Table of Content

|  |  |  |
| --- | --- | --- |
| S.NO | Topic | SubTopic |
| 1 | Introduction to MongoDB | NoSQL vs RDBMS, Use Cases, MongoDB Overview |
| 2 | MongoDB Atlas Setup | Cloud setup, Atlas cluster, Compass connection |
| 3 | Local Setup & mongosh | Install locally, use mongosh, CLI basics |
| 4 | Create Database & Collections | Create DB/collections using mongosh |
| 5 | Basic CRUD Operations (Insert) | insertOne, insertMany, document structure |
| 6 | Basic CRUD (Read/Query) | find, findOne, filters, operators |
| 7 | Basic CRUD (Update/Delete) | updateOne, deleteMany, $set, $inc |
| 8 | JSON vs BSON | Structure, datatypes, storage representation |
| 9 | MongoDB Compass | GUI interface, monitoring, DB navigation |
| 10 | Schema-less Design Concepts | Flexibility vs structure, flat vs nested docs |
| 11 | Embedding vs Referencing | 1:1, 1:N, N:N, normalization, denormalization |
| 12 | Indexing Basics | Single, compound, multikey indexes |
| 13 | Advanced Index Types | TTL, Text, Hashed, Geospatial, Wildcard indexes |
| 14 | Aggregation Framework – Part 1 | $match, $group, $project, $lookup, etc. |
| 15 | Aggregation Framework – Part 2 | $sort, $skip, $bucket, $facet |
| 16 | Transactions & ACID | Multi-doc transactions, rollback, sessions |
| 17 | Real-world Use Case Practice | Build schema for cart/blog/e-commerce app |
| 18 | Final Review + Quiz | Mini-project review, schema test, MCQs |

Unit 1: Introduction to MongoDB

# 1.1 What is MongoDB?

**MongoDB** is a **NoSQL**, **document-oriented** database. It stores data in **JSON-like** documents called **BSON** (Binary JSON), making it more flexible than traditional relational databases (RDBMS).

# 1.2 Difference Between MongoDB and RDBMS

|  |  |  |
| --- | --- | --- |
| Feature | MongoDB (NoSQL) | RDBMS (MySQL, PostgreSQL) |
| Data Model | Document-oriented (BSON) | Table-based (rows and columns) |
| Schema | Dynamic (schema-less) | Fixed (predefined schema) |
| Joins | Limited joins (embedded docs or refs) | Supports JOIN operations |
| Scalability | Horizontal (via sharding) | Vertical scaling (limited by hardware) |
| Transactions | Supported (multi-doc since v4.0) | Supported natively |

# 1.3 Why Use MongoDB?

* Flexible schema design
* Developer-friendly (JSON-style)
* High performance and scalability
* Ideal for modern web apps (Node.js, MERN stack)

# 1.4 MongoDB Architecture

MongoDB stores data in this structure:

**MongoDB Server**

└── Database

└── Collection

└── Document (key-value pairs)

**Key Terms:**

* **Database**: A container for collections.
* **Collection**: A group of documents (equivalent to tables).
* **Document**: Actual data, stored in BSON format.

*Example Document:*

{

"name": "John Doe",

"age": 22,

"isStudent": true,

"skills": ["MongoDB", "Node.js", "React"]

}

# 1.4 Key Concepts to Remember

|  |  |
| --- | --- |
| Term | Description |
| NoSQL | Non-relational databases (flexible schema) |
| BSON | Binary JSON – stores MongoDB documents internally |
| Document | Primary unit of data in MongoDB (like a row, but flexible) |
| Collection | A group of related documents (like a table) |
| Schema-less | No fixed structure for fields in documents |

# 1.5 Summary

* MongoDB is a schema-less, NoSQL document database.
* It uses BSON (Binary JSON) for storing documents.
* It’s ideal for modern, scalable applications (especially in the MERN stack).
* Data is stored in collections and documents instead of rows and tables.

