also supports SQL.CrateDB combines the knowledge of SQL with the NoSQL extensibility and data flexibility, allowing users to use SQL for queries, JOIN and aggregation, and simple scaling at real-time speed.CrateDB has openness, flexibility, simple scalability, and high availability, and backup.CrateDB is for a quick search and querying data while written.CrateDB supports large amounts of data, providing highly available databases.But CrateDB is not a good choice when strong transaction consistency and highly normalized patterns that include many tables and connections are needed.In addition, CrateDB has poor SQL support and read performance.

InfluxDB is an open-source sequential database developed by InfluxData that is introduced because it ranks first in the sequential database leaderboard, or DB-Engines Ranking.InfluxDB is easy to deploy, easy to use, and can be deployed independently without any external dependency.InfluxDB provides a query language similar to SQL, is interface-friendly, and has rich aggregation operations and sampling capabilities.InfluxDB is a set of software designed for temporal storage, high-performance reading and writing, real-time operation, and high availability.InfluxDB performance is very powerful, compared with OpenTSDB, MongoDB, Graphite, Cassandra, InfluxDB performance advantages and cost advantages are obvious.In addition to the above features, InfluxDB also has unstructured data models, rich permission and aging management systems, as well as plug-free configuration and third-party-dependent native HTTP management interface.InfluxDB's class SQL query language is low cost and fast to learn.InfluxDB also has some drawbacks, such as poor performance of concurrent queries.But with such strong performance and cost advantages, InfluxDB's flaws do not affect its ranking and reviews.

After spending a lot of time on the timing database, I believe we have an answer to the opening question.Not that why we need a timing database, but that it is a hard production and research need.As one of the current Internet hot spots, time sequence database has rich exploability and application prospects.Our need for the timing database is more because of the technological progress requirements caused by the continuous exploration of the Internet technology.