Advantages of NoSQL Databases

NoSQL databases were created in response to the limitations of traditional relational database technology. When compared to relational databases, NoSQL databases are often more scalable and provide superior performance. In addition, the flexibility and ease of use of their data models can speed development in comparison to the relational model, especially in the cloud computing environment.

Each specific type of NoSQL database has different strengths, but all share fundamental characteristics that allow them to:

- Handle large volumes of data at high speed with a scale-out architecture
- Store unstructured, semi-structured, or structured data
- Enable easy updates to schemas and fields
- Be developer-friendly
- Take full advantage of the cloud to deliver zero downtime

These capabilities provide users with many advantages compared to relational databases.

Handle large volumes of data at high speed with a scale-out architecture

SQL databases are most often implemented in a scale-up architecture, which is based on using ever-larger computers with more CPUs and more memory to improve performance.

NoSQL databases were created in internet and cloud computing eras that made it possible to more easily implement a scale-out architecture. In a scale-out architecture, scalability is achieved by spreading the storage of data and the work to process the data over a large cluster of computers. To increase capacity, more computers are added to the cluster.

This scale-out architecture is particularly painless to implement in cloud computing environments where new computers and storage can be easily added to a cluster.

The scale-out architecture of NoSQL systems provides a clear path to scalability when data volume or traffic grows. Achieving the same type of scalability with SQL databases can be expensive, require lots of engineering, or may not be feasible.

A financial services company like IHS Markit requires high performance both for ingesting data and for delivering it. Moving from a relational database to MongoDB, IHS Markit reports that it is able to deliver timely financial information to its customers 250x faster.

Store unstructured, semi-structured, or structured data

Relational databases store data in structured tables that have a predefined schema. To use relational databases, a data model must be designed and then the data is transformed and loaded into the database.

When data is used in applications, the data then must be retrieved using SQL, and adapted to the form used in the application. Then, when the data is written back, it must be transformed again back into the relational tables.

NoSQL databases have proven popular because they allow the data to be stored in ways that are easier to understand or closer to the way the data is used by applications. Fewer transformations are required when the data is stored or retrieved for use. Many different types of data, whether structured, unstructured, or semi-structured, can be stored and retrieved more easily.

In addition, the schemas of many NoSQL databases are flexible and under the control of the developers, making it easier to adapt the database to new forms of data. This removes bottlenecks in the development process associated with asking a database administrator to redesign a SQL database.

NoSQL databases support widely used data formats:

- Big data of all kinds – text data as well as time-series data

- JSON files, which are nested human-readable files consisting of names and value pairs. This format can capture highly complex parent-child hierarchical structures, which can be efficiently stored in document databases
- Simple binary values, lists, maps, and strings can be handled at high speed in key-value stores
- Sparse data can be efficiently stored in columnar databases, where null values take up no room at all. They are also effective for information that does not change frequently (nonvolatile data)
- Networks of interrelated information can be stored in graph databases.

Enable easy updates to schema and fields

NoSQL databases have become popular because they store data in simple straightforward forms that can be easier to understand than the type of data models used in SQL databases.

In addition, NoSQL databases often allow developers to directly change the structure of the data.

- Document databases don't have a set data structure to start with, so a new document type can be stored just as easily as what is currently being stored.
- With key-value and column-oriented stores, new values and new columns can be added without disrupting the current structure.
- In response to new kinds of data, graph database developers add nodes with new properties and arcs with new meanings.

Developer-friendly

Adoption of NoSQL databases has primarily been driven by uptake from developers who find it easier to create various types of applications compared to using relational databases.

Document databases such as MongoDB use JSON as a way to turn data into something much more like code. This allows the structure of the data to be under the control of the developer.

In addition, NoSQL databases store data in forms that are close to the kind of data objects used in applications, so fewer transformations are required when moving data in and out of the databases.

NoSQL databases can store data in native formats, which means developers don't have to adapt the data to the store. Storing data "as is" means not having a front-end ETL system to shoe-horn semi-structured data into row and column formats, and fewer applications to develop or buy to get a new database launched.

Most NoSQL databases have a strong community of developers surrounding them. This means that there is an ecosystem of tools available and a community of other developers with which to connect.

Take full advantage of the cloud to deliver zero downtime

The scale-out architecture that most NoSQL databases use does more than provide a clear path to scaling to accommodate huge datasets and high volumes of traffic. Delivering a database using a cluster of computers also allows the database to expand and contract capacity automatically.

In addition, many NoSQL databases can be upgraded and allow the structure of the database to change with zero downtime.

Developers in general and JavaScript developers in particular gravitate toward MongoDB, affirming its ongoing record as the database most wanted by developers according to StackOverflow.

Learn more about the specific advantages of MongoDB or get started right away with MongoDB Atlas, the fully managed version of MongoDB that runs on all the major public clouds.

Related resources

- AWS re:Invent 2022 Presentation, From RDBMS to NoSQL
- What is NoSQL? NoSQL Databases Explained
- What are the main differences between NoSQL and SQL?
- When should you use a NoSQL database?
- What are the 4 different types of NoSQL databases?
- NoSQL data modeling and schema design
- Exploring NoSQL Database Examples