



Open in app

Get started



Published in DataDrivenInvestor



Sam Fangman

Follow

Jan 28, 2019 · 5 min read · Listen



Save



DDI • gain access to expert views

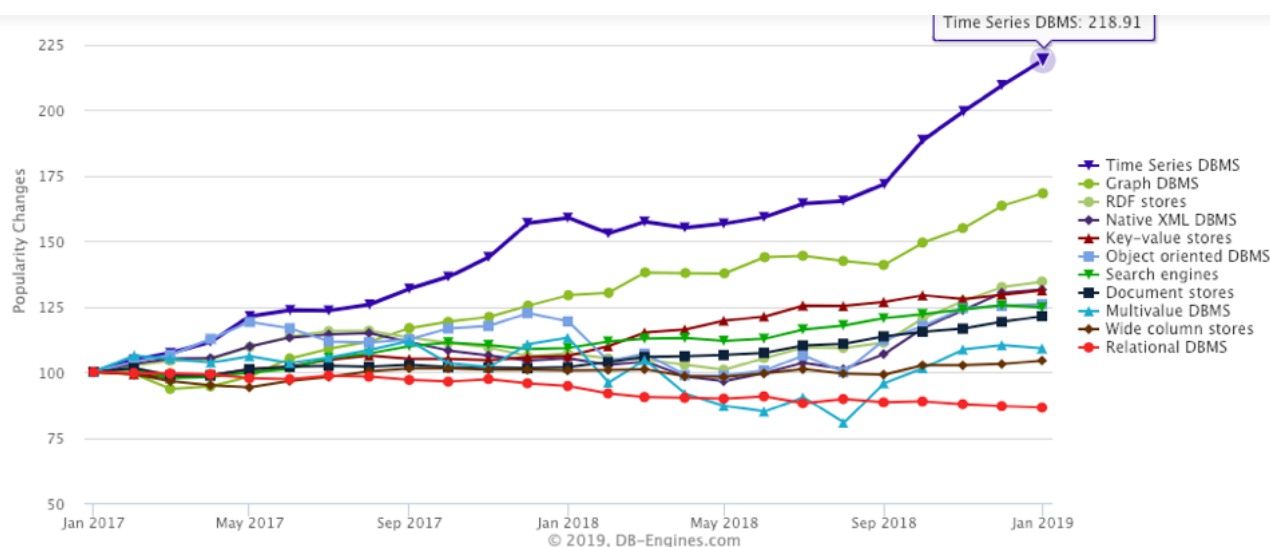


The Time Has Come for a New Type of Database

Exploring the emergence of the Time-Series Database

Long a mainstay of data management, numerous advances over the past decade have pushed database technology toward exciting new realms. Optimized solutions now utilize a variety of hardware platforms to store and analyze data in different ways, with in-memory and graph databases notable in their recent spikes in popularity. However, no database has had as much current momentum as the Time-Series Database. According to DB-Engines.com, the Time-Series Database has seen an increase in popularity of over 218% in the past two years.



[Open in app](#)[Get started](#)

Source: https://db-engines.com/en/ranking_categories

What is Time-Series Data?

Identifying as sequences of timestamped data points collected over a period of time, time-series data is sent to the database as a new write rather than an update, with their timestamps comprising the dataset primary axis. While these datasets are composed of snapshots at specific instances, the primary concern of time-series data is how the system changes over time. A good picture of time-series data is found in a basic financial record for an account. Each individual data point is a snapshot of the account balance at a given time, providing a transaction history for the account when collected.

What is a Time-Series Database?

A Time-Series Database (TSDB) is simply a database optimized for handling time-series data. TSDBs are columnar in structure and indexed by data timestamps, with internal data points consisting of key/value pairs with an accompanying timestamp. This allows for multiple metrics collected at a specific time to be stored in a single database entry.

