Big Data Analytics

Experiment No. 01

Aim:- To install and configure MongoDB to execute NoSQL commands.

Theory:-

What is MongoDB?
MongoDB is a NoSQL (Not only SQL) database that stores large volumes of data in the form of documents. MongoDB removes the concept of "rows" of conventional and relational data models by introducing "documents." This offers the developers the flexibility to work with evolving data models.

What is MySQL?
MySQL is a free, open-source, relational database management system that stores data in the form of tables containing rows and columns. It uses RDBMS to ensure referential integrity between the rows of a table and interprets queries to fetch information from the database.

MySQL vs. MongoDB: One-on-one Comparision
Now that you know the objectives of these database management systems, let's look at some of the differences between them.

Feature	MySQL	MongoDB
Data Structure	It stores each individual record as a table cell with rows and columns	It stores unrelated data in JSON-like documents

Schema	MySQL requires a schema definition for the tables in the database	MongoDB doesn't require any prior schema
Languages	Supports Structured Query Language (SQL)	Supports JSON Query Language to work with data
Foreign Key	Supports the usage of Foreign keys	Doesn't support the usage of Foreign keys
Replication	Supports master-slave replication and master-master replication	Supports sharding and replication
Scalability	SQL Database can be scaled vertically	MongoDB database can be scaled both vertically and horizontally

Join Operation	Supports Join operation	Doesn’t support Join operation
Performance	Optimized for high performance joins across multiple tables	Optimized for write performance
Risks	Prone to SQL injection attack	Since there’s no schema, lesser risks involved
Community Support	There are currently (always increasing) about 222k repositories and 7Million commits on GitHub for support on MySQL	There are currently (always increasing) about 177k repositories and 923k commits on GitHub for support on MongoDB

Which One to Choose?

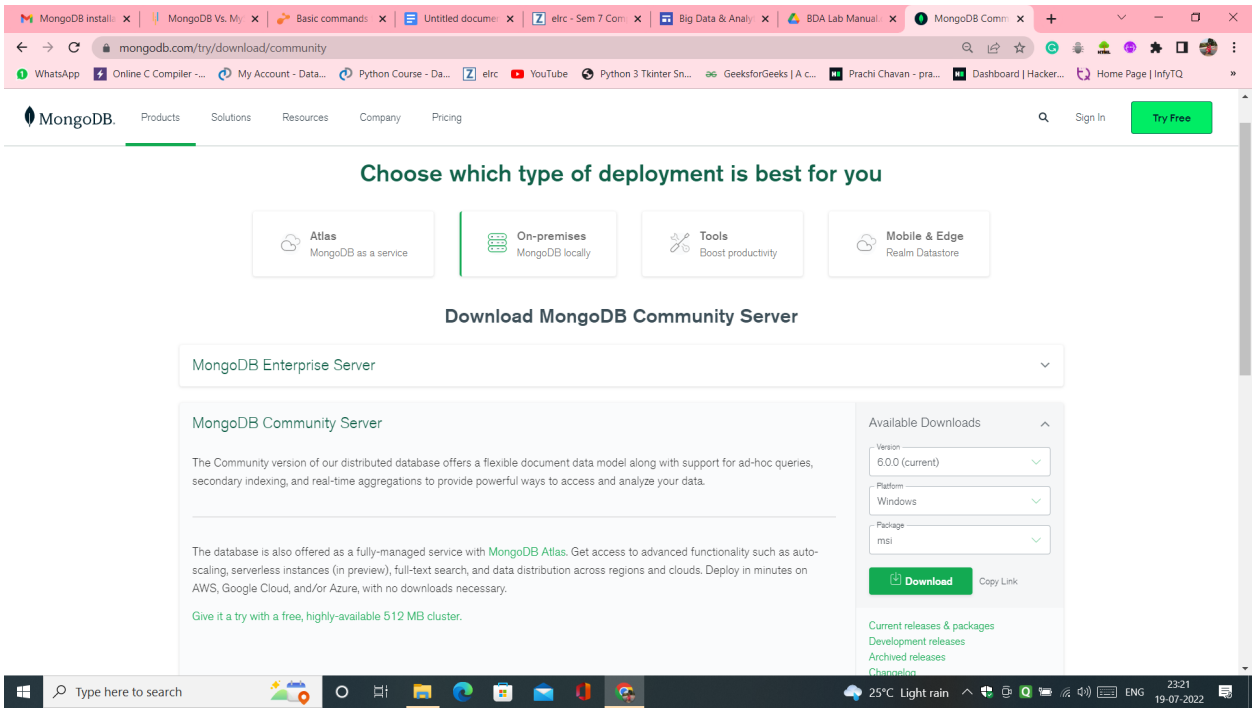
Applications, like an accounting system that requires multi-row transactions, would be better suited for a relational database. MySQL is an excellent choice if you have structured data and need a traditional relational database.	MongoDB is well-suited for real-time analytics, content management, the Internet of Things, mobile, and other types of applications. It is an ideal choice if you have unstructured and/or structured data with rapid growth potential.

Steps to install and configure MongoDB:-

1. Install MongoDB On Windows

To install MongoDB on Windows, first download the latest release of MongoDB from https://www.mongodb.com/try/download/community. Make sure you get the correct version of MongoDB depending on your Windows version.

1. In the **Version** dropdown, select the version of MongoDB to download.
2. In the **Platform** dropdown, select **Windows**.
3. In the **Package** dropdown, select **msi**.
4. Click **Download**.

