NoSQL:

NoSQL, also referred to as “not only SQL”, “non-SQL”, is an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases. While it can still store data found within relational database management systems (RDBMS), it just stores it differently compared to an RDBMS. The decision to use a relational database versus a non-relational database is largely contextual, and it varies depending on the use case.

Instead of the typical tabular structure of a relational database, NoSQL databases, house data within one data structure, such as JSON document. Since this non-relational database design does not require a schema, it offers rapid scalability to manage large and typically unstructured data sets.

Main Differences Between SQL and NoSQL Databases

Feature	SQL Databases	NoSQL Databases
Key Focus	Reducing data duplication	Scaling and rapid application change
Data Storage Model	Tables with fixed rows and colums	Document: JSON documents; Key-value: key-value pairs; Wide-column: tables with rows and dynamic columns; Graph: nodes and edges
Schemas	Rigid	Flexible
Data to Object Mapping	Requires ORM (object-relational mapping)	Typically doesn't require ORMs. E.g. MongoDB documents map directly to data structures in popular programming languages.
Scaling	Vertical (scale-up with a larger server)	Horizontal (scale-out across commodity servers)

TRUE ENGINEER

Types of NoSQL databases

1. Key-value store

2. Document Store

3. Graph Store

Key-value store

This is typically considered the simplest form of NoSQL databases. This schema-less data model is organized into a dictionary of key-value pairs, where each item has a key and a value.

Document store

As suggested by the name, document databases store data as documents. They can be helpful in managing semi-structured data, and data are typically stored in JSON, XML, or BSON formats.

Graph store

This type of database typically houses data from a knowledge graph. Data elements are stored as nodes, edges and properties.

Scale-Out (Horizontal Scalability):

•*Definition:* NoSQL databases are designed to scale horizontally, meaning that as the amount of data or traffic increases, you can simply add more servers or nodes to the database to handle the load.

•*Benefits:* This allows for distributing data across multiple servers, improving performance and accommodating growing datasets and user loads. It contrasts with traditional relational databases, which often scale vertically by adding more power (CPU, RAM) to a single server.

Flexibility (Schema-less or Schema-flexible):

•*Definition:* NoSQL databases often allow for flexible data models. Unlike traditional relational databases that require a predefined schema, NoSQL databases can handle diverse data types and structures.

•*Benefits:* Developers can adapt the database schema on the fly, making it easier to work with evolving and dynamic data. This is particularly useful in scenarios where the data structure is not well-defined or may change frequently.

Redundancy (High Availability):

•*Definition:* Many NoSQL databases are designed with built-in redundancy features to ensure high availability. This involves replicating data across multiple nodes or servers.

•*Benefits:* Redundancy improves fault tolerance. If one node fails, the system can continue to function using data from the remaining nodes. This is critical for systems that require constant availability and minimal downtime.

Role of NoSQL Business Drivers?

The business drivers for adopting NoSQL databases are often characterized by the **Velocity, Agility, Volume, and Variety**.

These drivers highlight the specific challenges and requirements that organizations face in the era of big data and dynamic business environments.

1.Velocity:

1. *Definition:* Velocity refers to the speed at which data is generated, processed, and analyzed.

2. *Role:* NoSQL databases are designed to handle high-velocity data, allowing businesses to capture, store, and process data in real-time or near-real-time. This is crucial for applications that require rapid data ingestion and real-time analytics, such as social media updates, sensor data, or financial transactions.

2.Agility:

1. *Definition:* Agility in the context of databases refers to the ability to **quickly adapt to changing business requirements, data structures, and application needs**.

2. *Role:* NoSQL databases provide schema flexibility, allowing developers to work with varying and evolving data models without the constraints of a fixed schema. This agility is particularly beneficial in dynamic business environments where requirements may change frequently.

1.Volume:

1. *Definition:* Volume refers to the sheer size of data generated and stored by organizations.

2. *Role:* NoSQL databases excel at handling large volumes of data by offering horizontal scalability. As data grows, organizations can add more servers or nodes to the database, ensuring efficient storage and retrieval of massive datasets. This scalability is crucial for applications dealing with big data analytics and large-scale data storage.

2.Variety:

1. *Definition:* Variety refers to the diverse types and formats of data, including structured, semi-structured, and unstructured data.

2. *Role:* NoSQL databases are well-suited for managing diverse data types, such as documents, key-value pairs, column-family data, or graph data. This flexibility allows businesses to handle different data formats within the same database, supporting applications with varied data sources and structures.

MongoDB:

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need.

•MongoDB stores data in flexible, JSON-like documents.

•The document model maps to the objects in your application code, making data easy to work with.

•MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.

•MongoDB's document model is simple for developers to learn and use, while still providing all the capabilities needed to meet the most complex requirements at any scale.