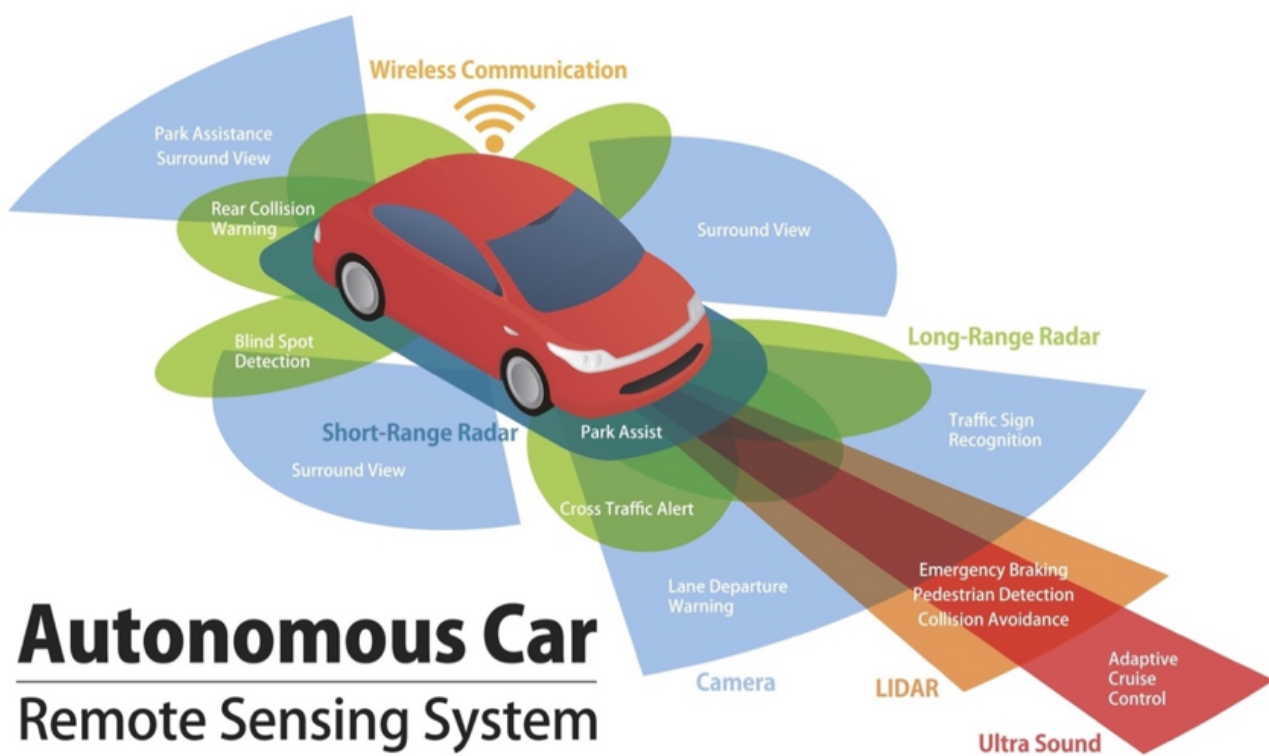
While time-series data can be stored in relational and NoSQL databases, both fall significantly short with regards to scale and usability. Relational databases in particular face challenges scaling up to handle large volumes of data, as vertically



[Open in app](#)[Get started](#)

the availability of any relational database.

TSDBs on the other hand are highly scalable, as they do not guarantee consistency (ACID properties). They can elastically scale up, with built-in capabilities for managing large data volumes. As time-series data is collected, it is common practice to keep high-precision data for only a set window, depending on the application. TSDBs allow for easy downsampling and data aggregation, so historical data outside that window can be reduced to long term trend data.



