When it comes to choosing a database the biggest decision is picking a **relational (SQL)** or **non-relational (NoSQL)** data structure. While both databases are viable options still there are certain key differences between the two that users must keep in mind when making a decision.

**Main differences between NoSQL and SQL**

**Type**

[SQL](https://www.geeksforgeeks.org/sql-tutorial/) databases are primarily called Relational Databases [(RDBMS)](https://www.geeksforgeeks.org/rdbms-full-form/); whereas [NoSQL databases](https://www.geeksforgeeks.org/introduction-to-nosql/) are primarily called non-relational or distributed databases.

**Language**

SQL databases define and manipulate data-based [structured query language (SQL)](https://www.geeksforgeeks.org/structured-query-language/). Seeing from a side this language is extremely powerful. SQL is one of the most versatile and widely-used options available which makes it a safe choice, especially for great complex queries. But from another side, it can be restrictive. SQL requires you to use predefined [schemas](https://www.geeksforgeeks.org/create-schema-in-sql-server/) to determine the structure of your data before you work with it. Also, all of your data must follow the same structure. This can require significant up-front preparation which means that a change in the structure would be both difficult and disruptive to your whole system. 

A NoSQL database has a dynamic schema for unstructured data. Data is stored in many ways which means it can be document-oriented, column-oriented, graph-based, or organized as a key-value store. This flexibility means that documents can be created without having a defined structure first. Also, each document can have its own unique structure. The syntax varies from database to database, and you can add fields as you go.

**Scalability**

In almost all situations SQL databases are vertically scalable. This means that you can increase the load on a single server by increasing things like [RAM](https://www.geeksforgeeks.org/random-access-memory-ram/), [CPU,](https://www.geeksforgeeks.org/central-processing-unit-cpu/) or [SSD](https://www.geeksforgeeks.org/introduction-to-solid-state-drive-ssd/). But on the other hand, NoSQL databases are horizontally scalable. This means that you handle more traffic by sharding, or adding more servers in your NoSQL database. It is similar to adding more floors to the same building versus adding more buildings to the neighborhood. Thus NoSQL can ultimately become larger and more powerful, making these databases the preferred choice for large or ever-changing data sets.

**Structure**

 SQL databases are table-based on the other hand NoSQL databases are either key-value pairs, document-based, graph databases, or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions such as an accounting system or for legacy systems that were built for a relational structure.

**Property followed**

SQL databases follow [ACID properties](https://www.geeksforgeeks.org/acid-properties-in-dbms/) (Atomicity, Consistency, Isolation, and Durability) whereas the NoSQL database follows the Brewers [CAP theorem](https://www.geeksforgeeks.org/the-cap-theorem-in-dbms/) (Consistency, Availability, and Partition tolerance).

**Support**

Great support is available for all SQL databases from their vendors. Also, a lot of independent consultants are there who can help you with SQL databases for very large-scale deployments but for some NoSQL databases you still have to rely on community support and only limited outside experts are available for setting up and deploying your large-scale NoSQL deploy.

**When To Use: SQL vs NoSQL**

SQL is a good choice when working with related data. Relational databases are efficient, flexible, and easily accessed by any application. A benefit of a relational database is that when one user updates a specific record, every instance of the database automatically refreshes, and that information is provided in real-time.

SQL and a relational database make it easy to handle a great deal of information, scale as necessary and allow flexible access to data only needing to update data once instead of changing multiple files, for instance. It’s also best for assessing data integrity. Since each piece of information is stored in a single place, there’s no problem with former versions confusing the picture.

While NoSQL is good when the availability of big data is more crucial, SQL is valued for ensuring data validity. It’s also a wise decision when a business needs to expand in response to shifting customer demands. NoSQL offers high performance, flexibility, and ease of use.

NoSQL is also a wise choice when dealing with large or constantly changing data sets, flexible data models, or requirements that don’t fit into a relational model. Document databases, like CouchDB, MongoDB, and Amazon DocumentDB, are useful for handling large amounts of unstructured data.

**Key Highlights on SQL vs NoSQL**

| **SQL** | **NoSQL** |
| --- | --- |
| RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) | Non-relational or distributed database system. |
| These databases have fixed or static or predefined schema | They have a 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](https://www.geeksforgeeks.org/mysql-common-mysql-queries/), [PostgreSQL](https://www.geeksforgeeks.org/what-is-postgresql-introduction/), Oracle, MS-SQL Server, etc | **Examples:**[MongoDB](https://www.geeksforgeeks.org/mongodb-tutorial/), [GraphQL](https://www.geeksforgeeks.org/graphql-query/), [HBase](https://www.geeksforgeeks.org/architecture-of-hbase/), [Neo4j](https://www.geeksforgeeks.org/neo4j-introduction/), |

# **SQL vs NoSQL**

There are a lot of databases used today in the industry. Some are SQL databases, some are NoSQL databases. The conventional database is SQL database system that uses tabular relational model to represent data and their relationship. The NoSQL database is the newer one database that provides a mechanism for storage and retrieval of data other than tabular relations model used in relational databases.

Following is a list of differences between SQL and NoSQL database:

|  |  |  |
| --- | --- | --- |
| **Index** | **SQL** | **NoSQL** |
| 1) | Databases are categorized as Relational Database Management System (RDBMS). | NoSQL databases are categorized as Non-relational or distributed database system. |
| 2) | SQL databases have fixed or static or predefined schema. | NoSQL databases have dynamic schema. |
| 3) | SQL databases display data in form of tables so it is known as table-based database. | NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores. |
| 4) | SQL databases are vertically scalable. | NoSQL databases are horizontally scalable. |
| 5) | SQL databases use a powerful language "Structured Query Language" to define and manipulate the data. | In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database. |
| 6) | SQL databases are best suited for complex queries. | NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries. |
| 7) | SQL databases are not best suited for hierarchical data storage. | NoSQL databases are best suited for hierarchical data storage. |
| 8) | MySQL, Oracle, Sqlite, PostgreSQL and MS-SQL etc. are the example of SQL database. | MongoDB, BigTable, Redis, RavenDB, Cassandra, Hbase, Neo4j, CouchDB etc. are the example of nosql database |