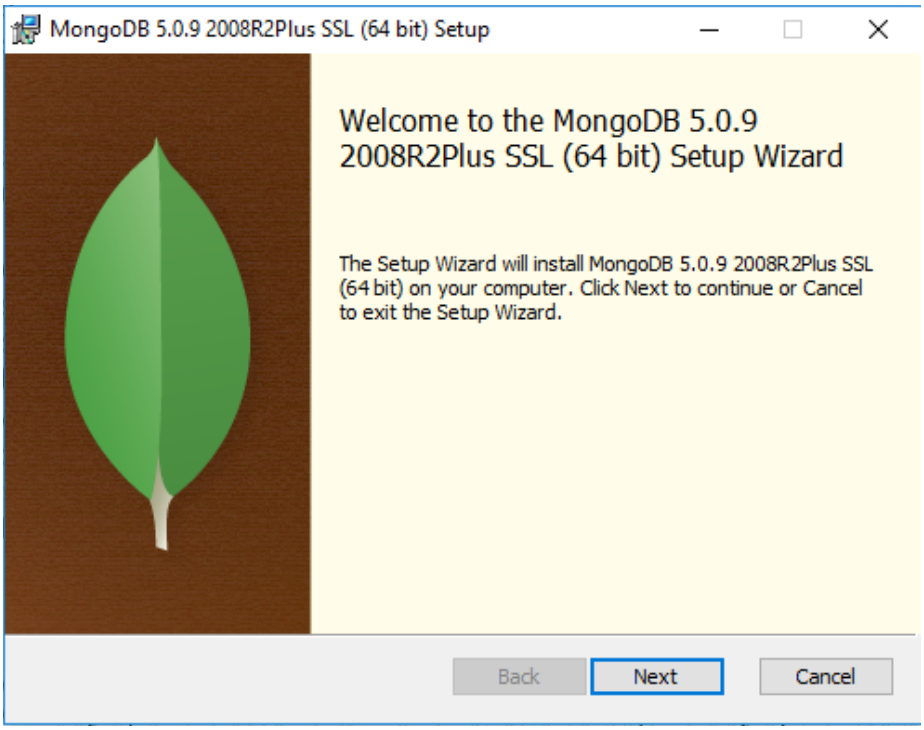


2. Run the MongoDB installer.

For example, from the Windows Explorer/File Explorer:

1. Go to the directory where you downloaded the MongoDB installer (.msi file). By default, this is your Downloads directory.

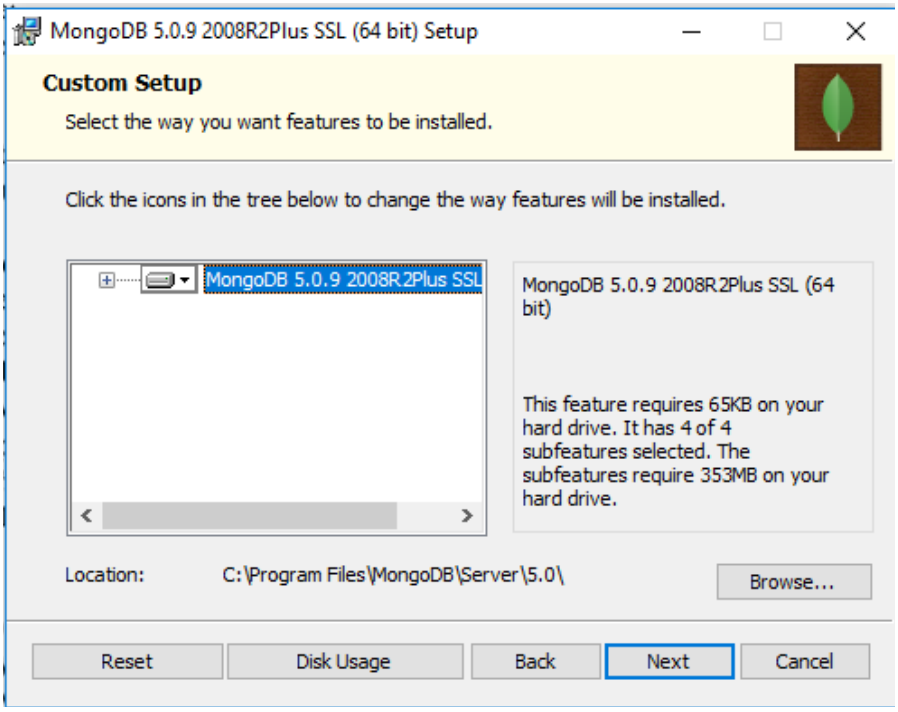
2. Double-click the .msi file.



3. Follow the MongoDB Community Edition installation wizard.

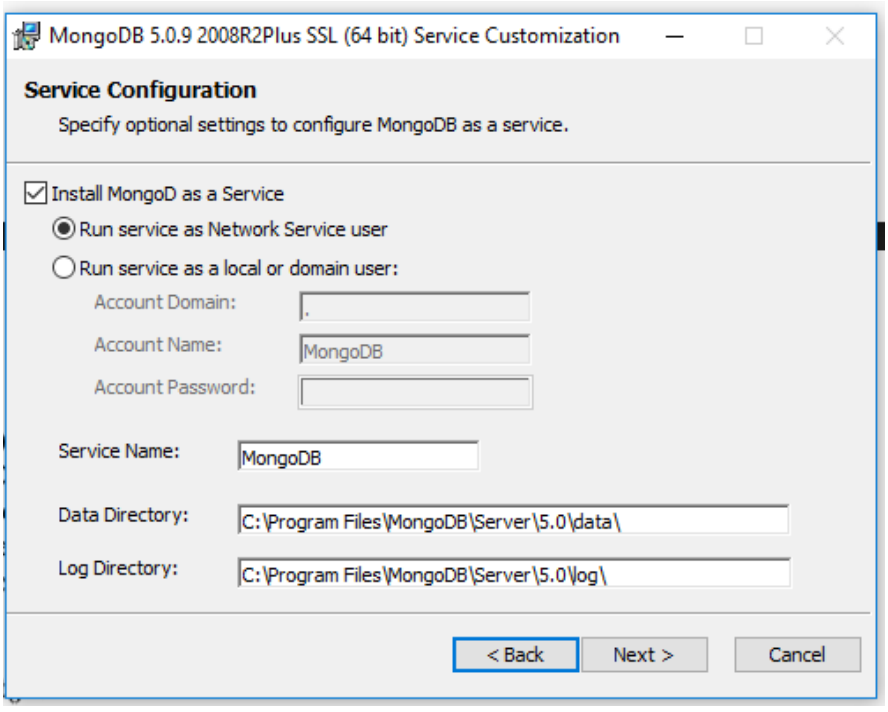
The wizard steps you through the installation of MongoDB and MongoDB Compass.

