MongoDB

- **MongoDB** is a **NoSQL database** that stores data in **ON-like documents** (BSON format).
- It is designed for **high performance**, **scalability**, and **flexibility**.
- Unlike traditional relational databases (like MySQL), MongoDB does **not use tables**, **rows**, or **SQL**.

SQL vs MongoDB

SQL Term	MongoDB Term
Database	Database
Table	Collection
Row	Document
Column	Field
Primary Key	_id Field

Features of MongoDB

- **Document-oriented**: Data is stored in BSON documents.
- **Schema-less**: No predefined schema required.
- **Scalable**: Supports horizontal scaling via sharding.
- **Indexing**: Supports primary and secondary indexes for fast searches.
- **High Availability**: Through replication (Replica Sets).
- **Rich Queries**: Supports powerful queries and aggregation framework.

JSON & BSON

- MongoDB internally uses **BSON** (Binary ON) to store documents.

Example JSON document:
JSON

```
{
  "_id": 1,
  "name": "Piyush",
  "age": 30,
  "skills": ["Java", "MongoDB"]
```

```
}
```

MongoDB Architecture

- **Client**: Sends queries.
- **Server**: Manages database and handles requests.
- **Collections**: Groups of related documents.
- **Documents**: The actual data in BSON format.
- **Replica Sets**: Ensure data redundancy and failover.
- **Sharding**: Breaks large data sets into smaller chunks.

Basic MongoDB Commands

1. Database Commands

```
show dbs                    // List databases
use mydb                    // Switch to database
db                          // Show current database
```

2. Collection Commands

```
db.createCollection("users") // Create collection
show collections             // List collections
```

3. Insert Data

```
db.users.insertOne({ name: "Piyush", age: 25 });
db.users.insertMany([
  { name: "Amit", age: 30 },
  { name: "Sagar", age: 22 }
]);
```

4. Find / Query Data

```
db.users.find()                         // Find all
db.users.find({ age: { $gt: 24 } })     // Find where age > 24
```

5. Update Data

```
db.users.updateOne({ name: "Alice" }, { $set: { age: 26 } });
```

6. Delete Data

```
db.users.deleteOne({ name: "Bob" });
```

Operators in MongoDB

- `$gt`, `$lt`, `$gte`, `$lte` – greater/less than
- `$eq`, `$ne` – equal/not equal
- `$in`, `$nin` – in/not in
- `$set` – update field
- `$push`, `$pull` – modify arrays

Indexing

```
db.users.createIndex({ name: 1 });
```

Aggregation Framework

Aggregation is used to process data and return computed results.

```
db.users.aggregate([
  { $match: { age: { $gt: 20 } } },
  { $group: { _id: "$name", total: { $sum: 1 } } }
]);
```

Use Cases of MongoDB

- Real-time analytics
- IoT applications
- Content management systems
- E-commerce product catalogs
- Mobile apps with flexible schema needs

Advantages of MongoDB

- Dynamic schema for flexibility
- High performance on large volumes of unstructured data
- Easy horizontal scaling
- Well-supported by cloud services (like MongoDB Atlas