# Mongo DB

# Report

## Introduction:

## What is Database?

`A database is a collection of data that is organized, which is also called structured data. It can be accessed or stored in a computer system. It can be managed through a Database Management System (DBMS), a software used to manage data. Database refers to related data in a structured form.



## How do Document Databases Work?

A document database has information retrieved or stored in the form of a document or other words semi-structured database. Since they are non-relational, so they are often referred to as NoSQL data.

The document database fetches and accumulates data in forms
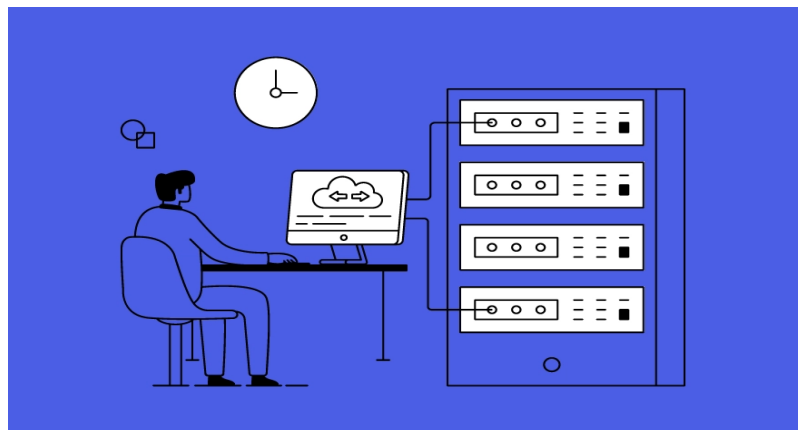


of key-value pairs but here, the values are called as Documents. A document can be stated as a complex data structure. Document here can be a form of text, arrays, strings, JSON, XML, or any such format. The use of nested documents is also very common. It is very effective as most of the data created is usually in the form of JSON and is unstructured.

# What Is Database Management?

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database.

DBMS provides an environment to store and retrieve the data in coinvent and efficient manner.

Key Features of DBMS

> **Data modeling:** A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.

> **Data storage and retrieval:** A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.

> **Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.

> **Data integrity and security:** A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.

> **Backup and recovery:** A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.

> **DBMS can be classified into two types:** Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL)

> **RDBMS:** Data is organized in the form of tables and each table has a set of rows and columns. The data are related to each other through primary and foreign keys.

- ➢ **NoSQL:** Data is organized in the form of key-value pairs, documents, graphs, or column-based. These are designed to handle large-scale, high-performance scenarios.

## MongoDB:

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON ( similar to JSON format).

## Features of Mongo DB:

- ➢ **Document Oriented**: MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.

- ➢ **Indexing**: Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.

- ➢ **Scalability**: MongoDB scales horizontally using sharding (partitioning data across various servers). Data is partitioned into data chunks using the shard

key, and these data chunks are evenly distributed across shards that reside across many physical servers. Also, new machines can be added to a running database.

➢ **Replication and High Availability**: MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.

➢ **Aggregation**: Aggregation operations process data records and return the computed results. It is similar to the GROUPBY clause in SQL. A few aggregation expressions are sum, avg, min, max, etc

## Installation:

## Requirements to Install MongoDB on Windows:

➢ MongoDB 4.4 and later only support 64-bit versions of Windows.
➢ MongoDB 7.0 Community Edition supports the following 64-bit versions of Windows on x86_64 architecture:
  o Windows Server 2022
  o Windows Server 2019
  o Windows 11
➢ Ensure that the user is running mongod and mongos has the necessary permissions from the following groups:
  o Performance Monitor Users
  o Performance Log Users