1. **Choose Setup Type**
   You can choose either the **Complete** (recommended for most users) or **Custom** setup type. The **Complete** setup option installs MongoDB and the MongoDB tools to the default location. The **Custom** setup option allows you to specify which executables are installed and where.



4. Service Configuration

Starting in MongoDB 5.0.9, you can set up MongoDB as a Windows service during the installation or just install the binaries.

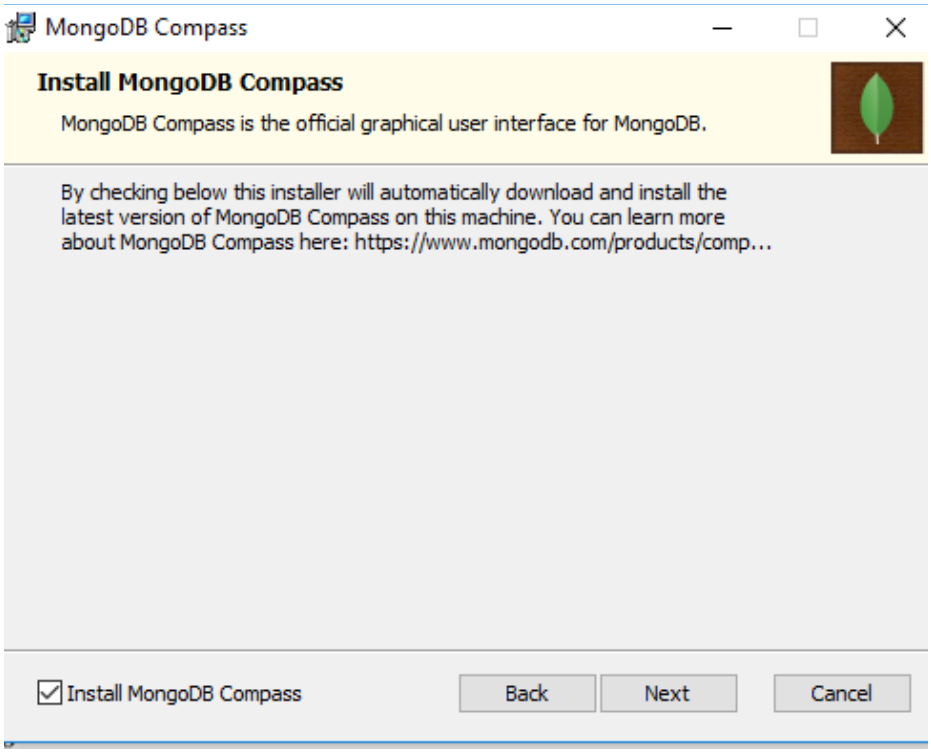


- Select **Install MongoDB as a Service**.
- Select:
  - **Run the service as a Network Service user** (Default)
    This is a Windows user account that is built-in to Windows

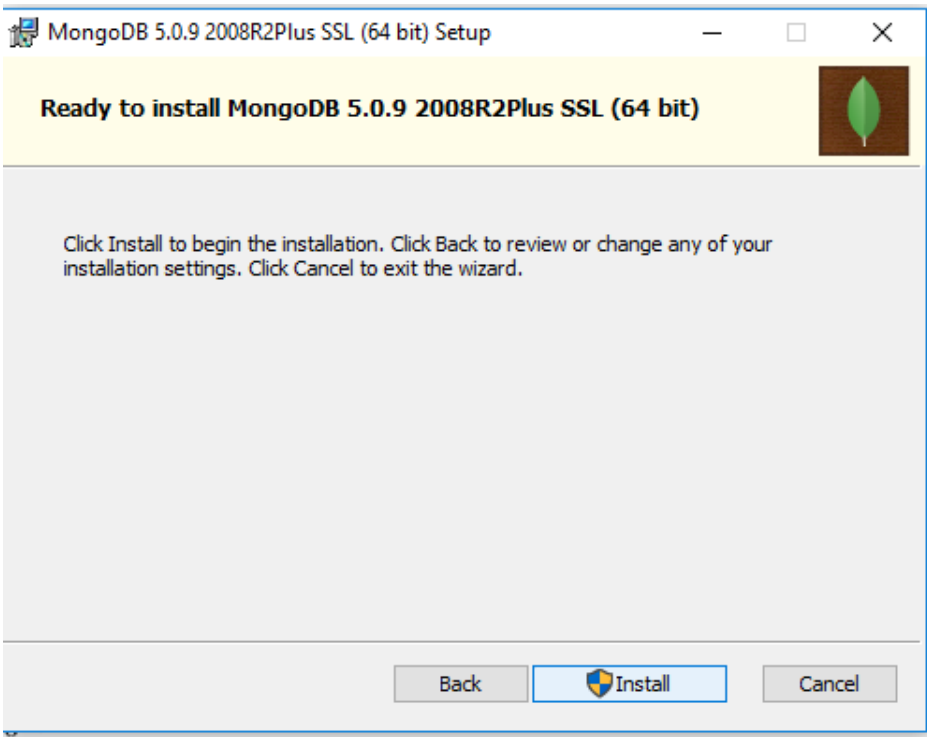
- **Service Name**. Specify the service name. The default name is `MongoDB`. You must choose another name if you already have a service with the specified name.
- **Data Directory**. Specify the data directory, which corresponds to the `--dbpath`. If the directory does not exist, the installer will create the directory and sets the directory access to the service user.
- **Log Directory**. Specify the Log directory, which corresponds to the `--logpath`. If the directory does not exist, the installer will create the directory and sets the directory access to the service user.

