📝 MongoDB

🔷 Introduction

- **MongoDB** is a NoSQL, document-oriented database.
- Stores data in **JSON**.
- Schema-less: documents can have different structures.

🔷 Basic Terminology

RDBMS	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field

🔷 MongoDB Data Types

- String, Number (Int, Long, Double), Boolean
- Array
- Object (Embedded documents)
- Date
- Null
- ObjectId (Unique identifier)

📂 Collection Commands

```
db.createCollection("students")      # Create collection
show collections                     # List collections
db.students.drop()                   # Drop collection
```

📄 Document Commands

```
// Insert
        db.students.insertOne({ name: "John", age: 22, course: "MERN" })
        db.students.insertMany([{ name: "A" }, { name: "B" }])

// Read
        db.students.find()
        db.students.find({ age: { $gt: 18 } })

// Update
        db.students.updateOne({ name: "John" }, { $set: { age: 23 } })
        db.students.updateMany({}, { $set: { active: true } })

// Delete
        db.students.deleteOne({ name: "John" })
        db.students.deleteMany({ active: false })
```

MongoDB Query Operators

There are many query operators that can be used to compare and reference document fields.

Comparison

The following operators can be used in queries to compare values:

- $eq : Values are equal
- $ne : Values are not equal
- $gt : Value is greater than another value
- $gte : Value is greater than or equal to another value
- $lt : Value is less than another value
- $lte : Value is less than or equal to another value
- $in : Value is matched within an array

Logical

The following operators can logically compare multiple queries.

- $and : Returns documents where both queries match
- $or : Returns documents where either query matches
- $not : Returns documents where the query does not match

🧠 **MongoDB Practice Set**

✅ **Task 1: Basic CRUD**

1. Create a database called **school**.
2. Create a collection called **students**.
3. Insert 5 documents with fields: **name, age, course, city**.
4. Find all students from **city "Delhi"**.
5. Update age of student named "**Amit" to 25**.
6. Delete student whose name is **"Ravi"**.

✅ **Task 2: Advanced Queries**

1. Find students with **age > 20 and course = "Web"**.
2. Find students who are not from **"Delhi"**.
3. Add a new field **isActive: true** to all documents.