# Steps to Install MongoDB on Windows using MSI:

To install MongoDB on Windows, first, download the MongoDB server and then install the MongoDB shell. The Steps below explain the installation process in detail and provide the required resources for the smooth download and install MongoDB.
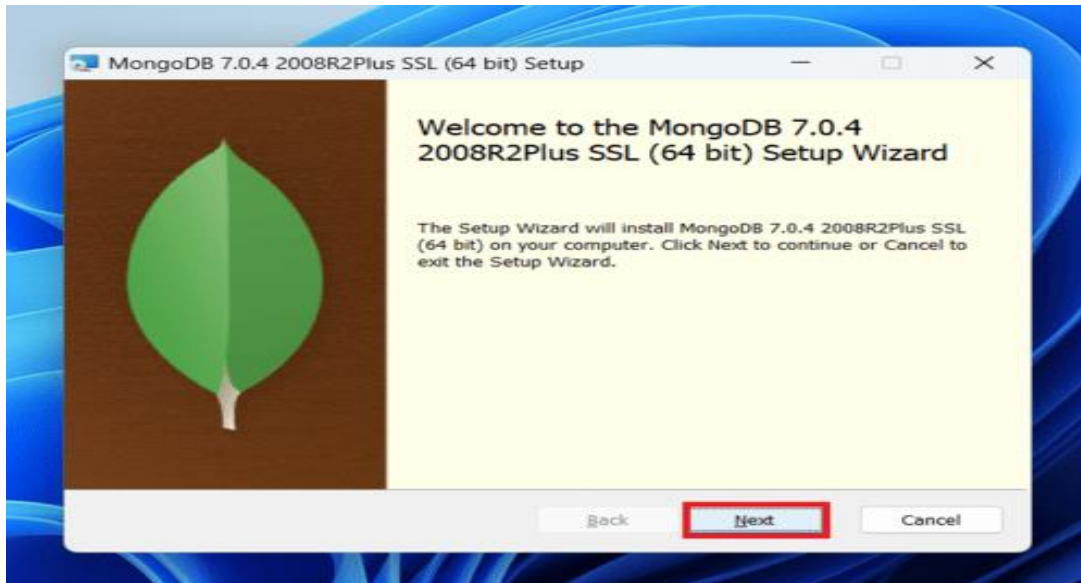
**Step 1:** Go to the MongoDB Download Center to download the MongoDB Community Server.
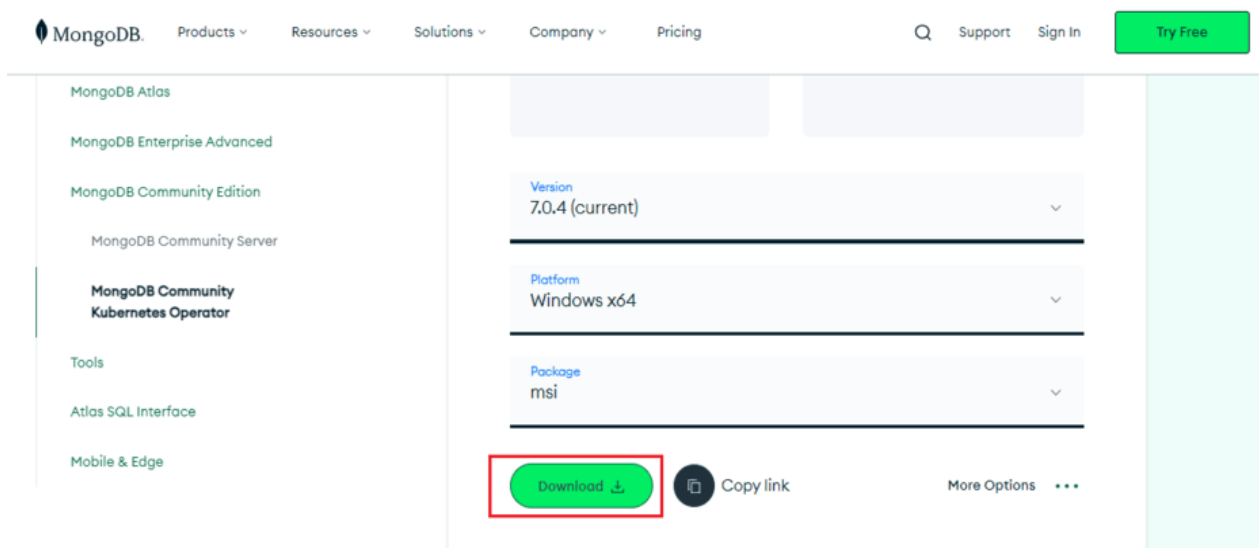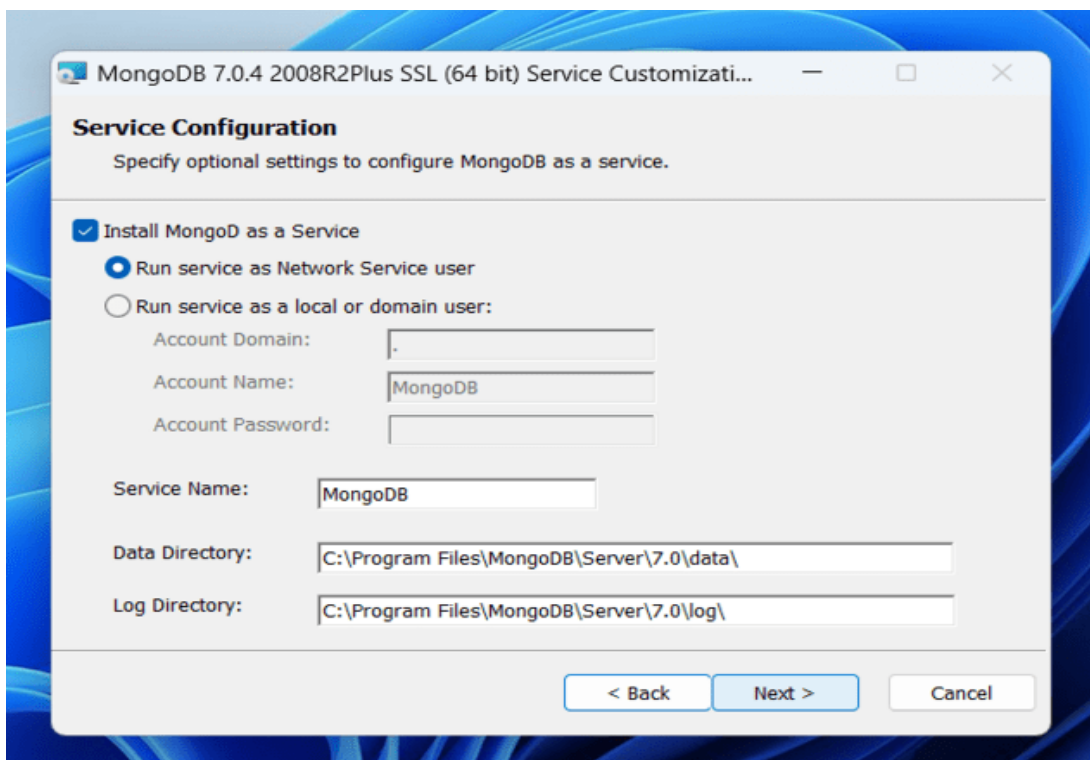