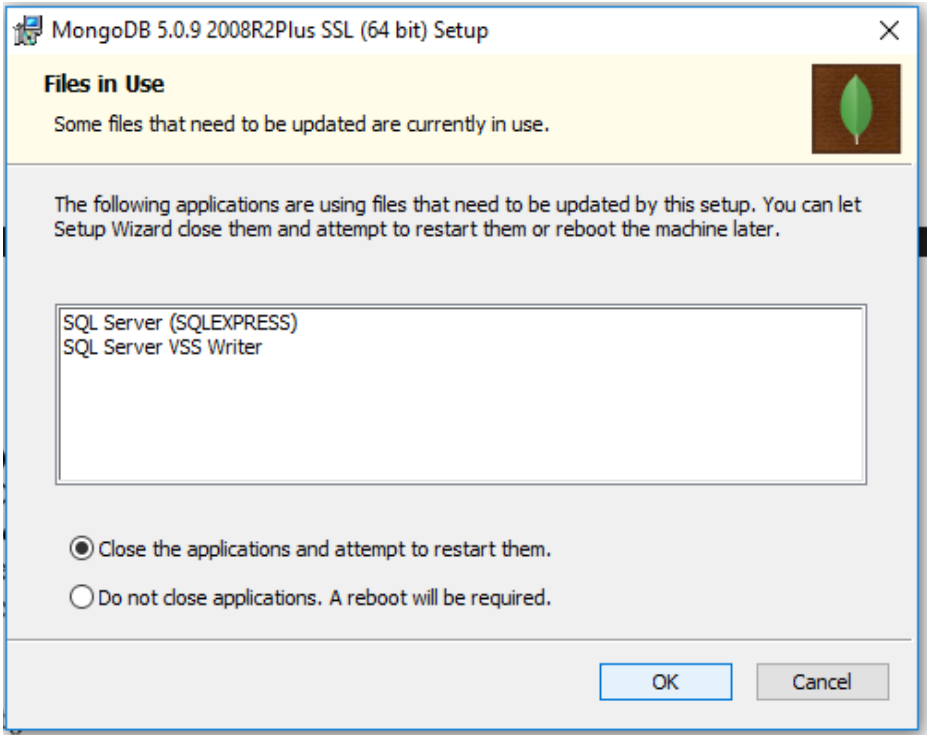
5. Install MongoDB Compass

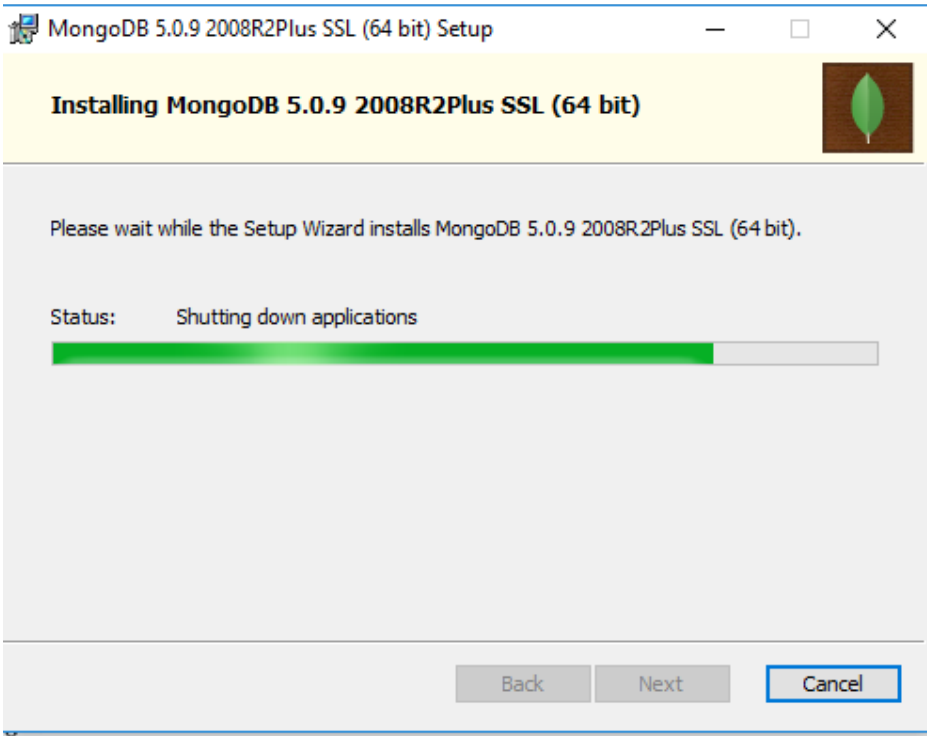
Optional. To have the wizard install MongoDB Compass, select **Install MongoDB Compass** (Default).

