SQL & NoSQL databases: How they differ

There are two primary databases used for storing digital data: SQL (relational databases) and NoSQL (non-relational databases). Though both methods effectively store data, they differ in their structures, scalability, relationships, language, and support.

In this article, you'll learn about each type of database, how they are similar and different from one another, and how to decide which type of database is suitable for your particular data application.

What is SQL ?

Structured Query Language (SQL) is a programming language that allows both technical and non-technically-minded users to query, manipulate, and change data in a relational database.

Organized into columns and rows within a table, SQL databases use a relational model that work best with well-defined structured data, such as names and quantities, in which relations exist between different entities. Within a SQL database, tables are linked through "foreign keys" that form relations between different tables and fields, such as customers and orders or employees and departments.
SQL databases are scalable vertically, meaning that you can increase the maximum load by adding further storage components like RAM or SSD. While in some cases this may mean that SQL databases are limited by the resources available on the server, cloud-based storage and other technologies can provide more scalability with SQL.

What is NoSQL?

NoSQL databases are non-relational databases that store data in a manner other than the tabular relations used within SQL databases. While SQL databases are best used for structured data, NoSQL databases are suitable for structured, semi-structured, and unstructured data. As a result, NoSQL databases don't follow a rigid schema but instead have more flexible structures to accommodate their data-types. Furthermore, instead of using SQL to query the database, NoSQL databases use varying query languages (some don't even have a query language).

NoSQL databases are scalable horizontally, meaning that they use multiple nodes in a cluster to handle increased workloads. This allows data architects to simply scale them by supplementing clusters with additional servers.

Main differences between NoSQL and SQL

At a high level, NoSQL and SQL databases have many similarities.In addition to supporting data storage and queries, they both also allow one to retrieve, update, and delete stored data. However, under the surface lie some significant differences that affect NoSQL versus SQL performance, scalability, and flexibility.

Here are some of the main differences between SQL versus NoSQL databases:

Structure

SQL databases are table based, while NoSQL databases can be document-oriented, key-value pairs, or graph structures. In a NoSQL database, a document can contain key value pairs, which can then be ordered and nested.

Scalability

SQL databases scale vertically, usually on a single server, and require users to increase physical hardware to increase their storage capacities. In effect, while cloud-storage options are available, SQL databases can be prohibitively expensive for businesses when dealing with vast amounts of big data.

NoSQL databases offer horizontal scalability, meaning that more servers simply need to be added to increase their data load. This means that NoSQL databases are better for modern cloud-based infrastructures, which offer distributed resources.

Language

SQL databases use SQL (Structured Query Language). NoSQL databases use JSON (JavaScript Object Notation), XML, YAML, or binary schema, facilitating unstructured data. SQL has a fixed-defined schema, while NoSQL databases are more flexible.

Support

SQL is a popular standard language that is well supported by many different database systems, while NoSQL has varying levels of support in various database systems.

Regarding support, you’ll generally find that more help is available for SQL databases than NoSQL. This is because SQL is a more established technology and thus has many more users and developers who can help you with your problems. In contrast, NoSQL is still relatively new, with less help available on forums or through the community. Your support options may be limited if you run into difficulties using it.

When to use: SQL vs. NoSQL

Deciding when to use NoSQL versus SQL is essential because they differ in structure, capabilities, and ideal use cases.

A relational database like SQL is a great option if you’re looking to build an application structured around a relationship between data tables. SQL also works well when you want to ensure your data is consistent across tables. However, relational databases aren’t always the best choice regarding flexibility or scaling.

A non-relational NoSQL database doesn’t use structured tables but instead uses flexible schemas for unstructured data storage. This gives you more ability to scale your project as needed. However, it also means you have less control over consistency and data relationships.