Here, You can select any version, Windows, and package according to your requirement. For Windows, we need to choose:

- Version: 7.0.4
- OS: Windows x64
- Package: msi

**Step 2:** When the download is complete open the msi file and click the *next button* in the startup screen:



**Step 3:** Now accept the End-User License Agreement and click the next button:

**Step 4:** Now select the *complete option* to install all the program features. Here, if you can want to install only selected program features and want to select the location of the installation, then use the *Custom option:*
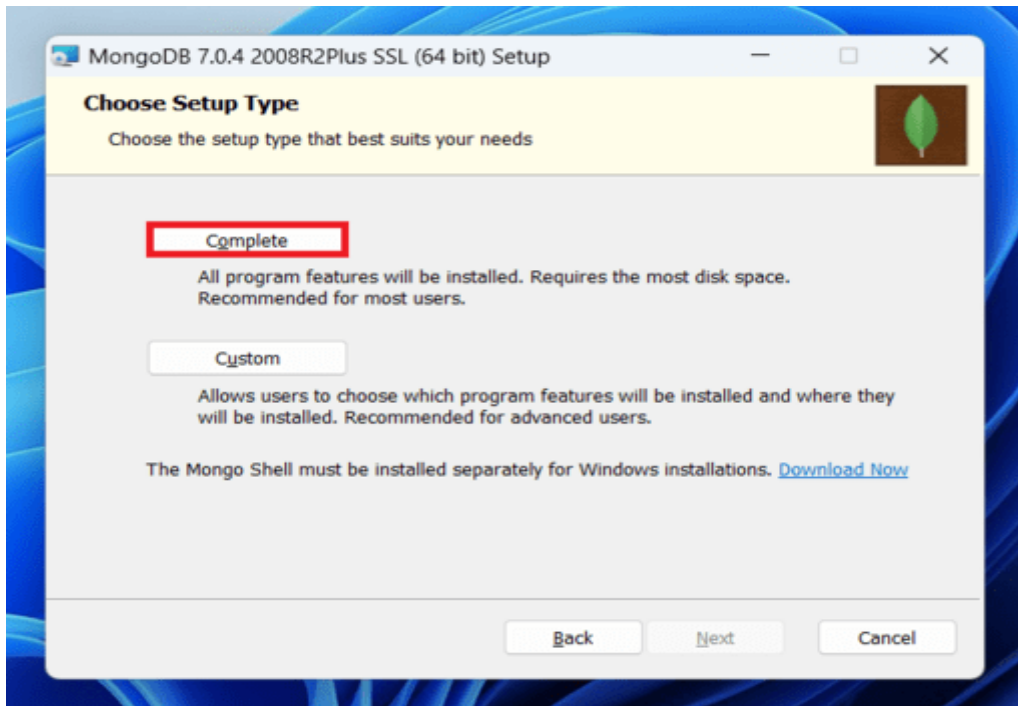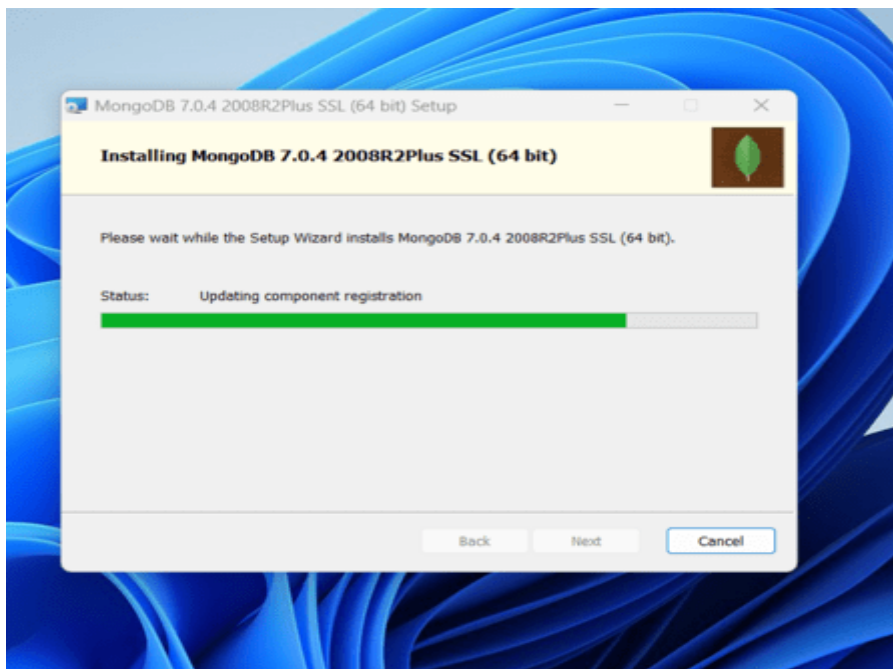


