REPORT ON MONGODB

MongoDB®

INTRODUCTION

MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and current versions are licensed under the Server Side Public License (SSPL). MongoDB is a member of the MACH Alliance.

MongoDB supports field, range query and regularexpression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

Fields in a MongoDB document can be indexed with primary and secondary indices.

WHAT IS DATABASE?

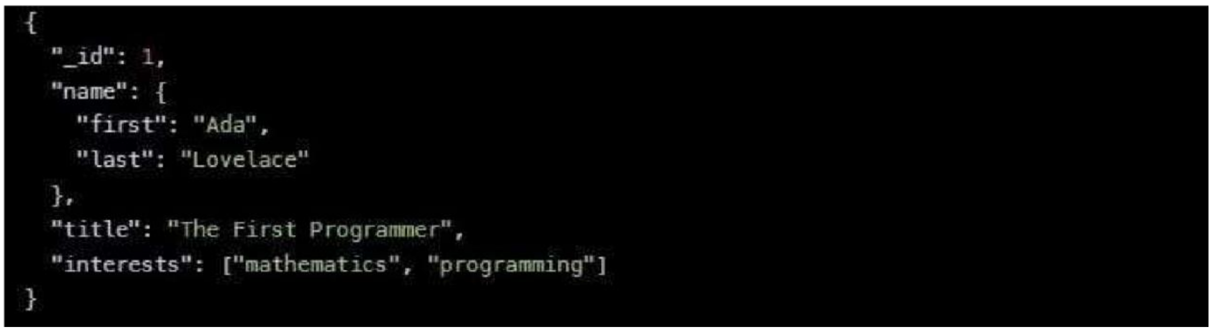
The software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a database system. Often the term "database" is also used loosely to refer to any of the DBMS, the database system or an application associated with the database.

What is MongoDB?

MongoDB is a document database built on a horizontal scale-out architecture that uses a flexible schema for storing data. Founded in 2007, MongoDB has a worldwide following in the developer community.

Instead of storing data in tables of rows or columns like SQL database, each record in a MongoDB database is a document described in BSON, a binary representation of the data. Applications can then retrieve this information in a JSON format.

Here’s a simple JSON document describing a historical figure:



```
{
  "_id": 1,
  "name": {
    "first": "Ada",
    "last": "Lovelace"
  },
  "title": "The First Programmer",
  "interests": ["mathematics", "programming"]
}
```

Document databases are highly flexible, allowing variations in the structure of documents and storing documents that are partially complete. One document can have others embedded in it. Fields in a document play the role of columns in a SQL database, and like columns, they can be indexed to increase search performance.

From its founding, MongoDB was built on a scale-out architecture, a structure that allows many small machines to work together to create fast systems and handle huge amounts of data.

MongoDB has always focused on providing developers with an excellent user experience, which, in addition to all its other properties, has made MongoDB a favorite of developers worldwide for a wide variety of applications.

Why Use MongoDB?

MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas.

As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. This format directly maps to native objects in most modern programming languages, making it a natural choice for developers, as they don't need to think about normalizing data. MongoDB can also handle high volume and can scale both vertically or horizontally to accommodate large data loads.

MongoDB was built for people building internet and business applications who need to evolve quickly and scale elegantly. Companies and development teams of all sizes use MongoDB for a wide variety of reasons.

DOCUMENTAL MODEL

The document data model is a powerful way to store and retrieve data in any modern programming language, allowing developers to move quickly.

Deployment Options

MongoDB is available in any major public cloud (such as AWS, Azure, and Google Cloud) through MongoDB Atlas, in large data centers through the Enterprise Advanced edition, or free through the sourceavailable Community edition.

Get Started Quickly

MongoDB has a great user experience for developers who can install MongoDB and start writing code immediately.

Fully Scalable

MongoDB's horizontal, scale-out architecture can support huge volumes of both data and traffic.

Find Community

MongoDB has developed a large and mature platform ecosystem. It has a worldwide community of developers and consultants, making it easy to get help. It also has an enterprise-grade support offering.

Using MongoDB enables your team to go further and faster when developing software applications that handle data of all sorts in a scalable way.

MongoDB is an excellent choice if you need to: Support rapid iterative development. Enable collaboration of a large number of teams. Scale to high levels of read and write traffic. Scale your data repository to a massive size. Evolve the type of deployment as the business changes. Store, manage, and search data with text, geospatial, or time-series dimensions.

MongoDB as a company has grown because the number of use cases with these characteristics continues to grow.

What are the Advantages of MongoDB?

MongoDB has become one of the most wanted databases in the world because it makes it easy for developers to store, manage, and retrieve data when creating applications with most programming languages.

To understand whether MongoDB is right for you, let’s look at the advantages of MongoDB for developers. You can also check out the top five MongoDB features.

