

DAY – 2

MongoDB

What is NoSQL?

NoSQL stands for "Not Only SQL". It represents a class of database systems that provide a mechanism for storage and retrieval of data that is modeled in ways other than tabular relations used in traditional relational databases.

NoSQL databases are designed to:

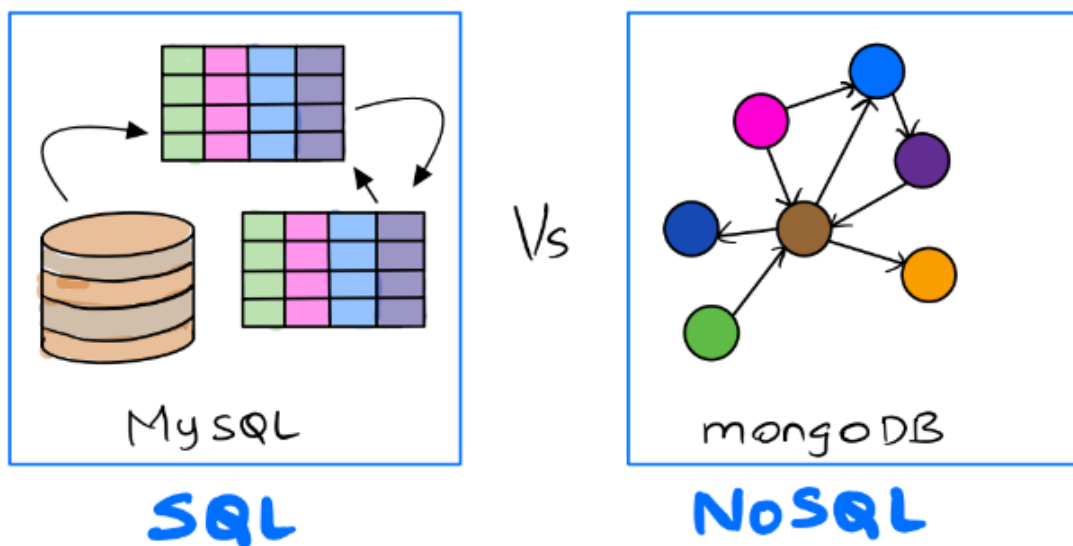
- Handle large volumes of unstructured or semi-structured data
- Support horizontal scaling and distributed architectures
- Offer high performance for read/write operations
- Accommodate changing data models without downtime

Introduction to MongoDB

- MongoDB is a NoSQL database used for storing large volumes of unstructured or semi-structured data.
- Instead of storing data in tables and rows (like relational databases), MongoDB stores data in JSON-like documents.
- MongoDB stores data in flexible documents. Instead of having multiple tables you can simply keep all of your related data together. This makes reading your data very fast.
- MongoDB is a document database which is often referred to as a non-relational database. This does not mean that relational data cannot be stored in document databases.
- It means that relational data is stored differently. A better way to refer to it is as a non-tabular database.
- MongoDB is a **document-oriented NoSQL database** developed by MongoDB Inc. It stores data in **JSON-like documents** called BSON (Binary JSON). MongoDB is popular for:
 - Its flexible schema
 - High performance
 - Scalability
 - Powerful querying and indexing

Feature	Description
Document-Oriented	Stores data as flexible, schema-less documents (BSON – Binary JSON).
High Performance	Fast reads and writes, ideal for real-time applications.
Flexible Schema	Fields can vary across documents; no fixed structure.
Horizontal Scalability	Easy to scale with sharding (data distribution across multiple servers).
Strong Security	Supports authentication, authorization, and encryption.

Why Use NoSQL (MongoDB) Over Traditional SQL Databases



1. Handling Unstructured Data

Traditional relational databases are ideal for structured data with a fixed schema. But real-world data (social media, IoT, logs) is often unstructured. MongoDB handles such data effortlessly due to its schema-less design.

