The five critical differences between SQL and NoSQL are:

- SQL databases are relational, and NoSQL databases are non-relational.
- SQL databases use structured query language (SQL) and have a predefined schema. NoSQL databases have dynamic schemas for unstructured data.
- SQL databases are vertically scalable, while NoSQL databases are horizontally scalable.
- SQL databases are table-based, while NoSQL databases are document, key-value, graph, or wide-column stores.
- SQL databases are better for multi-row transactions, while NoSQL is better for unstructured data like documents or JSON.

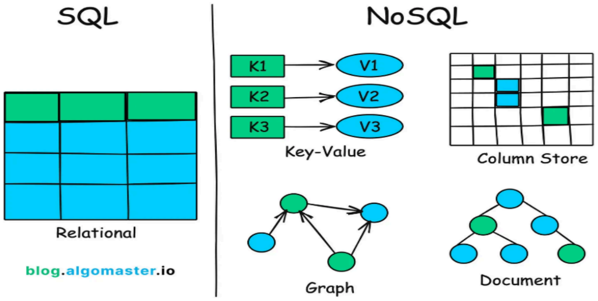
Which is better: SQL or NoSQL?

The decision of which type of database to use - SQL or NoSQL - will depend on the particular needs and requirements of the project. For example, if you need a fast, scalable, and reliable database for web applications then a NoSQL system may be preferable. On the other hand, if your application requires complex data queries and transactional support then an SQL system may be the better choice. Ultimately, there is no one-size-fits-all solution - it all comes down to what you need from your database and which type of system can provide that in the most efficient manner. It's best to research both options thoroughly before making a decision.

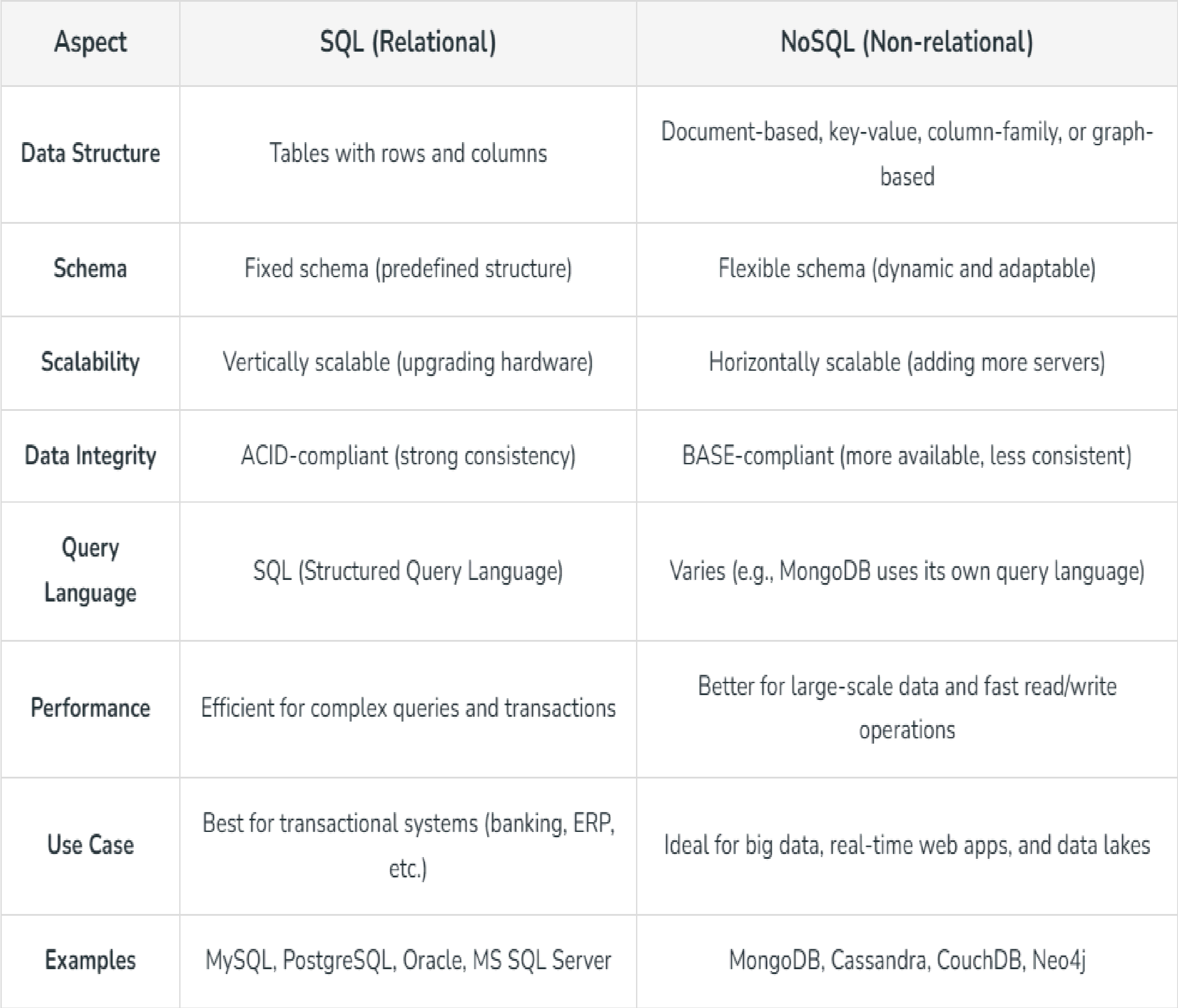
Below, learn in-depth about the most important distinctions between SQL vs NoSQL databases and the best systems available on the market.

