redis

NoSQL Database

An introduction on NoSQL databases: Types of NoSQL databases, history, uses, and more.

Spin up a NoSQL Cluster for free

Try a NoSQL Database: Redis Enterprise

Overview

What is a NoSQL database?

A NoSQL database (also known as “no SQL” or “not only SQL”) is a distributed, non-relational database designed for large-scale data storage and massively parallel, high-performance data processing across many commodity systems. Unlike the traditional relational database approach, NoSQL gives you a way to work with data closer to the application. It is a modern data storage paradigm that provides data persistence for environments where high performance is the primary requirement. Within a NoSQL database, data is stored so that both writing and reading are fast, even under heavy load.

NoSQL databases are typically distributed systems where several machines work together in clusters. Each piece of data is replicated over those machines to deliver redundancy and high availability. The ability to store vast amounts of data in a distributed manner makes NoSQL databases faster to develop and deploy. NoSQL databases are built for specific data models and have flexible schemas, allowing programmers to create and manage modern applications.

A NoSQL datastore is non-relational, distributed, flexible, and scalable. Additionally, some common features of NoSQL database management systems include data clustering, flexible schema, replication support, and eventual consistency, which compares to traditional SQL database systems’ usual ACID transaction capability ensuring data integrity.

Unlike SQL databases, NoSQL database systems were initially developed as open source, with almost all of them now having an enterprise option available. Open source allows for a large community of supporters who contribute to the database, which increases innovation and the variety of solutions it addresses. Having an open source option means that developers can quickly and easily try new NoSQL databases at no cost.

What is Redis NoSQL?

Redis is an open source, in-memory key-value data structure store, which can be used as a database, cache, or message broker. Redis Enterprise is based on Redis open source and is a NoSQL database but also includes vital functionality to make it enterprise-hardened, with the inclusion of Redis support. Redis Enterprise is infinitely and linearly scalable, has 5-9s high availability, and can be easily geo-distributed. Redis open source is entirely compatible with Redis Enterprise.

A brief history of NoSQL databases

With the start of Web 2.0, the creation of various data types exploded, and the price of storage significantly dropped, causing the challenges associated with maintaining and accessing data to shift. Relational databases, consisting of rows and columns, were designed for structured data requiring complex analysis and operations of often connected data.

With the rapid growth of unstructured data, relational databases' structural and scaling limitations came into focus. Flexibility and the ability to handle large amounts of data quickly became requirements. In the late 2000s, NoSQL databases emerged, with the advantages of storing data more intuitively and in formats that suit today's applications. NoSQL databases solve problems that can't be solved with SQL or relational databases.

What is a document database?

A document database (also called a NoSQL document store) is a non-relational database that stores data as structured documents. It is a modern way to store data in JSON format rather than simple rows and columns. A document can be a PDF, a document, or an XML or JSON file.

A Document

```
{
“BookID”:“978-1449396091”,
“Title”:“Redis-The Definitive Guide”,
“BookID”:“Salvatore Sanfilippo”
“Year”:“2021”,
}
```

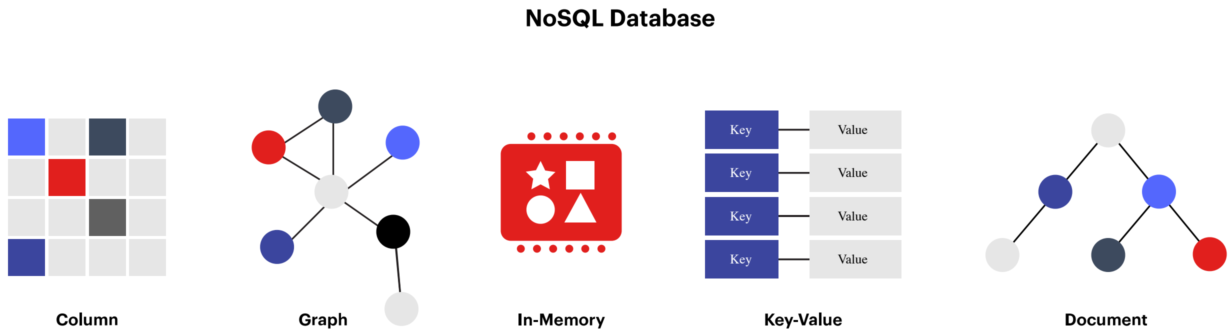
Key	Value
BookID	978-1449396091
Title	Redis - The Definitive Guide
Author	Salvatore Sanfilippo
Year	2021