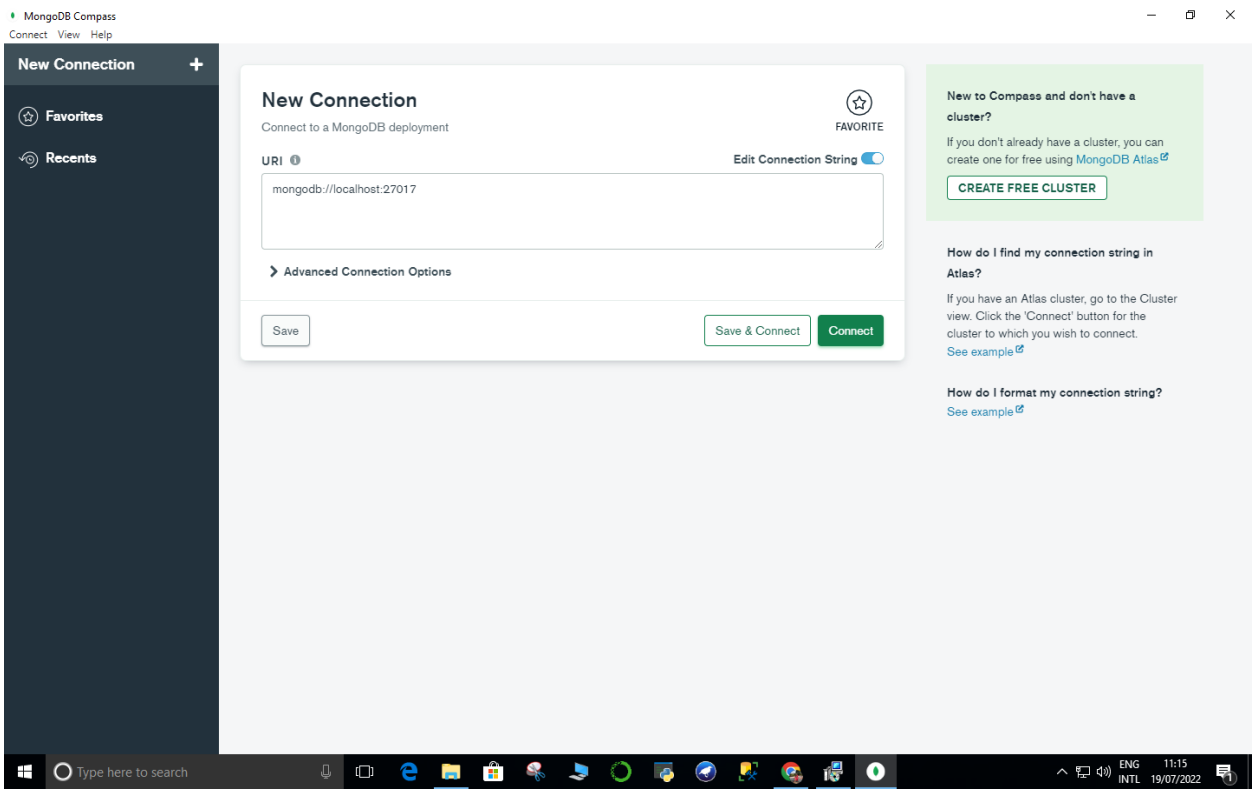


6. When ready, click **Install**.



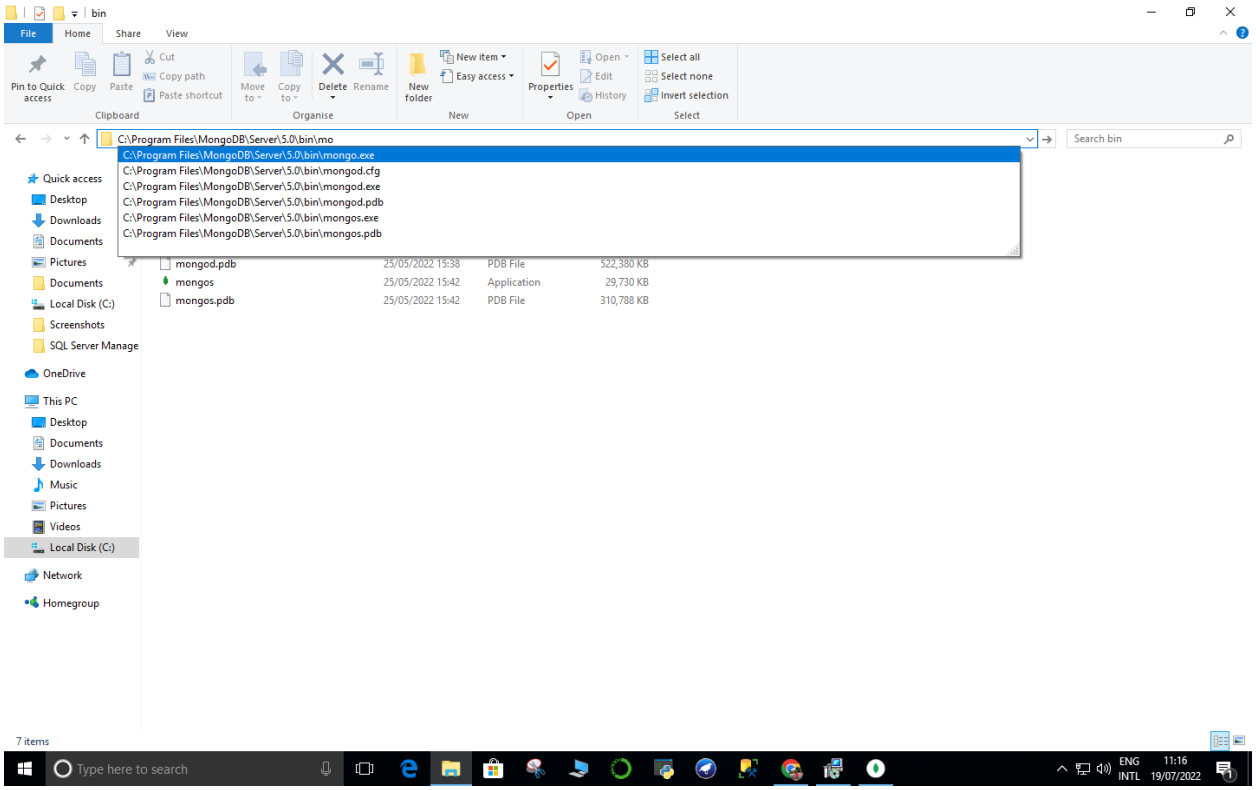



MongoDB 5.0.9 2008R2Plus SSL (64 bit) Setup
Installing MongoDB 5.0.9 2008R2Plus SSL (64 bit)
Please wait while the Setup Wizard installs MongoDB 5.0.9 2008R2Plus SSL (64 bit).
Status: Shutting down applications
Back
Next
Cancel


MongoDB Compass
Connect View Help
New Connection
Favorites
Recents
New Connection
Connect to a MongoDB deployment
FAVORITE
URI
Edit Connection String
mongodb://localhost:27017
Advanced Connection Options
Save
Save & Connect
Connect
New to Compass and don't have a cluster?
If you don't already have a cluster, you can create one for free using MongoDB Atlas
CREATE FREE CLUSTER
How do I find my connection string in Atlas?
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
See example
How do I format my connection string?
See example
Type here to search
ENG
INTL
11:15
19/07/2022

7. Running MongoDB

Go to the location where mongo DB is installed “C:\Program Files\MongoDB\Server\5.0\bin” and open the command prompt at that location and execute the following command in the command prompt.