Unit 2: MongoDB Atlas Setup

# 2.1 Why MongoDB Atlas?

MongoDB Atlas is a **cloud-based database platform** offered by MongoDB Inc. It helps you:

* Host MongoDB without local installation.
* Easily scale, back up, and manage data.
* Share access with collaborators from anywhere.
* Avoid managing server infrastructure.

## Step 1: Create a MongoDB Atlas Account

1. Go to: <https://www.mongodb.com/cloud/atlas/register>
2. Fill in the details or use Google/GitHub to sign up.
3. After login, click **"Build a Database"**.

## Step 2: Create a Free Shared Cluster

1. Choose:
   * **Cloud Provider**: AWS / GCP / Azure
   * **Region**: Choose nearest to your location (e.g., Mumbai for India)
2. Cluster Tier:  
   Choose the free option → M0 Sandbox (Always Free)
3. Name your cluster (e.g., Cluster0) and click **Create Cluster**.

## Step 3: Configure Security

#### 1. Add a Database User

* Username: student
* Password: strongpassword123 (use your own safe password)

Save this information securely.

#### 2. Add IP Address to Access List

* Choose: **Allow access from anywhere**
* This will set: 0.0.0.0/0

In real-world production apps, **use your current IP only** for better security.

## Step 4: Connect to the Cluster

After the cluster is ready:

1. Click **"Connect"** → Choose **"Connect using MongoDB Compass"**.
2. Copy the **Connection String** like:

mongodb+srv://student:strongpassword123@cluster0.mongodb.net/

## Step 5: Install and Use MongoDB Compass

#### Install MongoDB Compass:

Download from: <https://www.mongodb.com/try/download/compass>

Install and open Compass.

#### Connect:

1. Paste the connection URI into Compass.
2. Click **Connect**.

You will see your cluster and databases visually!

# 2.2 Bonus: Modify Connection URI

To connect via terminal (mongosh), use a modified URI like:

mongosh "mongodb+srv://student:<password>@cluster0.mongodb.net/myDatabase"

Replace <password> and myDatabase as needed.

# 2.3 Summary Table

|  |  |
| --- | --- |
| Step | Action |
| Create Atlas Account | Sign up at mongodb.com |
| Build Free Cluster | M0 cluster (AWS/GCP/Azure) |
| Add DB User | Create username + password |
| Whitelist IP | Use 0.0.0.0/0 for demo purposes |
| Get Connection String | Use in Compass or shell |
| Install Compass | GUI client to connect and view documents |
| Create DB/Collection | Add data manually via Compass |

Unit 3 MongoDB Set Up

# 3.1 MongoDB installation in Windows

## Step 1: Download MongoDB Community Server

1. Visit the official MongoDB download page:  
   👉 <https://www.mongodb.com/try/download/community>
2. Choose the following options:
   * **Version**: Latest Stable (e.g., 6.x)
   * **Platform**: Windows
   * **Package**: .msi (Windows Installer)
3. Click **Download** and wait for the file to finish.

## Step 2: Run the MongoDB Installer

1. **Double-click** the .msi file you downloaded.
2. In the **Setup Wizard**, click **Next**.
3. **Choose Setup Type**:
   * Select **Complete** (recommended for beginners).
4. **Service Configuration**:
   * Keep default settings:
     + **Install MongoDB as a Service**
     + Run service as **Network Service User**
5. Optional: You can also install **MongoDB Compass** (GUI tool) from the same wizard.
6. Click **Install** and allow the installer to finish.

## Step 3: Verify MongoDB Installation

After installation, MongoDB binaries are usually located at:

C:\Program Files\MongoDB\Server\<version>\bin

To make the command-line tools accessible from any folder, **add this bin path to the System Environment Variables**:

### Add to PATH:

1. Open Windows Search → type “Environment Variables”.
2. Click “Edit the system environment variables”.
3. Click **Environment Variables** button.
4. Under “System variables”, select Path, then click **Edit**.
5. Click **New**, then add the path:

C:\Program Files\MongoDB\Server\6.0\bin

Adjust version number (6.0) based on what you installed.