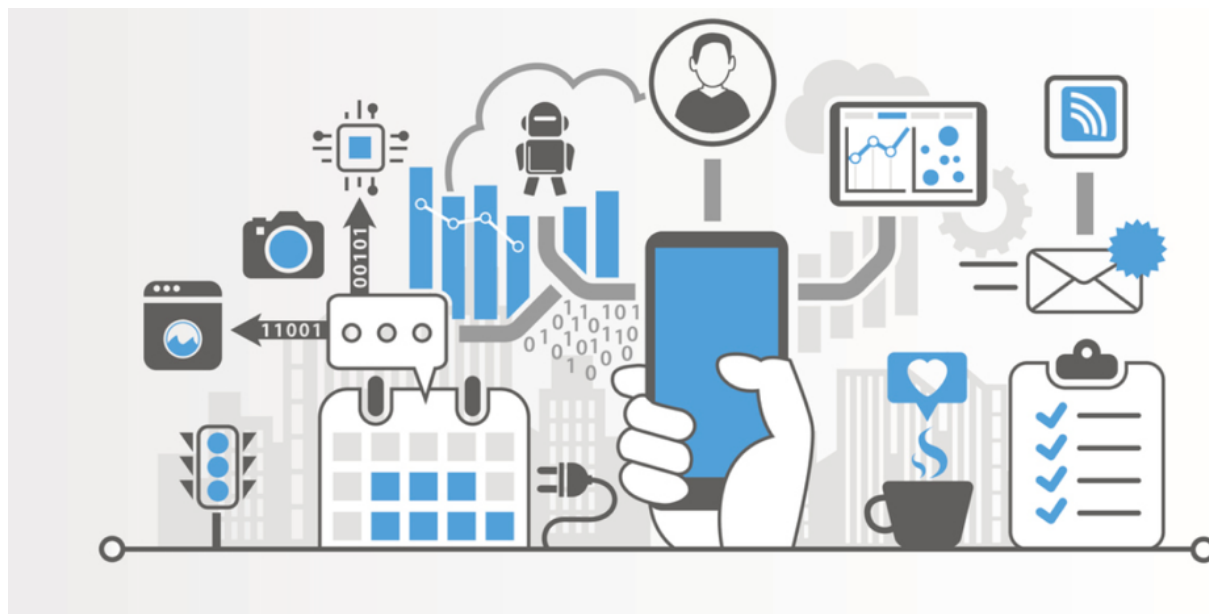
Source: <https://innovation-destination.com/2018/02/16/7-factors-critical-success-self-driving-cars/>

Though NoSQL databases are easily scalable, their usability for time-series data pales in comparison to purpose-built TSDBs. TSDBs contain numerous built-in common functions and features to process and analyze time-series data, such as continuous queries for real time analysis. Getting the same functionality in a NoSQL database requires a significant investment in development, testing, and maintenance, and even if a NoSQL database is customized to handle time-series data functions, the TSDB still wins out. The specialized functionality of TSDBs allow for significantly higher write throughput for data entry and faster query times for



[Open in app](#)[Get started](#)

Time and context. Time-series data is everywhere around us, waiting to drive business decisions. Financial trading algorithms constantly look to make decisions based on changing conditions in the market. DevOps environments are concerned with changes in system or application behavior. Transportation and logistics teams continuously seek how to optimize their supply chain. All this time-series data is readily available, and the market is beginning to take note of that.



Source: <https://www.flexwareinnovation.com/10-principles-of-the-internet-of-things-iot/>

Perhaps the most significant source of momentum for the TSDB has been the rise of IoT and connected devices. Autonomous cars constantly track changes in the surrounding environment, smart home sensors record metrics like changes in your home's temperature or the presence of intruders, and manufacturing machine telemetry utilizes streams of data to predict machine failure before the failure even occurs. With every sensor and device feeding back time-series data, the demand for an applicably optimized database has proportionally grown. The TSDB enables enterprises across sectors to fully utilize the abundance of time-series data to fuel business success.

Who is capitalizing on the momentum?

From its onset, open-source technology has been at the center of the Time-Series



[Open in app](#)[Get started](#)

maintaining, and customizing the technology. As a result, TSDBs can be appealing for businesses with well-established IT departments that have available resources to dedicate towards the solution. For companies that do not have the headcount or talent available, a number of providers offer paid solutions on top of open-source frameworks ([TimeScale](#), [RiakTS](#), [Kdb+](#), and [Canary](#)) allowing even smaller organizations the opportunity to reap the benefits of the TSDB. Further, a number of cloud providers, including [AWS](#) and [Azure](#), offer paid TSDB solutions, which can be appealing for companies already invested in public cloud infrastructure.

As the amount of time-series data continues to grow, the TSDB will further emerge as a crucial piece in enterprise data management stacks. While TSDBs will never replace the general abilities of a relational or NoSQL database, they will provide the functionality needed to keep pace in a growing data landscape.



Curious with how you can start using your time-series data? The [Trace3 Data Intelligence](#) team is here to discuss solutions that will help you start using your data more efficiently.





Open in app

Get started

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