The Power of Document-Oriented Databases

MongoDB is the pioneer of what has come to be called NoSQL databases, which developed because RDBMS systems based on SQL did not support the scale or rapid development cycles needed for creating modern applications.

NoSQL is an umbrella term; it includes document-oriented databases like MongoDB, columnar databases, in-memory databases, and more.

In MongoDB, records are stored as documents in compressed BSON files. The documents can be retrieved directly in JSON format, which has many benefits:

It is a natural form to store data.

It is human-readable.

Structured and unstructured information can be stored in the same document.

You can nest JSON to store complex data objects.

JSON has a flexible and dynamic schema, so adding fields or leaving a field out is not a problem.

Documents map to objects in most popular programming languages.

Most developers find it easy to work with JSON because it is a simple and powerful way to describe and store data.

Perhaps most importantly, the developer controls the database schema. Developers adjust and reformat the database schema as the application evolves without the help of a database administrator. When needed, MongoDB can coordinate and control changes to the structure of documents using schema validation.
MongoDB created Binary JSON format (BSON) to support more data types than JSON. This new format allows for faster parsing of the data. Data stored in BSON can be searched and indexed, tremendously increasing performance. MongoDB supports a wide variety of indexing methods, including text, decimal, geospatial, and partial.

MONGODB 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:

Windows Server 2022

Windows Server 2019

Windows 11

Ensure that the user is running `mongod` and `mongos` has the necessary permissions from the following groups:

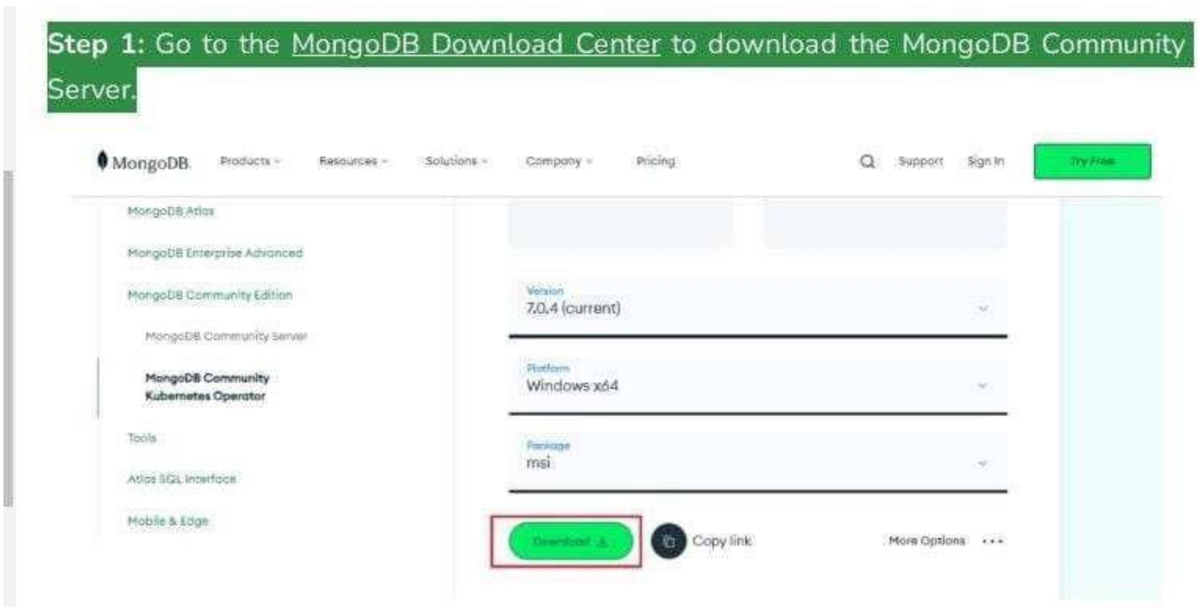
Performance Monitor Users

Performance Log Users

STEPS TO INSTALL MONGODB:

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.


Step 1: Go to the MongoDB Download Center to download the MongoDB Community Server.

Version: 7.0.4 (current)

