**DAY-2: MONGO DB:**

**What Is MongoDB?**

MongoDB is a **NoSQL** database that stores data in **JSON-like documents** (BSON format). Unlike traditional relational databases, MongoDB is **schema-less**, making it highly flexible and scalable.

Importance of NOSQL:

**Why MongoDB Is Important**

* **Schema Flexibility**: No need to define rigid table structures.
* **Scalability**: Built for horizontal scaling and high availability.
* **Document-Oriented**: Stores data in rich, nested documents.
* **Real-Time Performance**: Great for applications needing fast reads/writes.
* **Backend Friendly**: Easily integrates with Node.js, Python, Java, etc.

**How MongoDB Works**

MongoDB stores data in **collections**, which are groups of **documents**. Each document is a key-value pair structure similar to JSON.

1. **Client sends request** → via driver (e.g., pymongo)
2. **MongoDB server** → processes the request
3. **Storage engine** → reads/writes data to disk
4. **Response** → returned to client

**Core MongoDB Concepts**

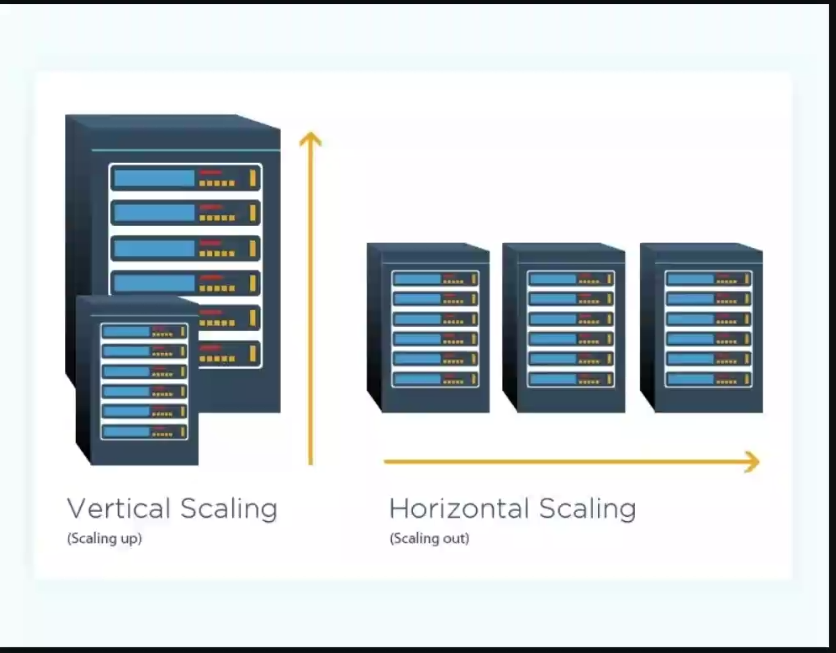
| **Concept** | **Description** |
| --- | --- |
| Database | Container for collections |
| Collection | Group of MongoDB documents |
| Document | JSON-like data structure (BSON) |
| ObjectId | Unique identifier for each document |
| CRUD | Create, Read, Update, Delete operations |
| Index | Improves query performance |
| Aggregation | Powerful data processing pipeline |

***The need of NOSQL databases in present:***



As we see the latest applications and softwares require huge amount of data transaction in just a minute. The traditional database such as relational database cannot handle this situations so where NOSQL databases such as mongo db comes into play.

***SCALING:***



VERTICAL SCALING :

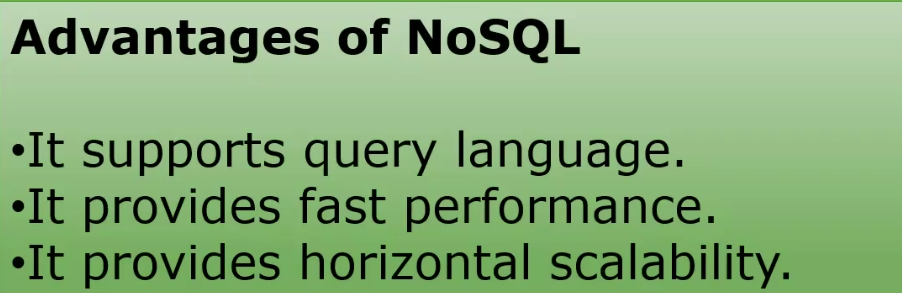
* Upgrading the existing system in terms of hardware

HORIZONTAL SCALING:

* Extending the system power with multiple systems

**-Vertical scaling has limit of upgrading**

**-horizontal scaling has no limit in upgrading, unlimited.**



**Create (Insert Documents)**

Used to add new documents to a collection.

db.students.insertOne({ name: "Anbu", age: 22, course: "Backend Dev" })

db.students.insertMany([

{ name: "Ravi", age: 24, course: "Data Science" },

{ name: "Priya", age: 21, course: "Web Dev" }

])

**Read (Query Documents)**

Used to retrieve documents from a collection.

db.students.find()

db.students.find({ age: { $gt: 22 } })

db.students.findOne({ name: "Anbu" })

db.students.find({}, { name: 1, course: 1, \_id: 0 })

**Update (Modify Documents)**

Used to update existing documents.

db.students.updateOne(

{ name: "Anbu" },

{ $set: { course: "Full Stack Dev" } }

)

db.students.updateMany(

{ age: { $lt: 23 } },

{ $set: { status: "Junior" } }

)

db.students.replaceOne(

{ name: "Ravi" },

{ name: "Ravi", age: 25, course: "AI", status: "Senior" }

**Delete (Remove Documents)**

Used to delete documents from a collection.

db.students.deleteOne({ name: "Priya" })

db.students.deleteMany({ status: "Junior" })