JSON

JSON is a popular Redis module that provides in-memory manipulation of JSON documents at high velocity and volume. With JSON, you can natively store document data in a hierarchical, tree-like format to scale and query documents efficiently, significantly improving performance over storing and manipulating JSON with Lua and core Redis data structures. The native data type it applies is the Standard ECMA-404, using the Standard JSON Data Interchange Syntax. JSON outperforms any other technique for storing JSON objects in Redis, such as using Lua for manipulating JSON or MessagePack objects.

Type of NoSQL databases

There are five major NoSQL database types: key-value store, document store, in-memory, column-oriented database, and graph database. Implementing a NoSQL database is often a combination of these five database types.



Key-value stores are the least complex of the NoSQL databases. They are a collection of key-value pairs, and their simplicity makes them the most scalable of NoSQL databases types. Key-value can be a string, a number, and an entirely new set of key-value pairs encapsulated in an object. Use cases include shopping carts, user preferences, and leaderboards.

Document stores are one step up in complexity from key-value stores. NoSQL document-based databases store information in documents with specific keys, similar to a key-value store, but with different benefits and disadvantages. Document stores appear the most natural among the NoSQL database types because they store everyday documents. They allow for complex querying and calculations on this often already aggregated form of data. Document databases store data in CML, YAML, JSON, or binary documents such as BSON. Using specific keys makes document stores similar to key-value stores. Use cases include online retail, trading platforms, and mobile app development across industries.

In-memory databases are data stores that are purpose-built to rely on DRAM for data storage to enable sub-millisecond responses. This is in contrast to most NoSQL and SQL databases that store data on disk or SSD/flash memory. It means that each time you query an in-memory database or update data in a database, you only access the main memory. There's no disk involved in these operations. And this is good, because the main memory is much faster than any disk. In-memory databases can also retain persistent data by keeping each operation on a disk in a transaction log or a snapshot. Use cases include leaderboards, session stores, and real-time analytics.

Column-oriented databases store data in tables that contain large numbers of columns (in a flexible schema). In a column-oriented database, it's easy to add another column because none of the existing columns are affected by it. Storing each column separately allows for quicker scans when only a few columns are involved. Use cases include performing analytics and reporting, including summing values and counting entries.

A graph database is the most complex data store, geared toward efficiently storing relations between entities. Graph databases are the answer when the data is highly interconnected, such as social networks, scientific paper citations, or capital asset clusters. Use cases include fraud detection, social networks, and knowledge graphs.

SQL vs. NoSQL

What are the similarities and differences between SQL and NoSQL? In short, SQL is a relational database management system and NoSQL is non-relational database management system. SQL (Structured Query Language) is a type of database management system that is based on the relational model. It uses a structured schema, and data is organized into tables with rows and columns. SQL databases are typically used for transactional systems and applications that require complex queries and relationships between multiple data sets.

NoSQL (Not Only SQL) is a type of database management system that does not use a fixed schema. It is designed to handle unstructured and semi-structured data, and can scale horizontally to handle large amounts of data. NoSQL databases are typically used for big data and real-time web applications, and are often used in conjunction with big data tools.

DBMS	SQL databases	NoSQL databases
Type	Relational database	Non-relational database
Structure	SQL databases organize and store data by table and fixed columns and rows	NoSQL databases can be key-value, document column-oriented, and graph
Schema	Fixed schema	Dynamic schema
Scalability	Vertical scalability	Horizontal and vertical scalability
Query	Structured Query Language (SQL)	No declarative query language; it depends on the database type

What are the benefits of NoSQL?

Next section ▶ Key-Value Databases