Platform: Windows x64

Package: msi

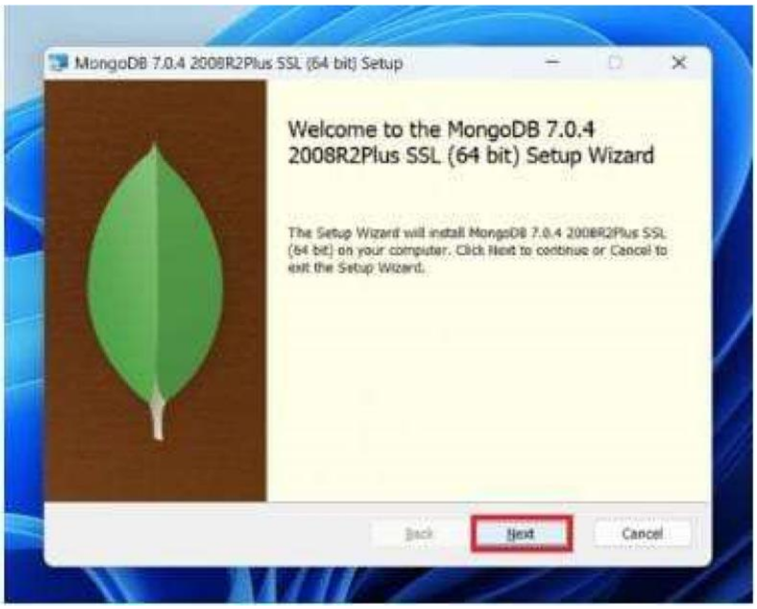
Here, You can select any version, Windows, and package according to your requirement. For Windows, we need to choose:

Version: 7.0.4

OS: Windows x64

Package: msi

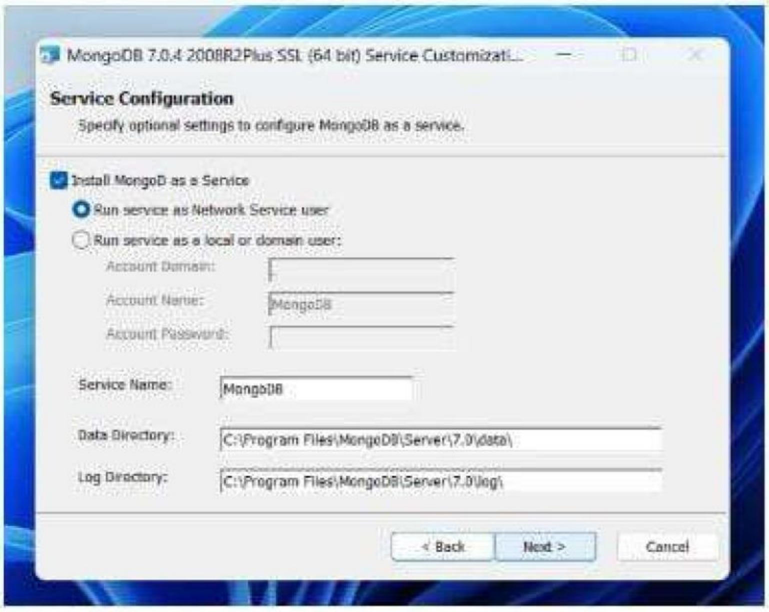
Step 2: When the download is complete open the msi file and click the *next button* in the startup screen:


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 7.0.4 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

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


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Service Customizati...

Service Configuration

Specify optional settings to configure MongoDB as a service.

[✓] Install MongoD as a Service

[● Run service as Network Service user]

○ Run service as a local or domain user:

Account Domain:

Account Name: MongoDB

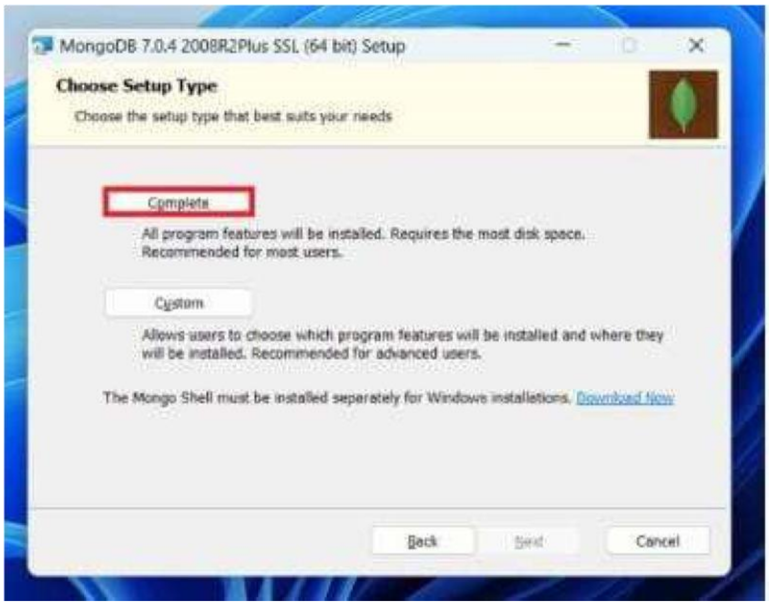
Account Password:

Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\7.0\data\

Log Directory: C:\Program Files\MongoDB\Server\7.0\log\

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:*


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

Choose the setup type that best suits your needs

Complete

All program features will be installed. Requires the most disk space. Recommended for most users.

