

Differences between SQL and NoSQL databases:

SQL databases are relational databases (RDBMS) that define and store relationships in the form of tables. SQL is the query language used to manipulate data in the relational database. SQL databases have a predefined schema and are vertically scalable. Examples of SQL databases are MySQL, Oracle, MS-SQL Server etc.

NoSQL databases are non-relational (distributed) databases that can be document-based, key-value pairs graph databases. NoSQL databases do not use a declarative query language and use dynamic schema for unstructured data. The format can be XML, JSON, Binary JSON (MongoDB) etc. NoSQL databases are horizontally scalable and used primarily for big data and real-time web apps. Examples to NoSQL databases are MongoDB, Cassandra, HBase etc.

SQL databases prioritize data validity and use ACID (Atomicity, Consistency, Isolation and Durability) as their standard while NoSQL databases are used when speed is more important than validity of data. BASE (Basically Available, Soft state, Eventually consistent) is the model for many NoSQL databases.

Types of NoSQL databases

There are 4 basic types of NoSQL databases:

Key-Value Store – It has a Big Hash Table of keys & values {Example- Amazon S3 (Dynamo)}

Document-based Store- It stores documents made up of tagged elements. {Example- MongoDB, CouchDB}

Column-based Store- Each storage block contains data from only one column, {Example- HBase, Cassandra}

Graph-based-A network database that uses edges and nodes to represent and store data. {Example- Neo4J}

Data Modeling – SQL vs NoSQL

Data models define how data is connected to each other and how they are processed and stored inside the system.

SQL (relational) data modeling is driven by the structure of available data and the design theme is “What answers do I have?”

NoSQL data modeling is driven by the types of queries to be supported and the design theme is “What questions do I have?” The data models are application- specific so the information is stored in a way that optimizes query performance.

Data Modeling in NoSQL databases

Data modeling in relational databases has traditionally been front-loaded and can be inflexible and slow to adapt to the requirements of a rapidly changing business environment. In contrast, for NoSQL databases, data modeling needs to be done iteratively throughout the application life cycle and a flexible data model is needed to be able to support the large volume and variety of data being generated by modern applications. This adaptive data modeling, or flexible schema design, lets you incorporate new data into your applications without a predefined schema.

Basic principles of NoSQL data modeling:

- (1) Denormalization
Applicability: Key-Value Stores, Document Databases, Column-based Databases
- (2) Aggregates
Applicability: Key-Value Stores, Document Databases, Column-based Databases
- (3) Application Side Joins
Applicability: Key-Value Stores, Document Databases, Column-based Databases, Graph Databases
- (4) Atomic Aggregates
- (5) Enumerable keys
- (6) Dimensionality Reduction
- (7) Index Table
- (8) Composite Key Index

Data Modeling Tools for NoSQL databases:

1. Erwin DM NoSQL
2. KDM
3. hackolade
4. Creately
5. DynamoDB (Amazon), Azure DB (Microsoft) built-in modeling tools (TBD)