Why NoSQL is Used Over SQL

NoSQL is preferred over SQL in many cases because it offers more flexibility and scalability. The primary benefit of using a NoSQL system is that it provides developers with the ability to store and access data quickly and easily, without the overhead of a traditional relational database. As a result, development teams can focus on delivering features and core business logic faster, without worrying about the underlying data storage implementation.



Differences Between SQL and NoSQL



Aspect	SQL (Relational)	NoSQL (Non-relational)
Data Structure	Tables with rows and columns	Document-based, key-value, column-family, or graph-based
Schema	Fixed schema (predefined structure)	Flexible schema (dynamic and adaptable)
Scalability	Vertically scalable (upgrading hardware)	Horizontally scalable (adding more servers)
Data Integrity	ACID-compliant (strong consistency)	BASE-compliant (more available, less consistent)
Query Language	SQL (Structured Query Language)	Varies (e.g., MongoDB uses its own query language)
Performance	Efficient for complex queries and transactions	Better for large-scale data and fast read/write operations
Use Case	Best for transactional systems (banking, ERP, etc.)	Ideal for big data, real-time web apps, and data lakes
Examples	MySQL, PostgreSQL, Oracle, MS SQL Server	MongoDB, Cassandra, CouchDB, Neo4j

What is MongoDB?

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and allows us to work with that data efficiently. It is categorized under the NoSQL database because the storage and retrieval of data in the MongoDB are not in the form of tables. The MongoDB database is developed and managed by MongoDB. Inc. under SSPL(Server Side Public License) and initially released in February 2009.

Features of MongoDB

MongoDB offers a wide range of features that make it a preferred choice for modern applications.

1. Schema-less Database

Unlike traditional relational databases, MongoDB collections:
Allow different structures within the same collection.
Do not require fixed column definitions.
Enable easy updates and modifications.

2. Document Oriented

In MongoDB, all the data is stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields(key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

3. Indexing

In the MongoDB database, every field in the documents is indexed with primary and secondary indices; this makes it easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then the database searches each document with the specified query which takes lots of time and is not so efficient.

4. Scalability

MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.

5. Replication

MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

6. Aggregation

It allows users to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods

7. High Performance

The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.