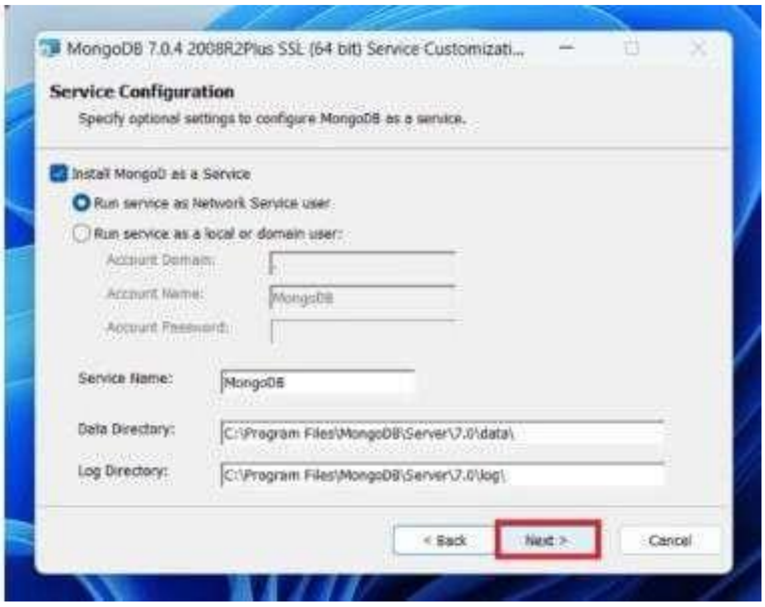
Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

The Mongo Shell must be installed separately for Windows installations. Download Now

Back Next Cancel

Step 5: Select "Run service as Network Service user" and copy the path of the data directory. Click Next:


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Service Customizati...

Service Configuration

Specify optional settings to configure MongoDB as a service.

[✓] Install MongoD as a Service

[●] Run service as Network Service user

[ ] Run service as a local or domain user:

Account Domain: .

Account Name: MongoDB

Account Password:

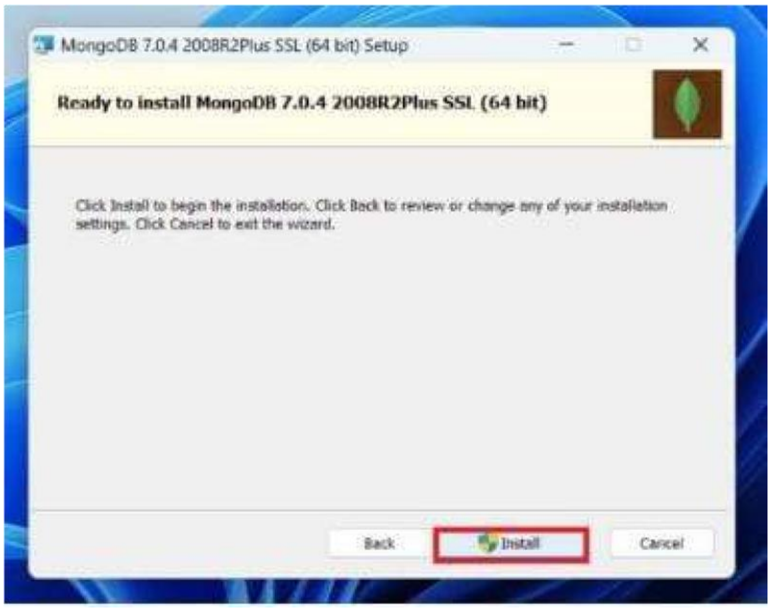
Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\7.0\data\

Log Directory: C:\Program Files\MongoDB\Server\7.0\log\

< Back Next > Cancel

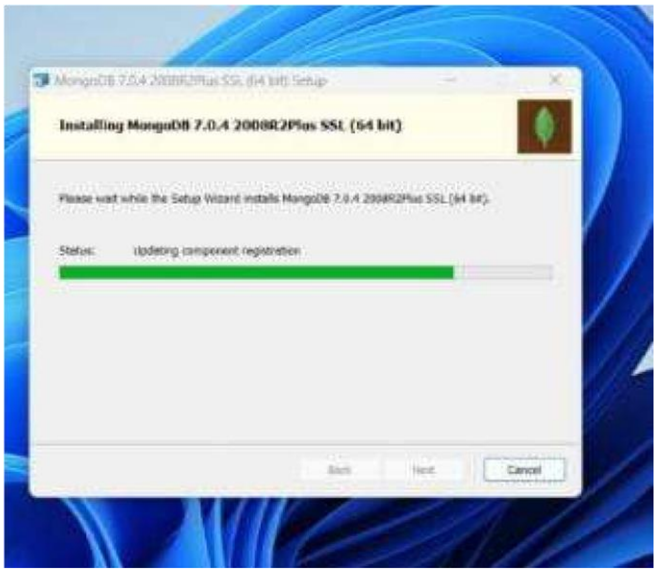
Step 6: Click the *Install button* to start the MongoDB installation process:


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 7.0.4 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

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


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

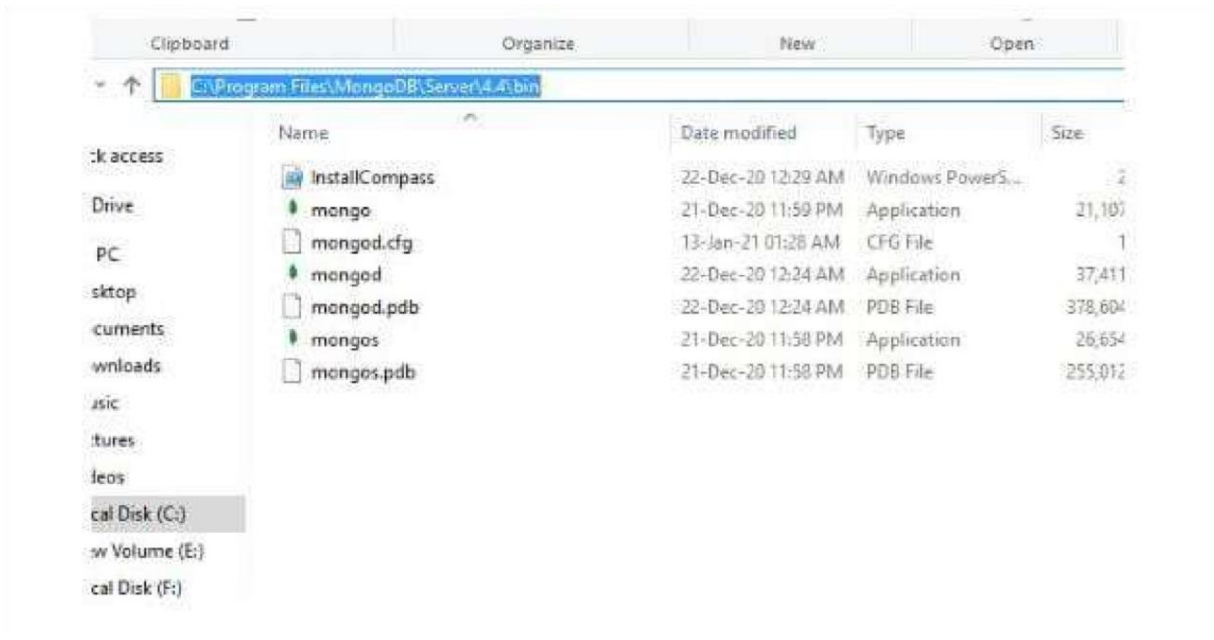
Installing MongoDB 7.0.4 2008R2Plus SSL (64 bit)

Please wait while the Setup Wizard installs MongoDB 7.0.4 2008R2Plus SSL (64 bit).

Status: Updating component registration

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

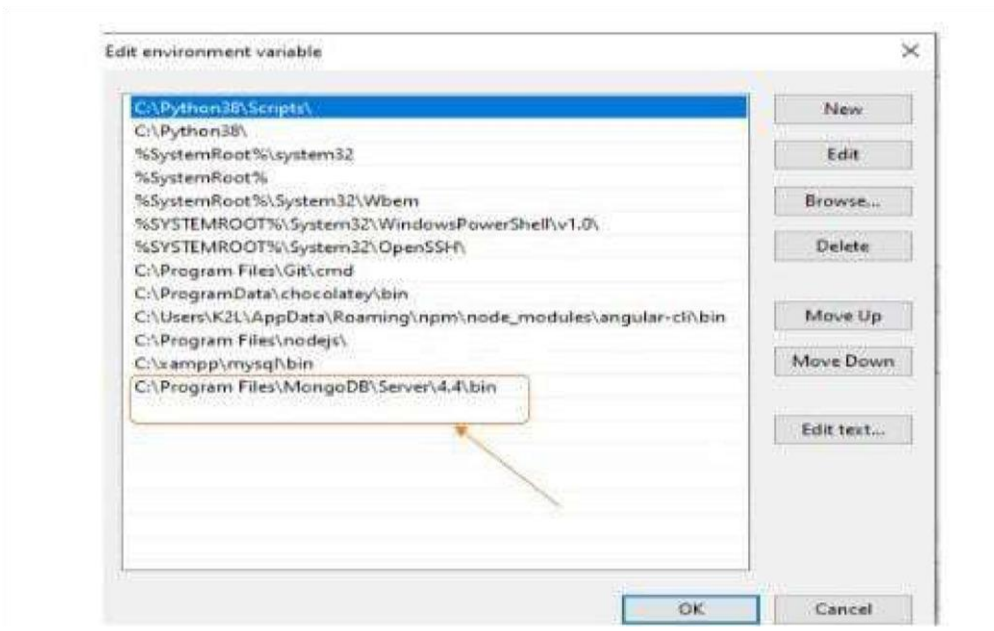
Step 9: Now we go to the location where MongoDB installed in step 5 in your system and copy the bin path:



C:\Program Files\MongoDB\Server\4.4\bin

Name	Date modified	Type	Size
InstallCompass	22-Dec-20 12:29 AM	Windows PowerS...	2
mongo	21-Dec-20 11:59 PM	Application	21,10
mongod.cfg	13-Jan-21 01:28 AM	CFG File	1
mongod	22-Dec-20 12:24 AM	Application	37,411
mongod.pdb	22-Dec-20 12:24 AM	PDB File	378,60
mongos	21-Dec-20 11:58 PM	Application	26,65
mongos.pdb	21-Dec-20 11:58 PM	PDB File	255,01

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:



```
C:\Python38\Scripts\
C:\Python38\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Git\cmd
C:\ProgramData\chocolatey\bin
C:\Users\K2L\AppData\Roaming\npm\node_modules\angular-cli\bin
C:\Program Files\nodejs\
C:\xampp\mysql\bin
C:\Program Files\MongoDB\Server\4.4\bin
```

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**

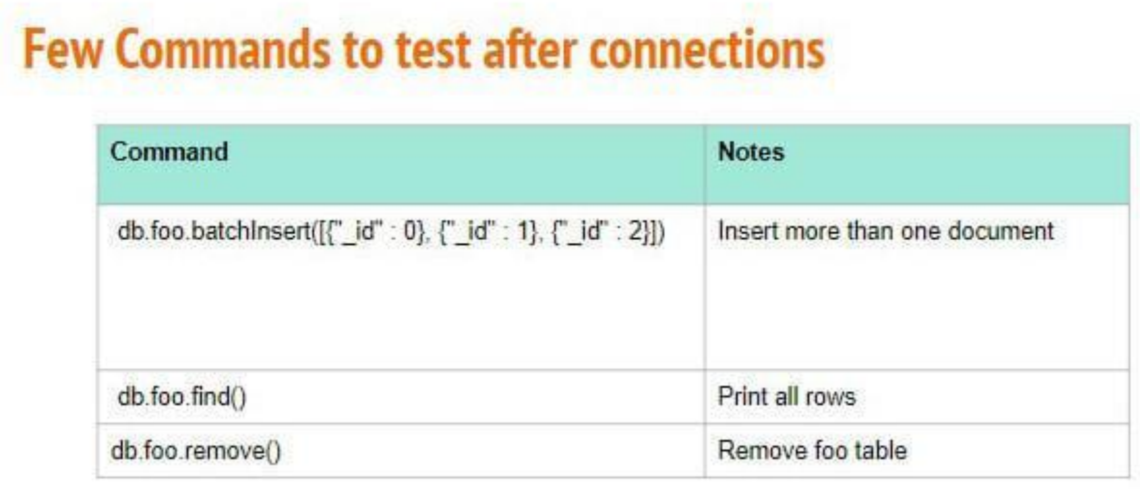
When you run this command you will get an error i.e. *C:/data/db/ not found*.

