

What is a NoSQL database?

The term NoSQL, short for “not only SQL,” refers to non-relational databases that use a non-tabular format to store data, rather than in rule-based, relational tables like [relational databases](https://cloud.google.com/learn/what-is-a-relational-database) (<https://cloud.google.com/learn/what-is-a-relational-database>). NoSQL databases use a flexible schema model that supports a wide variety of unstructured data, such as documents, key-value, wide columns, graphs, and more.

Organizations choose NoSQL databases for their flexibility, horizontal scalability, and ease of development.

Learn how Google Cloud’s NoSQL databases [Bigtable](https://cloud.google.com/bigtable) (<https://cloud.google.com/bigtable>), [Memorystore](https://cloud.google.com/memorystore) (<https://cloud.google.com/memorystore>), and [Firestore](https://cloud.google.com/firestore) (<https://cloud.google.com/firestore>) can help you enhance your applications and provide great customer experiences without worrying about scalability, reliability, or frequent data changes.

Get started for free (<https://console.cloud.google.com/freetrial>)

The five types of NoSQL databases

There are five main types of NoSQL databases:

Document database

Document databases, also called document-oriented databases or a document store, are used to store and query semi-structured data. Data is stored in a JSON-like document similar to the data objects that developers use in application code, making it easier to create and update applications without referencing a primary schema. Document databases are most commonly used for blogging platforms, ecommerce and real-time analytics applications, and CMS systems.

Key-value database

Key-value databases, also referred to as key-value stores, are the simplest type of NoSQL databases. Data is stored in a “key-value” structure, where a unique key is paired with a value, such as strings, numbers, booleans, complex objects, and more. Data can be written or queried using the key, which is used to store or retrieve its associated value. Key-value stores are most commonly used for user preferences, shopping carts, and user profiles in web applications.

Column-oriented database

Column-oriented databases or wide-column stores, store and read data in rows and are organized as a set of columns. While similar to the tabular format of relational databases, column names and formatting in wide-column stores can vary from row to row in a single table. They are optimal for analytics use cases, where you may need to query across specific columns in a database and aggregate the value of a given column quickly. Wide-column stores are most commonly used for catalogs, fraud detection, and recommendation engines.

Graph databases

Graph databases organize data as nodes, focusing on the relationships between data elements. These connections between nodes, referred to as edges, are stored as first-class elements, enabling richer representations of data relationships while offering more simplified storage and navigation. Graph

databases are most commonly used in systems that map relationships, including social media platforms, reservation systems, and logistics applications.

In-memory database

In-memory databases store data in memory, providing no latency for real-time applications. Redis is an example of an in-memory NoSQL database. In-memory databases are most commonly used for caching, messaging, streaming, and real-time analytics.

How does NoSQL work?

NoSQL features are unique to the database you choose. However, they typically share several similar high-level qualities:

- They follow flexible schemas that don't require you to determine or declare a fixed schema for your data, making them ideal for semi-structured and unstructured data
- They scale horizontally, using range or hash distributions, instead of having to depend on vertical scaling to add capacity
- They are optimized for specific data models and workload patterns, such as key-value, wide-column, or in-memory
- Many NoSQL databases exhibit consistency at some later point (for example, eventual consistency model) rather than following the stricter ACID (atomicity, consistency, isolation, durability) properties of relational and SQL databases
- The majority of NoSQL databases don't support cross-shard transactions or flexible isolation modes

These features make non-relational databases ideal for applications that require large scale, reliability, high availability, and frequent data changes.

Advantages of NoSQL

Flexible data model and schema

NoSQL databases store many different types of data and offer flexible schemas, ideal for semi-structured and unstructured data. You can easily adapt them to new types of data and evolve the schema to meet any changing data requirements.

Agile development

The flexibility of NoSQL complements agile app development. NoSQL databases can store many types of data in its native format and allows the data model to be defined and adapted as you go, so developers can get going fast, spend less time on data transformation, and iterate quickly.

