



# MONGO DB



Pallavi Mahalatkar

# INTRODUCTION

MongoDB is a leading open-source NoSQL database known for its high performance, flexibility, and ease of scalability. Unlike traditional relational databases that use tables and rows, MongoDB stores data in dynamic, JSON-like documents, which allows for more versatile data models and structures. This document-oriented storage model makes MongoDB ideal for handling large volumes of unstructured or semi-structured data, such as data generated by modern web and mobile applications.

The database was developed by MongoDB Inc., originally released in 2009, and has since grown to become one of the most widely used NoSQL databases in the world. Its ability to store data in a format that closely resembles objects in application code makes it very popular among developers, simplifying data integration and manipulation.

One of the standout features of MongoDB is its flexible schema. Unlike relational databases that require a fixed schema, MongoDB allows for schema evolution, meaning the structure of the documents can change over time. This flexibility can significantly speed up development cycles, as developers can modify the data model without major disruptions.

Security is a critical aspect of MongoDB, which offers features like authentication, authorization, encryption, and auditing to protect sensitive data. It also supports role-based access control (RBAC) to fine-tune user permissions and ensure secure access to the database.

In summary, MongoDB's combination of a flexible data model, powerful querying and aggregation capabilities, robust scalability, and comprehensive security features make it a compelling choice for modern application development. Its ability to efficiently handle diverse and evolving data types positions it as a versatile solution for a wide range of use cases, from simple web applications to complex, data-intensive systems.

## Why Mongo DB ?

MongoDB is a leading open-source NoSQL database known for its high performance, flexibility, and ease of scalability. Unlike traditional relational databases that use tables and rows, MongoDB stores data in dynamic, JSON-like documents, which allows for more versatile data models and structures. This document-oriented storage model makes MongoDB ideal for handling large volumes of unstructured or semi-structured data, such as data generated by modern web and mobile applications.

## Where mongo DB is used ?

- Handling Large Volumes of Data
- **Scalability:**
- Flexible Schema Design
- High Performance
- Real-Time Analytics
- Developer Productivity
- Cloud Integration
- Big Data Applications
- Content Management Systems
- **Internet of Things (IoT)**
- E-Commerce
- Mobile Applications

In summary, MongoDB is used in scenarios where data flexibility, scalability, high performance, and developer productivity are critical. Its versatile nature allows it to be applied across various industries and use cases, making it a popular choice for modern application development.

## How to Download Mongo DB ?

Downloading and installing MongoDB involves several steps, depending on your operating system. Below are the instructions for downloading and installing MongoDB on Windows, macOS, and Linux.

### For Windows:

1. **Visit the MongoDB Download Center:**
  - Go to the [MongoDB Download Center](#).
2. **Select the MongoDB Version:**
  - Choose the "Community" server.
  - Select your version, platform (Windows), and package (MSI).
3. **Download the Installer:**
  - Click the "Download" button to download the MSI installer.
4. **Run the Installer:**
  - Double-click the downloaded MSI file to run the installer.
  - Follow the prompts in the setup wizard. You can use the default settings, but make sure to select the option to install MongoDB as a service.
5. **Configure MongoDB:**
  - During the installation, you might be prompted to choose a setup type. Select "Complete."
  - Optionally, install MongoDB Compass, a GUI for MongoDB, if prompted.
6. **Verify the Installation:**
  - Open Command Prompt and run `mongod --version` to verify that MongoDB is installed correctly.

## Common Steps After Installation:

### 1. Check MongoDB Status:

- For Windows: Use the Services application to check if MongoDB service is running.
- For macOS and Linux: Run `sudo systemctl status mongod` to check the status of MongoDB.

### 2. Access MongoDB Shell:

- Open a new terminal or command prompt window and run:

```
sh
Copy code
mongo
```

- This command starts the MongoDB shell, where you can interact with your database.

### 3. Configuration and Data Directory:

- MongoDB stores data in `/data/db` by default. Ensure this directory exists and is writable. You can configure different paths in the `mongod.conf` file.

By following these steps, you can successfully download, install, and start using MongoDB on your system.

THIS PICTORIAL EXAMPLES MAKE US UNDERSTAND HOW MANOGO DB WORKS



HENCE DB SOFTWARE LIKE ORACLE, MONGO



1. Database softwares are like rack builders
2. Oracle, Mongo builds and gives you just use it

HOW CAN A MEDICAL PERSON SEARCHES EXACTLY?



1. Some organised system to store
2. Take time to organise

Let us see so examples by executing the programs

## Lets Load the document

- Download the student csv from this [link](#)
- Import the data to the collection created [link](#)
- You should be able to see the uploaded data in mongo compass

By using the above links create a database in mongo db compass and save the csv file with data base name db and collection name student.

## Few Commands to test after connections

Command	Expected Output	Notes
show dbs	admin 40.00 KiB config 72.00 KiB db 128.00 KiB local 40.00 KiB	All Databases are shown
use db	switched to db db	Connect and use db
show collections	Students	Show all tables
db.foo.insert({"bar" : "baz"})		Insert a record to collection. Create Collection if not exists

### Explanation :

1.db>show dbs

This command lists all the databases present on the MongoDB server.

Output :

- admin 40.00 KiB
- config 108.00 KiB
- db 56.00 KiB
- local 72.00 KiB

Each line shows the name of a database and its size .for example,

The 'db' database.

2.db>use db:

Output:



already on db db

This message indicates that the shell is already using the DB database. If it wasn't, the output would have been switched to db db

### 3. **db> show collections:**

- This command lists all collections in the current database (db).
- **Output:** student
- This indicates that there is one collection named student in the db database.

### 4. **db> db.foo.insert({"bar" : "baz"}):**

- This command inserts a document with the content {"bar" : "baz"} into the foo collection of the db database.
- **Deprecation Warning:**
  - Deprecation Warning: Collection.insert() is deprecated. Use insert One, insert Many, or bulk Write.
  - This warning informs you that the insert() method is deprecated and suggests using insert One, insert Many, or bulk Write instead.

**Output:**

- `acknowledged: true`
  - This indicates that the insert operation was acknowledged by the database.
- `insertedIds: { '0': ObjectId('6658acbf6fd215de8acdcd6') }`
  - This shows the IDs of the inserted documents. Since only one document was inserted, there's only one ID: `ObjectId('6658acbf6fd215de8acdcd6')`



