**Step 5:** Select "Run service as Network Service user" and copy the path of the data directory. Click Next:
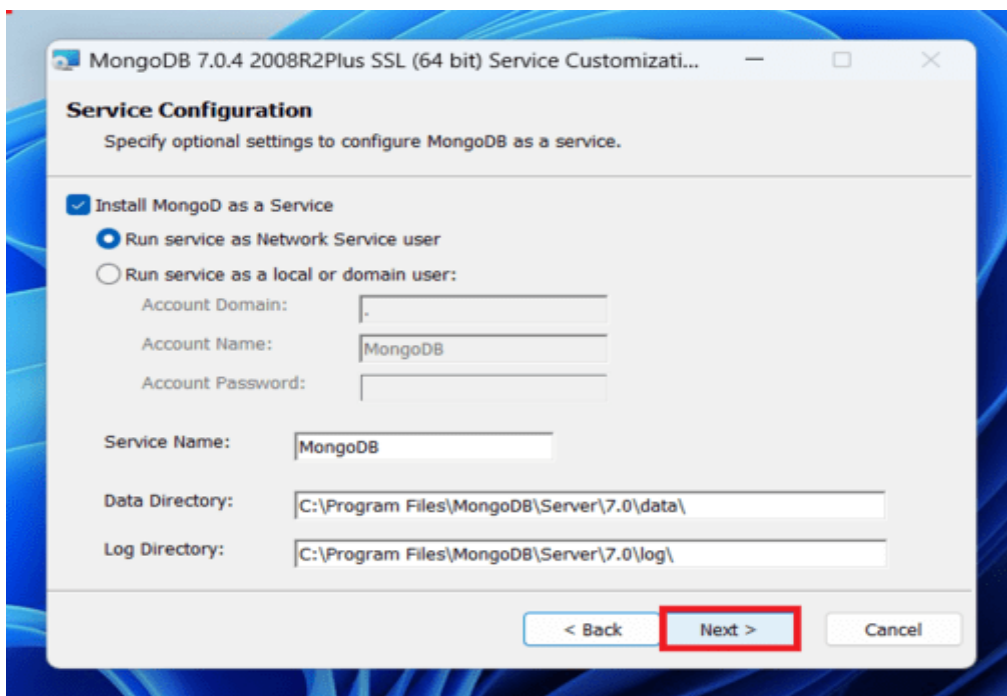
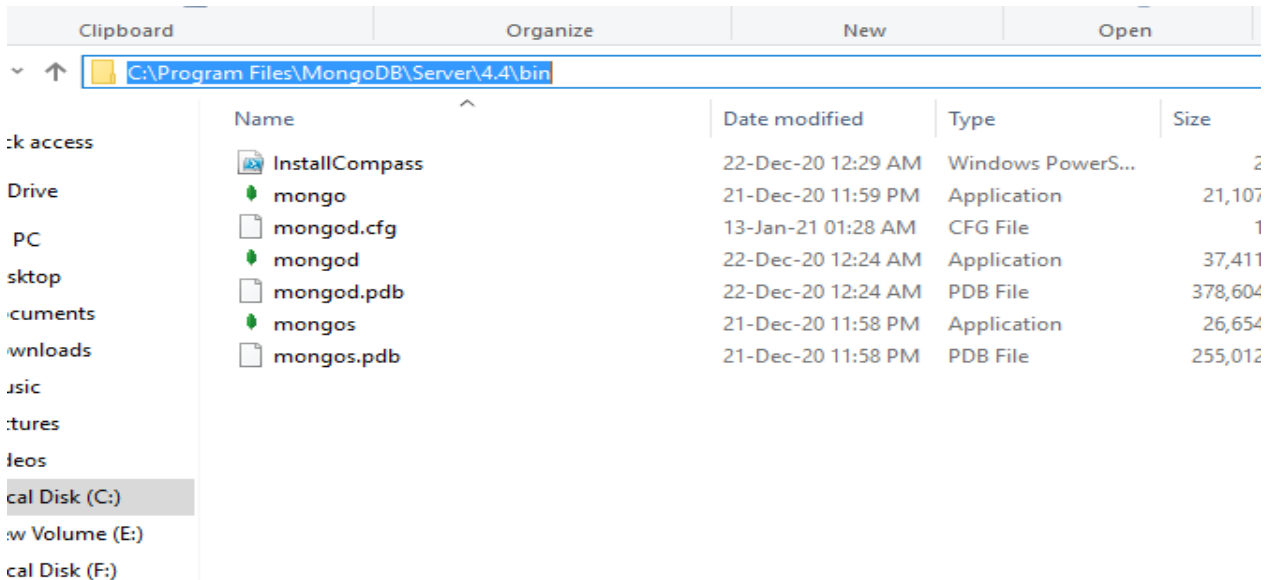**Step 6:** Click the *Install button* to start the MongoDB installation process:
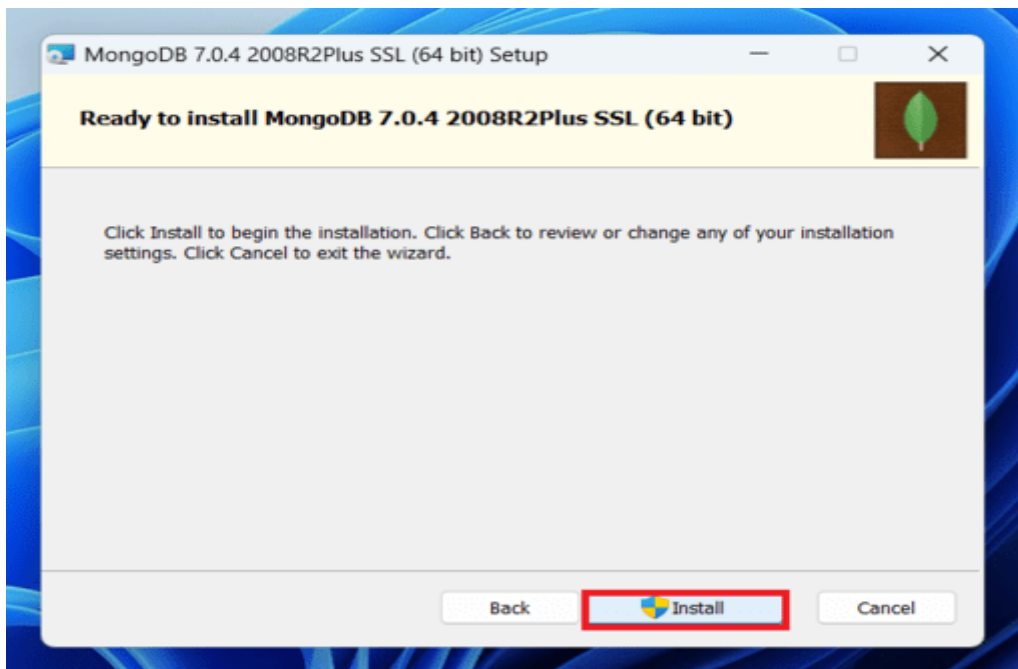