Scalability

Unlike relational databases, NoSQL databases make it easy to increase capacity as data and traffic grows—in most cases, with zero downtime. Cloud-based databases are even easier to scale based on demand, offering autoscaling features and flexible pricing models.

Massive data storage

NoSQL is designed to handle large, complex datasets, allowing organizations to adopt and scale big data, real-time analytics, and IoT use cases.

High availability

NoSQL data architectures are distributed by design and have no single point of failure. They also provide easy replication, making them more resistant to unplanned outages and disruptions.

Faster queries

Unlike relational databases, which are normalized to reduce data duplication, NoSQL is optimized for fast querying. It typically does not require complex joins, meaning that database queries return results more quickly.

Disadvantages of NoSQL

While NoSQL databases have gained huge popularity in recent years, there are some drawbacks to using them over relational databases. They are still relatively new and may lack the maturity of relational data systems. Overall, NoSQL comes with less developer expertise, fewer available tools and products, and less support should undocumented issues arise.

In addition, NoSQL has no lingua franca like SQL—each database may have its own language for querying and managing data.

In many cases, NoSQL databases lack the data integrity safeguards and high level of data consistency that are standard in SQL databases. However, there are some, like Firestore (<https://cloud.google.com/firestore>) and MongoDB (<https://cloud.google.com/mongodb>), that do support ACID transactions (<https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained>)

.

Still, eventual consistency models are sufficient for most NoSQL use cases, where a small millisecond delay doesn't matter. For many applications, high availability and speed far outweighs the need for strong global consistency.

SQL versus NoSQL: Use cases and examples

So why do organizations use non-relational NoSQL databases for some applications or stick with relational SQL databases for others? Picking the right database

(<https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained>)

often comes down to your use case. Here's when to use SQL versus NoSQL.

SQL databases excel in managing structured, relational data and complex queries. SQL databases are ACID compliant, making them relied on for transactional information.

SQL use cases:

- Financial transactions
- Healthcare data analysis
- Customer and transaction information

NoSQL databases excel in use cases that require large numbers of globally distributed users to deliver highly personalized experiences. These interactive applications are best served with the agility of NoSQL databases.

NoSQL databases also prioritize high availability, deliver consistent performance, scale out quickly without downtime, and can handle large volumes of different types of data.

These features of NoSQL databases make non-relational databases ideal for applications that require large scale, reliability and high availability.

The most common NoSQL uses include:

- Mobile, web, and IoT applications
- Real-time web applications
- Personalization, recommendations, and real-time customer experiences
- Inventory and catalog management
- Fraud detection and identity authentication
- Financial services and payments
- Ad tech

Solve your business challenges with Google Cloud

New customers get \$300 in free credits to spend on Google Cloud.

Get started (<https://console.cloud.google.com/freetrial>)

Talk to a Google Cloud sales specialist to discuss your unique challenge in more detail.

Contact us (<https://cloud.google.com/contact>)

Related products and services

Google Cloud databases are built to provide industry-leading reliability, global scale, and open standards, including non-relational databases to support various use cases for document, key-value, and wide-column databases.



Firestore

Highly scalable document database service for mobile, web, and server development.



Bigtable

Fully managed, high-performance NoSQL database service for large analytical and operational workloads.



Memorystore

Fully managed Redis and Memcached for sub-millisecond data access.



SOLUTION

MongoDB Atlas

Global cloud database service for modern applications.



SOLUTION

Memorystore for Redis

Leverage the highly scalable, available, secure Redis service without the burden of managing complex Redis deployments.



SOLUTION

Memorystore for Memcached

Leverage the highly scalable, available, secure Memcached service without the burden of managing complex Memcached deployments.



Take the next step

Start building on Google Cloud with \$300 in free credits and 20+ always free products.

Get started for free (<https://console.cloud.google.com/freetrial/>)

Need help getting started?

Contact sales (<https://cloud.google.com/contact/>)

Work with a trusted partner

Find a partner (<https://cloud.google.com/find-a-partner/>)

Continue browsing

See all products (<https://cloud.google.com/products/>)