1. Click OK → OK → OK to save and exit.

## Step 4: Test Installation (via Terminal)

Open **Command Prompt (cmd)** and type:

mongod --version

If installed correctly, you should see version info like:

db version v6.0.6

To test MongoDB server is running:

mongod

This starts the server. You will see logs like:

Waiting for connections on port 27017

Leave this window open while MongoDB is running.

## Step 5: Open a New Terminal and Use mongosh

MongoDB now uses mongosh (Mongo Shell) for interacting with the database.

Open a new command prompt window and run:

mongosh

You’ll see:

Current Mongosh Log ID: ...

Using MongoDB: 6.0.x

test>

You're now connected to the MongoDB shell and ready to start using it!

## Default Data Storage Location

When MongoDB runs, it stores data at:

C:\data\db

If this folder doesn’t exist, create it manually:

mkdir C:\data\db

MongoDB won’t start unless this folder exists.

## Summary

|  |  |
| --- | --- |
| Step | Description |
| Download Installer | From MongoDB official website |
| Install MongoDB | Complete setup with default service |
| Add to PATH | So you can run commands from anywhere in terminal |
| Start MongoDB Server | mongod |
| Connect with Shell | mongosh |
| Test Insert & Query | Use basic CRUD in MongoDB shell |

# **3.2 Mongoosh**

**Goal**: Set up MongoDB on your local system and interact with it using the **MongoDB Shell (mongosh)**.

## What is mongosh?

* mongosh is the **official MongoDB shell**, used to interact with your MongoDB database from the command line.
* It supports:
  + CRUD operations
  + Admin tasks
  + Writing JavaScript in shell
  + Connecting to local or remote MongoDB instances

## Step-by-Step Installation (Windows)

If you’ve already installed MongoDB (from Unit 1), skip to Step 5.

### Step 1: Download MongoDB Community Edition

* Go to: <https://www.mongodb.com/try/download/community>
* Choose:
  + **Platform**: Windows
  + **Package**: .msi
* Click **Download** and install it.

### Step 2: Install MongoDB

1. Launch the .msi file.
2. Select **Complete Setup**.
3. Check the box to install **MongoDB as a service**.
4. (Optional) Choose to install **MongoDB Compass**.
5. Complete the installation.

### Step 3: Set MongoDB in PATH

To use MongoDB from any terminal window:

1. Go to:  
   C:\Program Files\MongoDB\Server\<version>\bin
2. Copy the path and add it to **System Environment Variables**:
   * Search: “Edit the system environment variables”
   * Go to: Environment Variables → Path → Edit → New → Paste the path
   * Click OK and apply changes

### Step 4: Create MongoDB Data Directory

MongoDB stores its data in a folder. You need to create it:

mkdir C:\data\db

This is the default directory for MongoDB.

### Step 5: Start MongoDB Server

1. Open **Command Prompt** and run:

mongod

If successful, you’ll see logs saying:

Waiting for connections on port 27017

Your database server is now running.

Keep this terminal open! It’s running your database server.

### Step 6: Open a New Terminal and Launch mongosh

Now, open **another Command Prompt** and run:

mongosh

This connects to your local MongoDB server.

## Basic Commands to Practice

Once inside mongosh, try the following:

### 1. Create/Select a Database

use bookstore

If the database doesn’t exist, MongoDB creates it when you first insert data.

### 2. Insert a Document

db.books.insertOne({

title: "Clean Code",

author: "Robert Martin",

pages: 464

})

### 3. View Documents

db.books.find()

## Useful mongosh Commands

|  |  |
| --- | --- |
| Command | Description |
| show dbs | List all databases |
| use <dbName> | Switch to a database |
| show collections | List collections in the current DB |
| db.collection.insertOne() | Insert a single document |
| db.collection.find() | View all documents |
| db.collection.drop() | Delete a collection |
| exit | Exit the shell |

## Summary Table

|  |  |
| --- | --- |
| Task | Command / Action |
| Start MongoDB Server | mongod |
| Open MongoDB Shell | mongosh |
| Switch DB | use myDB |
| Insert Data | db.myCollection.insertOne({}) |
| View Data | db.myCollection.find() |
| Stop Server | Close the terminal where mongod is running |