2. Scalability and Performance

Relational databases typically scale **vertically** (adding more power to a single machine), which can be expensive and limited. NoSQL systems like MongoDB scale **horizontally** (adding

3. Better Support for Big Data & Real-Time Analytics

NoSQL databases are optimized for read/write throughput and distributed processing, making them more suitable for big data applications and real-time analytics.

4. Support for Modern Applications

MongoDB is well-suited for:

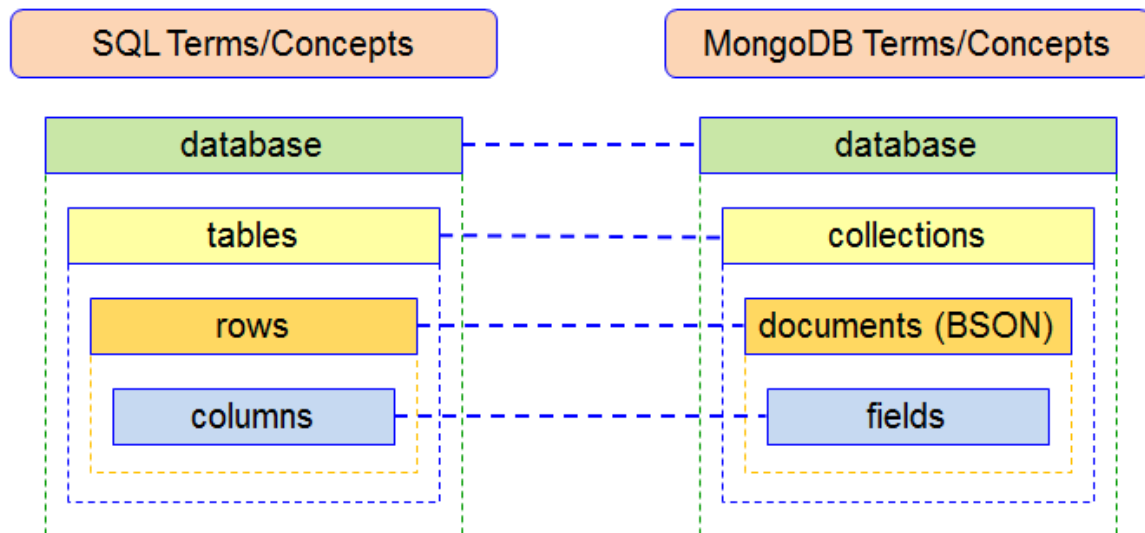
- Real-time analytics dashboards
- Content management systems
- Mobile app backends
- IoT platforms
- Product catalogs and recommendation systems

SQL VS NOSQL(MongoDB)

Both MongoDB and MySQL are popular database management systems (DBMS), but they are built for different purposes and have distinct features.

MongoDB is a NoSQL database, designed for handling unstructured data with high scalability, while MySQL is a traditional relational database management system (RDBMS), ideal for structured data with complex relationships.

Key differences between	
MySQL	MongoDB
Represents data as tables & rows	Represents data as JSON docs.
Must specify tables & columns	Don't need to declare the schema
Allows join operations	Doesn't allow join operations
Utilizes structured query language	Query language is JavaScript
Best for cloud-based services	Best data security is your priority



Use Cases, Pros & Cons – MongoDB

Most common Use Cases of MongoDB

- E-commerce platforms: Storing product catalogs with varying attributes
- Real-time analytics: Log monitoring, dashboards
- Mobile apps: Flexible schemas for user data, preferences
- Content management: Blogs, news articles with different metadata

Advantages of MongoDB

- Flexible document model
- Easy horizontal scaling
- Rich query and indexing support
- Strong community and ecosystem

Limitations of MongoDB

- Not ideal for complex multi-document transactions (though support has improved)
- Lacks ACID compliance for large-scale transactions compared to RDBMS
- Potential for data duplication if schema design isn't optimized
- Higher memory usage due to indexing and BSON overhead