Commands:

1. **Show All Databases**

Use the below command to get a list of all databases.
show dbs

2. **Create a new database**

To create a new database execute the following command.
use DATABASE_NAME

3. **Know your current selected database**

To know your current working/selected database execute the following command
Db

4. **Create collection**

To create the new collection execute the following commands
db.createCollection(name)

5. **To check collections list**

To get the list of collections created execute the following command
Show collections

6. **Drop collection**

To drop the selected collection execute the following command
>db.COLLECTION_NAME.drop()

7. **Insert document in collection**

>db.COLLECTION_NAME.insert(document)

8. Get collection document

To get the list documents in collection execute the following command
>db.COLLECTION_NAME.find()

9. Update document

To update the document in collection execute the following command
>db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)

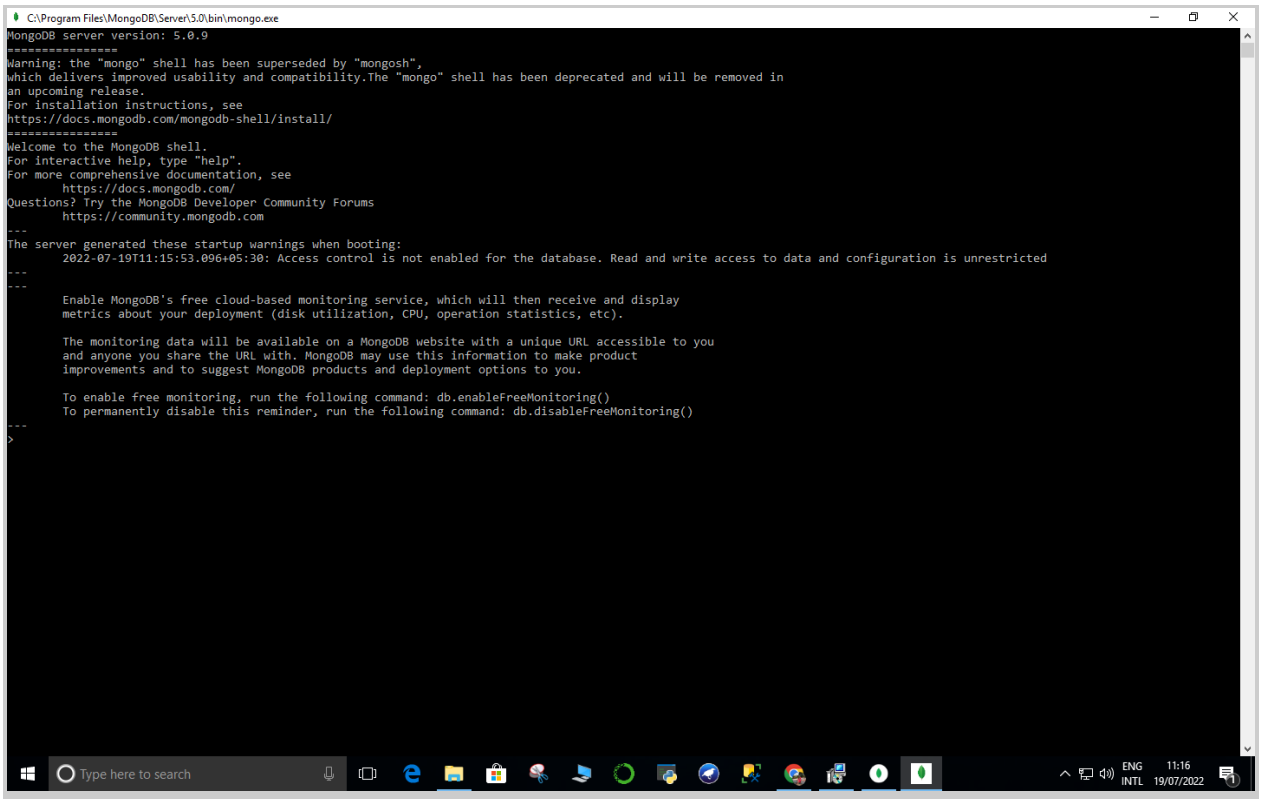
10. Save document

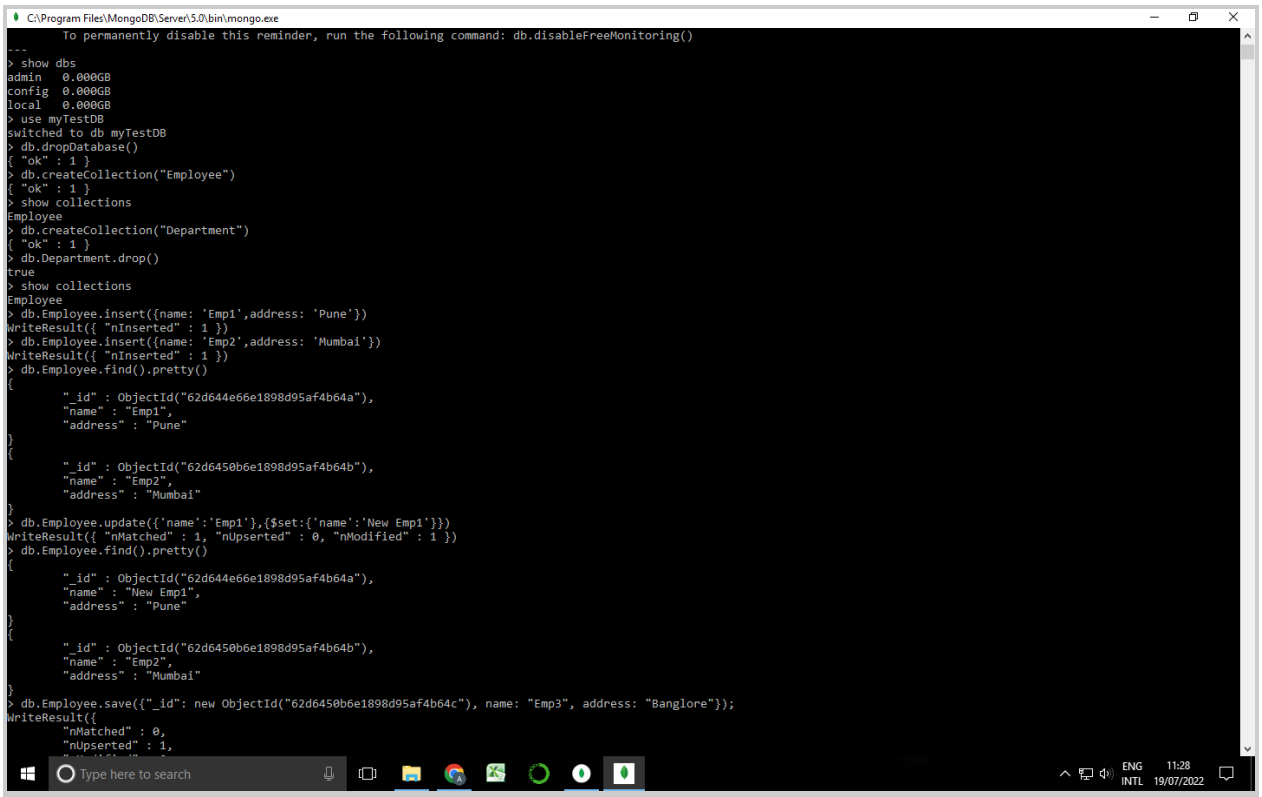
To save document in collection execute the following command
>db.COLLECTION_NAME.save({_id:ObjectId(),NEW_DATA})

11. Delete document

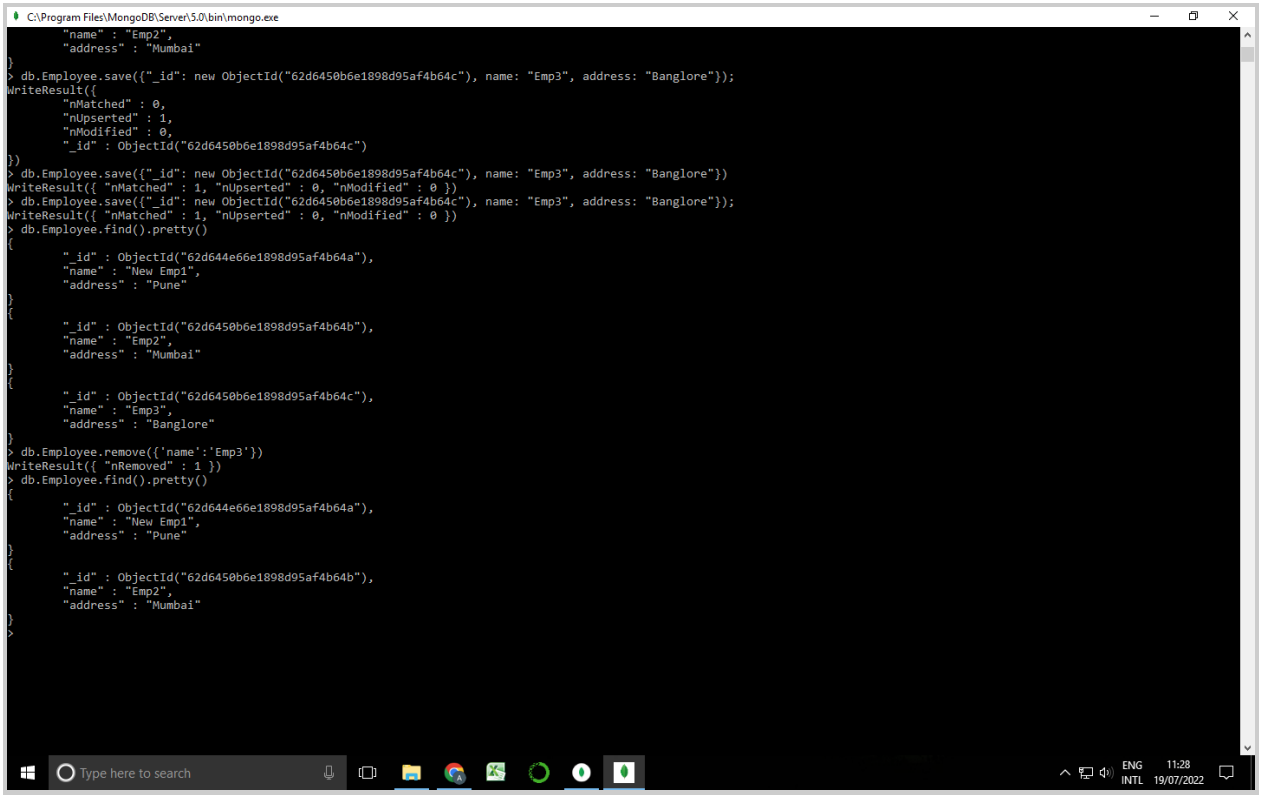
To delete document in selected collection execute the following command
>db.COLLECTION_NAME.remove(DELLETION_CRITTERIA)