**Step 7:** After clicking on the install button installation of MongoDB begins:

**Step 8:** Now click the *Finish button* to complete the MongoDB installation process:

**Step 9:** Now we go to the location where MongoDB installed in step 5 in your system and copy the bin path:



**Step 10:** Now, to create an environment variable open system properties >> Environment Variable >> System variable >> path >> Edit Environment variable and paste the copied link to your environment system and click Ok:
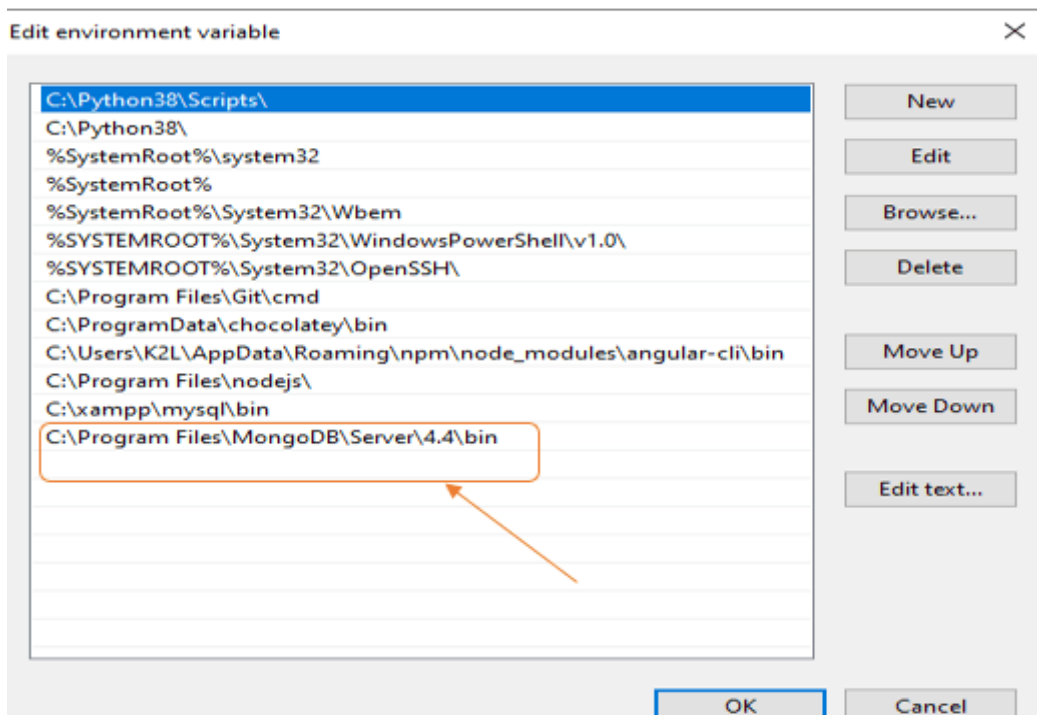
**Step 11:** After setting the environment variable, we will run the MongoDB server, i.e. mongod. So, open the command prompt and run the following command:

**mongod**

When you run this command you will get an error i.e. *C:/data/db/ not found*.

**Step 12:** Now, Open C drive and create a folder named "data" inside this folder create another folder named "db". After creating these folders. Again open the command prompt and run the following command:

**mongod**

Now, this time the MongoDB server(i.e., mongod) will run successfully.

## Run mongo Shell:

**Step 13:** Now we are going to connect our server (mongod) with the mongo shell. So, keep that mongod window and open a new command prompt window and write <u>mongo.</u> Now, our mongo shell will successfully connect to the mongod.

## Run MongoDB:

Now you can make a new database, collections, and documents in your shell

## Run Commands:

### Show Databases:

To display the database you are using, type db:

**show dbs**

The operation should return test, which is the default database.

### Switch Databases:

To switch databases, issue the `use <db>` helper :

**use<database>**

### Show All Tables:

To show all tables in database ,issue the following command:

**show collections**

### Insertion Of Record Or Data:

Insert a data into collection or table and also create collection if it is not exist by issuing following command:

**<database_name>.<collection_name>.insert({attribute:value})**

### Insertion Of Multiple Record Or Data:

To insert multiple data into collection ,we issue following command:

**<database_name>.<collection_name>.batchInsert([{attribute1:value1},{attribute2:value2}………..,{attribute-n:value-n}])**

### Print All The Rows:

To print the rows of tables following command is issued as follows:

**<database_name>.<collection_name>.find()**

This function also can be used to get particular row from table later on we will come across this.

# Databases:

In MongoDB, databases hold one or more collections of documents. To select a database to use, in `mongosh`, issue the `use <db>` statement

# Collections:

MongoDB stores documents in collections. Collections are analogous to tables in relational databases.

```
{
    na    {
    ag       na   {
    st       ag       name: "al",
    gr       st       age: 18,
}            gr       status: "D",
             }        groups: [ "politics", "news" ]
                  }
```

Collection

# Datatypes:

MongoDB Server stores data using the JSON format which supports some additional data types that are not available using the JSON format.

## Date:

`mongosh` provides various methods to return the date, either as a string or as a `Date` object:

- Date() method which returns the current date as a string.

- new Date() :constructor which returns a Date object using the ISODate() wrapper.

- ISODate() :constructor which returns a Date object using the ISODate() wrapper.

## Int32:

If a number can be converted to a 32-bit integer, mongosh will store it as Int32. If not, mongosh defaults to storing the number as a Double. Numerical values that are stored as Int32 in mongosh would have been stored by default as Double in the mongo shell.

## Long:

The Long() constructor can be used to explicitly specify a 64-bit integer.

## Decimal128:

Decimal128() values are 128-bit decimal-based floating-point numbers that emulate decimal rounding with exact precision.