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
```

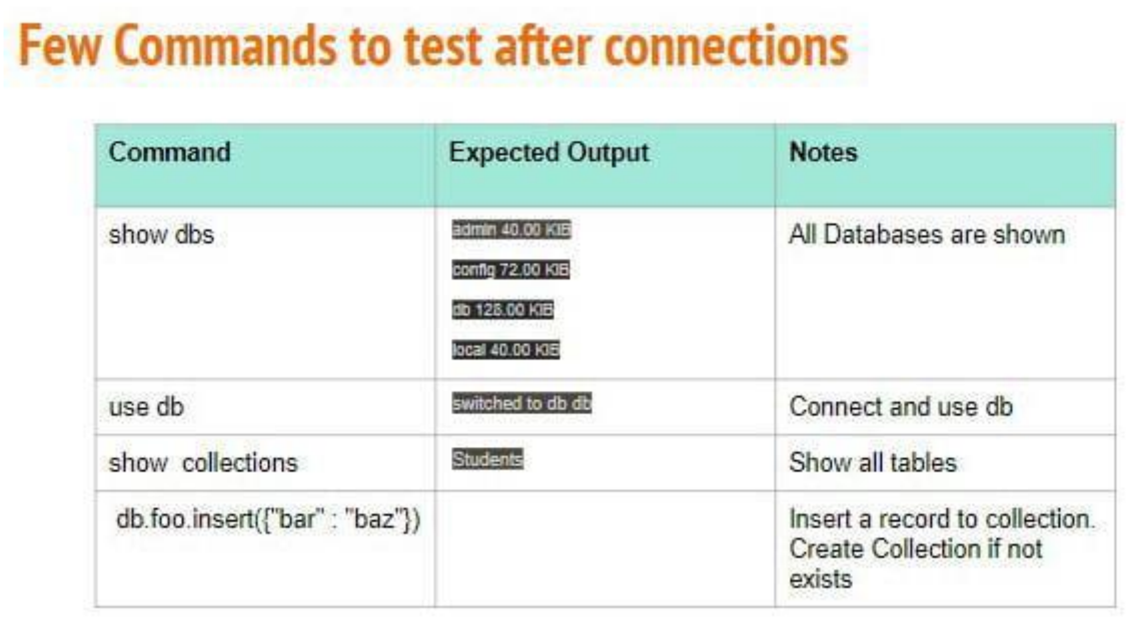
Now, this time the MongoDB server(i.e., mongod) will run successfully.

FEW COMMANDS:



Few Commands to test after connections

Command	Notes
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows
db.foo.remove()	Remove foo table



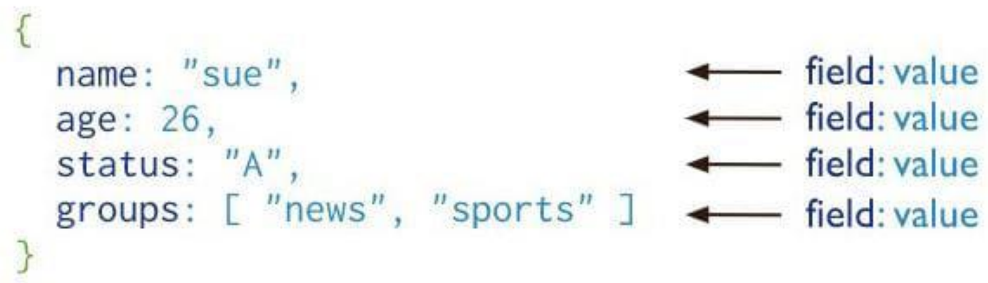
Few Commands to test after connections

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

DOCUMENTS,COLLECTONS AND DATATYPES:

DOCUMENTS:

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.



```
{
   name: "sue",
   age: 26,
   status: "A",
   groups: [ "news", "sports" ]
}
```

The advantages of using documents are:

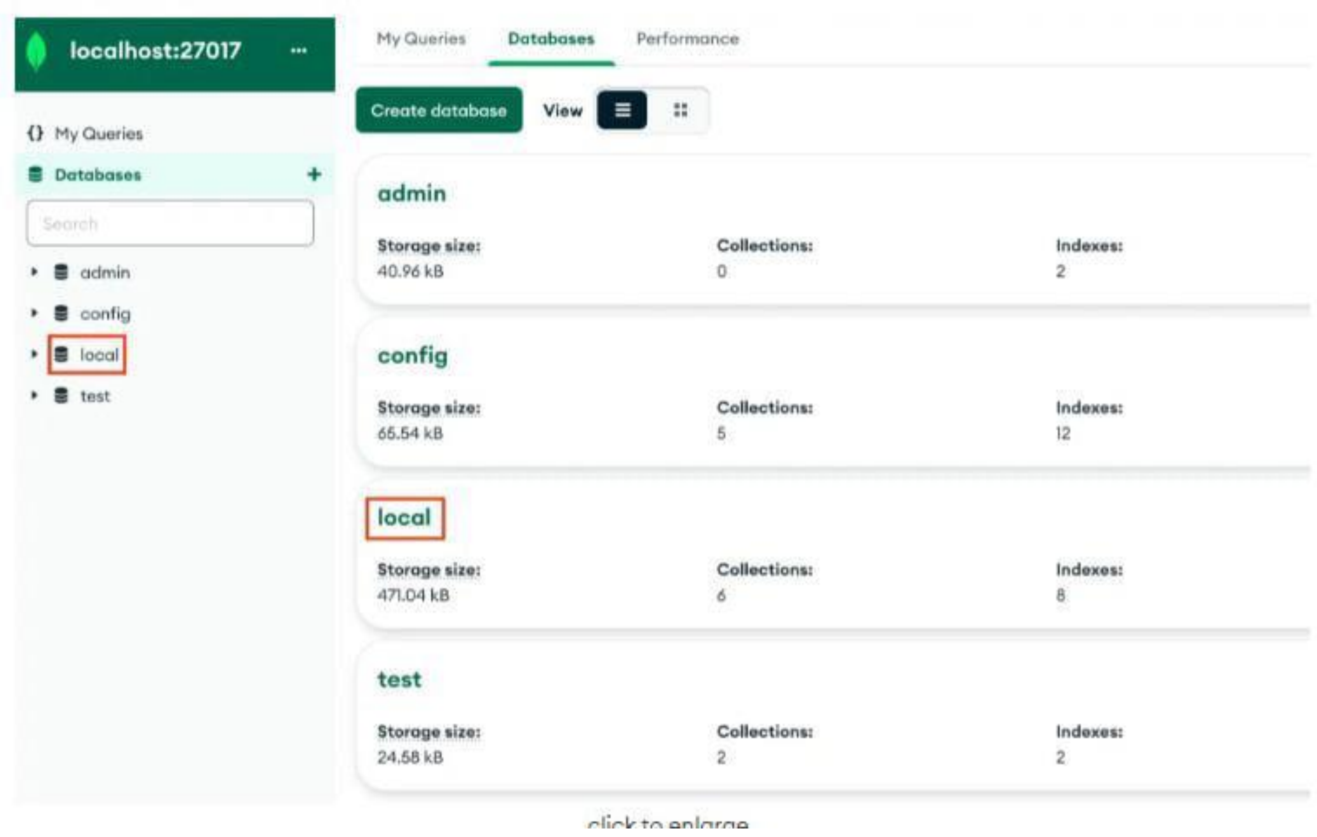
1.Documents correspond to native data types in many programming languages.

2.Embedded documents and arrays reduce need for expensive joins.

3.Dynamic schema supports fluent polymorphism.

COLLECTIONS :

A collection is a grouping of MongoDB documents. Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single database



admin
Storage size: 40.96 kB
Collections: 0
Indexes: 2

config
Storage size: 65.54 kB
Collections: 5
Indexes: 12

local
Storage size: 471.04 kB
Collections: 6
Indexes: 8

test
Storage size: 24.58 kB
Collections: 2
Indexes: 2

click to enlarge

DATATYPES:

The datatypes are:

1.Date :

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

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

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

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

2.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.

The Int32() constructor can be used to explicitly specify 32-bit integers.