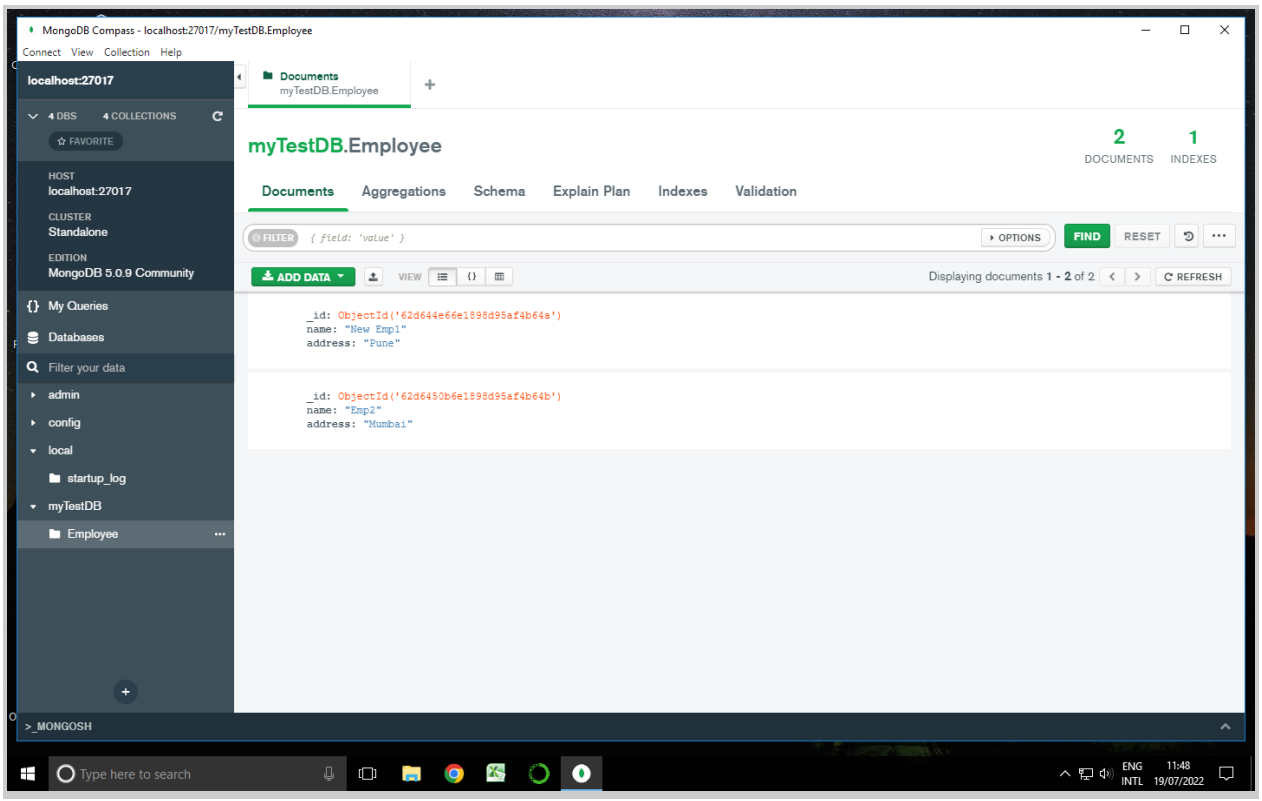
Output:-




C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> use myTestDB
switched to db myTestDB
> db.dropDatabase()
{ "ok" : 1 }
> db.createCollection("Employee")
{ "ok" : 1 }
> show collections
Employee
> db.createCollection("Department")
{ "ok" : 1 }
> db.Department.drop()
true
> show collections
Employee
> db.Employee.insert({name: 'Emp1',address: 'Pune'})
WriteResult({ "nInserted" : 1 })
> db.Employee.insert({name: 'Emp2',address: 'Mumbai'})
WriteResult({ "nInserted" : 1 })
> db.Employee.find().pretty()
{
"_id" : ObjectId("62d644e66e1898d95af4b64a"),
"name" : "Emp1",
"address" : "Pune"
}
{
"_id" : ObjectId("62d6450b6e1898d95af4b64b"),
"name" : "Emp2",
"address" : "Mumbai"
}
> db.Employee.update({'name':'Emp1'},{$set:{'name':'New Emp1'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Employee.find().pretty()
{
"_id" : ObjectId("62d644e66e1898d95af4b64a"),
"name" : "New Emp1",
"address" : "Pune"
}
{
"_id" : ObjectId("62d6450b6e1898d95af4b64b"),
"name" : "Emp2",
"address" : "Mumbai"
}
> db.Employee.save({"_id": new ObjectId("62d6450b6e1898d95af4b64c"), name: "Emp3", address: "Banglore"});
WriteResult({
"nMatched" : 0,
"nUpserted" : 1,
Type here to search
ENG
INTL
11:28
19/07/2022


C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
"name" : "Emp2",
"address" : "Mumbai"
}
> db.Employee.save({"_id": new ObjectId("62d6450b6e1898d95af4b64c"), name: "Emp3", address: "Banglore"});
WriteResult({
"nMatched" : 0,
"nUpserted" : 1,
"nModified" : 0,
"_id" : ObjectId("62d6450b6e1898d95af4b64c")
})
> db.Employee.save({"_id": new ObjectId("62d6450b6e1898d95af4b64c"), name: "Emp3", address: "Banglore"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.Employee.save({"_id": new ObjectId("62d6450b6e1898d95af4b64c"), name: "Emp3", address: "Banglore"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.Employee.find().pretty()
{
"_id" : ObjectId("62d644e66e1898d95af4b64a"),
"name" : "New Emp1",
"address" : "Pune"
}
{
"_id" : ObjectId("62d6450b6e1898d95af4b64b"),
"name" : "Emp2",
"address" : "Mumbai"
}
{
"_id" : ObjectId("62d6450b6e1898d95af4b64c"),
"name" : "Emp3",
"address" : "Banglore"
}
> db.Employee.remove({'name':'Emp3'})
WriteResult({ "nRemoved" : 1 })
> db.Employee.find().pretty()
{
"_id" : ObjectId("62d644e66e1898d95af4b64a"),
"name" : "New Emp1",
"address" : "Pune"
}
{
"_id" : ObjectId("62d6450b6e1898d95af4b64b"),
"name" : "Emp2",
"address" : "Mumbai"
}
>
Type here to search
ENG
INTL
11:28
19/07/2022



Conclusion:-

We installed, configured, and executed NoSQL commands in MongoDB.