

Comparison between NoSQL and SQL

What is a SQL database?

SQL, which stands for “Structured Query Language,” is the programming language that’s been widely used in managing data in relational database management systems (RDBMS) since the 1970s. In the early years, when storage was expensive, SQL databases focused on reducing data duplication.

Fast-forward to today, and SQL is still widely used for querying relational databases, where data is stored in rows and tables that are linked in various ways. One table record may link to one other or to many others, or many table records may be related to many records in another table. These relational databases, which offer fast data storage and recovery, can handle great amounts of data and complex SQL queries.

What is a NoSQL database?

NoSQL is a non-relational database, meaning it allows different structures than a SQL database (not rows and columns) and more flexibility to use a format that best fits the data. The term “NoSQL” was not coined until the early 2000s. It doesn’t mean the systems don’t use SQL, as NoSQL databases do sometimes support some SQL commands. More accurately, “NoSQL” is sometimes defined as “not only SQL.”

How SQL works

SQL databases are valuable in handling structured data, or data that has relationships between its variables and entities.

Scalability

In general, SQL databases can scale vertically, meaning you can increase the load on a server by migrating to a larger server that adds more CPU, RAM or [SSD](#) capability. While vertical scalability is used most frequently, SQL databases can also scale horizontally through sharding or partitioning logic, although that's not well-supported.

Structure

SQL database schema organizes data in relational, tabular ways, using tables with columns or attributes and rows of records. Because SQL works with such a strictly predefined schema, it requires organizing and structuring data before starting with the SQL database.

Properties

RDBMS, which use SQL, must exhibit four properties, known by the acronym ACID. These ensure that transactions are processed successfully and that the SQL database has a high level of reliability:

- Atomicity: All transactions must succeed or fail completely and cannot be left partially complete, even in the case of system failure.
- Consistency: The database must follow rules that validate and prevent corruption at every step.
- Isolation: Concurrent transactions cannot affect each other.
- Durability: Transactions are final, and even system failure cannot "roll back" a complete transaction.

Support

Because SQL databases have a long history now, they have huge communities, and many examples of their stable codebases online. There are many experts available to support SQL and programming relational data.

How NoSQL works

Unlike SQL, NoSQL systems allow you to work with different data structures within a database. Because they allow a dynamic schema for unstructured data, there's less need to pre-plan and pre-organize data, and it's easier to make modifications. NoSQL databases allow you to add new attributes and fields, as well as use varied syntax across databases.

Scalability

NoSQL databases scale better horizontally, which means one can add additional servers or nodes as needed to increase load.

Structure

NoSQL databases are not relational, so they don't solely store data in rows and tables. Instead, they generally fall into one of four types of structures:

- Column-oriented, where data is stored in cells grouped in a virtually unlimited number of columns rather than rows.
- Key-value stores, which use an associative array (also known as a dictionary or map) as their data model. This model represents data as a collection of key-value pairs.
- Document stores, which use documents to hold and encode data in standard formats, including XML, YAML, JSON (JavaScript Object Notation) and BSON. A benefit is that documents within a single database can have different data types.
- Graph databases, which represent data on a graph that shows how different sets of data relate to each other. Neo4j, RedisGraph (a graph module built into Redis) and OrientDB are examples of graph databases.

Properties

While SQL calls for ACID properties, NoSQL follows the CAP theory (although some NoSQL databases — such as IBM's DB2, MongoDB, AWS's DynamoDB and Apache's CouchDB — can also integrate and follow ACID rules).

The CAP theorem says that distributed data systems allow a trade-off that can guarantee only two of the following three properties (which form the acronym CAP) at any one time:

- Consistency: Every request receives either the most recent result or an error. MongoDB is an example of a strongly consistent system, whereas others such as Cassandra offer eventual consistency.
- Availability: Every request has a non-error result.
- Partition tolerance: Any delays or losses between nodes do not interrupt the system operation.

Support

While NoSQL has quickly been adopted, it has smaller user communities and, therefore, less support. NoSQL users do benefit from open-source systems, as opposed to the many SQL languages that are proprietary.

SQL	NoSQL
RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)	Non-relational or distributed database system.
These databases have fixed or static or predefined schema	They have dynamic schema
These databases are not suited for hierarchical data storage.	These databases are best suited for hierarchical data storage.
These databases are best suited for complex queries	These databases are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows CAP (consistency, availability, partition tolerance)
Examples: MySQL , PostgreSQL , Oracle, MS-SQL Server, etc	Examples: MongoDB , GraphQL , HBase , Neo4j , Cassandra , etc