**MONGODB AND SQL**

**SQL:**

SQL, or Structured Query Language, is the standard programming language used to manage and manipulate relational databases.

It is foundation for interacting with data in systems like MySQL, Oracle, PostgreSQL, and SQL Server.

SQL is used to store, retrieve, update, and delete data stored in structured tables. These tables are made up of rows and columns, where each column represents a field and each row represents a record.

**Key Features of SQL:**

* Data Definition: Create and modify database structures (tables, views).
* Data Manipulation: Insert, update, delete records (INSERT, UPDATE, DELETE).
* Data Querying: Retrieve data using commands like SELECT, WHERE, JOIN, ORDER BY.
* Transaction Control: Ensures data accuracy and integrity using COMMIT, ROLLBACK, etc.
* Security Management: Controls access to data with GRANT and REVOKE.

**REAL-TIME EXAMPLE:**

**1. E-Commerce Platforms (Product Inventory)**

* In e-commerce applications, SQL databases manage product catalogues, orders, inventory, and customer information:
* **Product Information**: Product details like name, price, and stock quantity are stored in SQL tables.
* **Customer Orders**: When a customer places an order, SQL queries are used to update inventory levels, calculate prices, and process payments.
* **Search and Filter**: SQL is used to retrieve products based on customer preferences, such as price, category, or availability.

**2**. **Healthcare Systems (Patient Data Management)**

SQL is used to manage sensitive data in healthcare systems such as patient records, appointments, and medical history:

* **Patient Records**: SQL databases store patient information, including names, addresses, medical history, and current prescriptions.
* **Appointments**: SQL queries are used to manage patient appointments, availability of doctors, and schedule updates.
* **Medical History**: When a doctor examines a patient, medical history is updated using SQL queries.

**Limitations of SQL Databases:**

Traditional Relational Database Management Systems (RDBMS) like MySQL, Oracle, and SQL Server are based on predefined schemas and store data in rows and columns. While this works well for structured, consistent data, it has key limitations in modern use cases.

**1. Rigid Schema Structure**

In SQL, each table must follow a fixed structure. If your data structure changes (e.g., adding a new column), you must update the schema, which can be time-consuming and error-prone.

**2. Scalability Challenges**

SQL databases scale **vertically** — meaning to handle more data, you need to upgrade the hardware (RAM, CPU). This is expensive and has limits. Horizontal scaling (adding more servers) is complex in SQL systems.

**3. Complex Joins**

Relational data is often split across multiple tables. To combine related data, SQL relies heavily on joins, which become slow and inefficient as the dataset grows.

**4. Not Ideal for Unstructured or Semi-Structured Data**

If you're dealing with data like user logs, social media content, sensor data, or JSON from APIs, SQL databases struggle to store and query this efficiently.

**MONGODB:**

MongoDB is an open-source, document-oriented NoSQL database designed for scalability, performance, and high availability.

Unlike traditional relational databases that store data in rows and tables, MongoDB stores data in flexible, JSON-like documents called BSON.

This structure makes it highly adaptable to modern application needs, especially those involving unstructured or semi-structured data.

It’s known for schema-less design, horizontal scalability, and support for large volumes of data.

**Why MongoDB is Preferred:**

**1. Flexible Schema (Schema-less)**

MongoDB allows you to store documents with varying structures in the same collection. This is ideal for:

* Rapidly changing data models
* Prototyping and startups
* Data from external APIs

**2. JSON-like Documents (BSON)**

MongoDB stores data in BSON format (Binary JSON), which supports:

* Embedded documents
* Arrays
* Complex nested data  
  This removes the need for complex joins.

**3. High Scalability with Sharding**

MongoDB supports horizontal scaling through sharding. You can distribute data across multiple machines, making it easier to handle:

* Big Data
* High-traffic applications
* Global users

**4**. **Built-in Replication and Fault Tolerance**

Using replica sets, MongoDB ensures high availability. Even if one node fails, another can take over.

**5. Developer-Friendly**

MongoDB maps easily to programming languages (like Python, Node.js, Java). Its flexibility allows faster development and deployment.

Difference:

| **Feature** | **SQL Databases** | **MongoDB** |
| --- | --- | --- |
| Schema | Fixed | Dynamic / Flexible |
| Data Model | Tables & Rows | JSON-like Documents |
| Scalability | Vertical | Horizontal (Sharding) |
| Relationships | Join-based | Embedded/Nested |
| Ideal for | Structured data, finance | Real-time, unstructured |
| Performance at scale | Slower with joins | High-speed reads/writes |
| Setup Complexity | Higher | Lower |

BOCK DIAGRAM:

