

# Mongo DB

## What is Mongo DB?

- MongoDB is a NoSQL database.
- It stores data as documents (JSON-like format).
- Each document = key-value pairs (like a Python dictionary).
- No need for tables, rows, or fixed columns like SQL.
- It uses BSON (Binary JSON) for storage.

Example:

```
{  
  
  "name": "Elakkiya",  
  
  "age": 21,  
  
  "skills": ["Python", "MongoDB"],  
  
  "address": {  
  
    "city": "Chennai",  
  
    "pin": 600001  
  
  }  
  
}
```

## Advantages of MongoDB

- Flexible – data can change anytime.
- Easy to use – stores data like JSON.
- Works fast with large data.
- Reliable – keeps data safe.
- Good for modern apps like mobile, web, IoT.
- Supports many languages like Python, Java, etc.

## Limitations of Traditional SQL Databases:

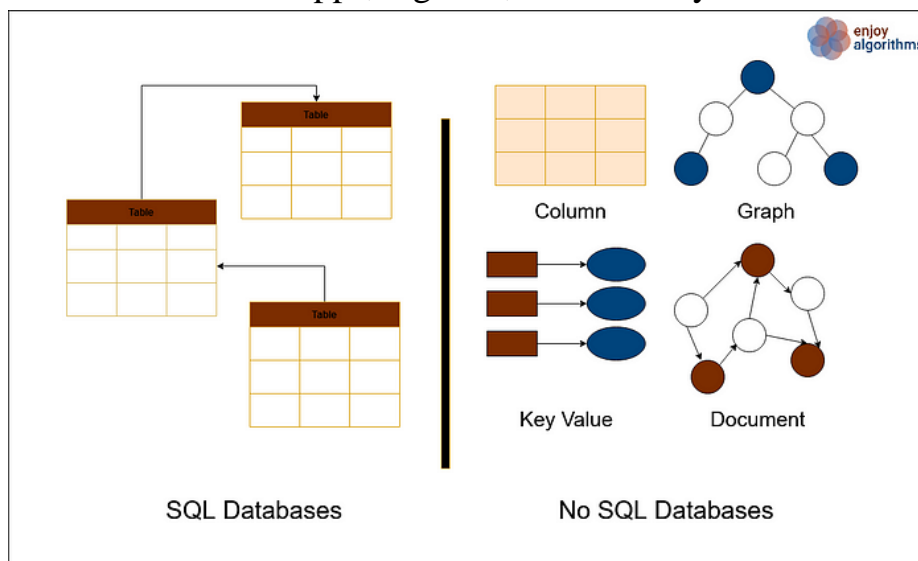
- **Rigid Schemas:** Changing table structures can be complex and slow.
- **Scaling Challenges:** Vertical scaling (adding resources to a single server) is expensive and limited.
- **Complex Joins:** Multi-table joins can slow down queries for big datasets.
- **Unstructured Data:** SQL databases struggle with data types like documents, multimedia, logs.

## What is No SQL?

- NoSQL means “Not Only SQL”.
- It's a type of database that stores data differently from traditional SQL databases.
- Instead of tables and rows, NoSQL stores data as documents, key-value pairs, graphs, or columns.
- Examples: MongoDB

## Why use No SQL?

- **Flexible schemas:** Data can have different formats and change over time.
- **Scalable:** Easy to add more servers and handle big data.
- **Fast:** Good for applications with lots of reads and writes.
- **Works well with unstructured or semi-structured data** like JSON, videos, logs.
- **Great for real-time apps, big data, and cloud systems.**



## Mongo DB vs SQL Databases:

### 1. Data Model

- **MongoDB:**  
Stores data as documents in JSON-like format (called BSON). Each document can have a different structure.
- **SQL:**  
Stores data in tables with fixed rows and columns. Schema must be defined before inserting data.

### 2. Schema

- **MongoDB:**  
Schema-less, so fields can vary between documents in the same collection.
- **SQL:**  
Fixed schema; tables have predefined columns and data types.

### 3. Query Language

- **MongoDB:**  
Uses MongoDB Query Language (MQL), which is JSON-based and supports rich queries on documents.
- **SQL:**  
Uses Structured Query Language (SQL), a powerful language for querying and managing relational data.

### 4. Relationships & Joins

- **MongoDB:**  
Supports embedding related data inside documents or referencing other documents.
- **SQL:**  
Supports complex JOINS across multiple tables to manage relationships efficiently.

## 5. Scalability

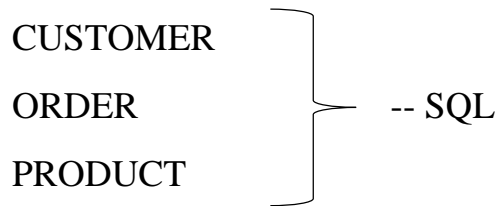
- **MongoDB:**  
Designed for horizontal scaling (sharding) — data is distributed across many servers easily.
- **SQL:**  
Mostly vertical scaling (adding resources to one server); horizontal scaling is complex.

## 6. Use Cases

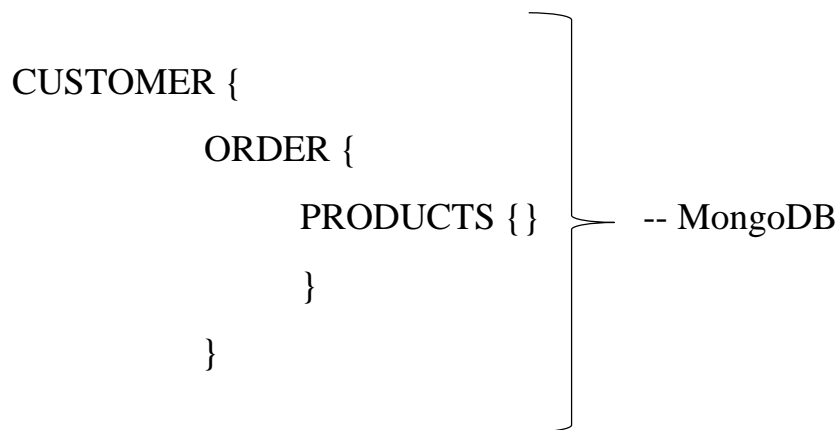
- **MongoDB:**  
Best for flexible, evolving data, big data, real-time analytics, mobile apps, and IoT.
- **SQL:**  
Ideal for structured data, complex queries, banking systems, ERP, and applications needing strong consistency.

Feature	MongoDB	SQL (e.g., MySQL)
Data Model	Document-based (NoSQL)	Table-based (Relational)
Schema	Flexible (schema-less)	Fixed schema (predefined)
Query Language	MQL	SQL
Joins	Limited	Powerful JOIN support
Scalability	Horizontal	Mostly vertical
Use Case	Big data, real-time apps, JSON data	Structured data, transactions

## SIMPLE UNDERSTANDING:



This is how SQL relational database works. It stores data in different tables and links them using foreign keys.



Here everything is nested inside each other. This is how Mongo DB works. It stores all related data together in one document.