

What is SQL?

SQL is a domain-specific language used to query and manage data. It works by allowing users to query, insert, delete, and update records in relational databases. SQL also allows for complex logic to be applied through the use of transactions and embedded procedures such as stored functions or views.

What is NoSQL?

NoSQL stands for Not only SQL. It is a type of database that uses non-relational data structures, such as documents, graph databases, and key-value stores to store and retrieve data. NoSQL systems are designed to be more flexible than traditional relational databases and can scale up or down easily to accommodate changes in usage or load. This makes them ideal for use in applications

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.

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.

Comparison of SQL vs NoSQL:

With a basic understanding of what SQL vs NoSQL is, let's take a look at this quick comparison chart to see what sets the two apart:

SQL:

- Stands for Structured Query Language
- Relational database management system (RDBMS)
- Suitable for structured data with predefined schema

- Data is stored in tables with columns and rows
- Follows ACID properties (Atomicity, Consistency, Isolation, Durability) for transaction management
- Supports JOIN and complex queries
- Uses normalized data structure
- Requires vertical scaling to handle large volumes of data
- Examples: MySQL, PostgreSQL, Oracle, SQL Server, Microsoft SQL Server

NoSQL:

- Stands for Not Only SQL
- Non-relational database management system
- Suitable for unstructured and semi-structured data
- Data is stored in collections or documents
- Does not necessarily follow ACID properties
- Does not support JOIN and complex queries
- Uses denormalized data structure
- Horizontal scaling is possible to handle large volumes of data
- Examples: MongoDB, Cassandra, Couchbase, Amazon DynamoDB, Redis