```
db.types.insertOne(

{

"_id": 1,

"value": Int32("1"),

"expectedType": "Int32"

}

)
```

3.**Decimal :**

Decimal values are 128-bit decimal-based floating-point numbers that emulate ecimal rounding with exact precision.

This functionality is intended for applications that handle monetary data, such financial, tax, and scientific computations.

The Decimal **BSON type** uses the IEEE 754 decimal128 floating-point numberin
format which supports 34 decimal digits (i.e. significant digits) and an exponen
range of −6143 to +6144.

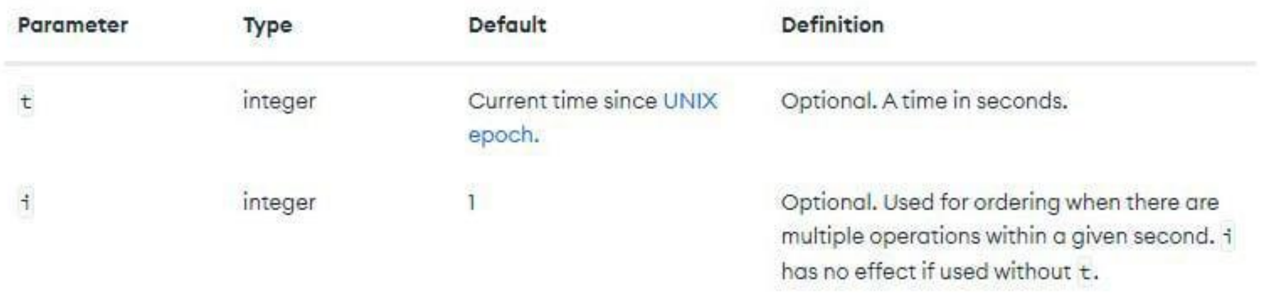
```
db.types.insertOne(

{

"_id": 5,

"value": Decimal128("1"),

"expectedType": "Decimal128"

}

)
```

4.Timestamp:

MongoDB uses a BSON Timestamp internally in the oplog.
The Timestamp type works similarly to the Java Timestamp type. Use the Date type for operations involving dates.

A Timestamp signature has two optional parameters.

```
Timestamp( { "t": <integer>, "i": <integer> } )
```



Parameter	Type	Default	Definition
`t`	integer	Current time since UNIX epoch.	Optional. A time in seconds.
`i`	integer	1	Optional. Used for ordering when there are multiple operations within a given second. `i` has no effect if used without `t`.

DATABASES:

In MongoDB, databases hold one or more collections of documents.

To select a database to use, log in to Atlas and do the following:

1.Navigate to the *Collections* tab.

2.Select the database from the list of databases in the left pane.

Create a Database

To create a new database, log in to Atlas and do the following:

1.Navigate to the *Collections* tab.

2.Click *Create Database*.

3.Enter the *Database Name* and the *Collection Name*.

Enter the database and the collection name to create the database and its first collection.

Click *Create*.

Upon successful creation, the database and the collection displays in the left pane in the Atlas UI.