

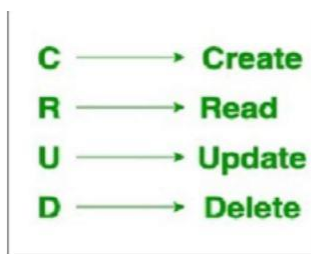
PRACTICAL NO 3

Aim: To implement CRUD operation in mongo db

Software: MongoDB Server, Command line, Mongo shell service

Description:

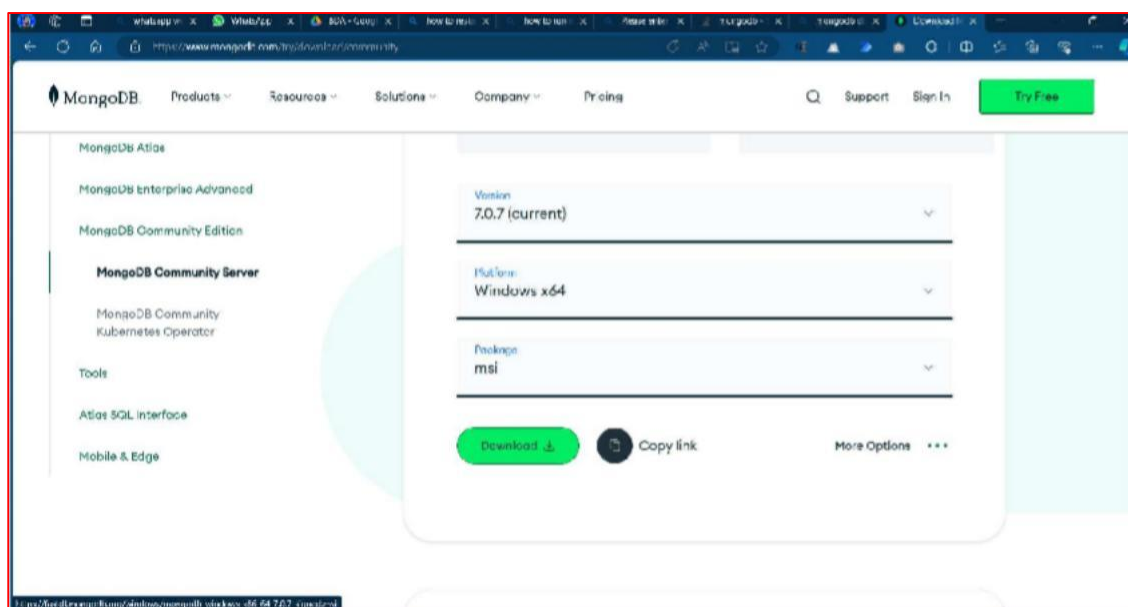
MongoDB for various things like building an application (including web and mobile), or analysis of data, or an administrator of a MongoDB database, in all these cases we need to interact with the MongoDB server to perform certain operations like entering new data into the application, updating data into the application, deleting data from the application, and reading the data of the application. MongoDB provides a set of some basic but most essential operations that will help you to easily interact with the MongoDB server and these operations are known as CRUD operations.



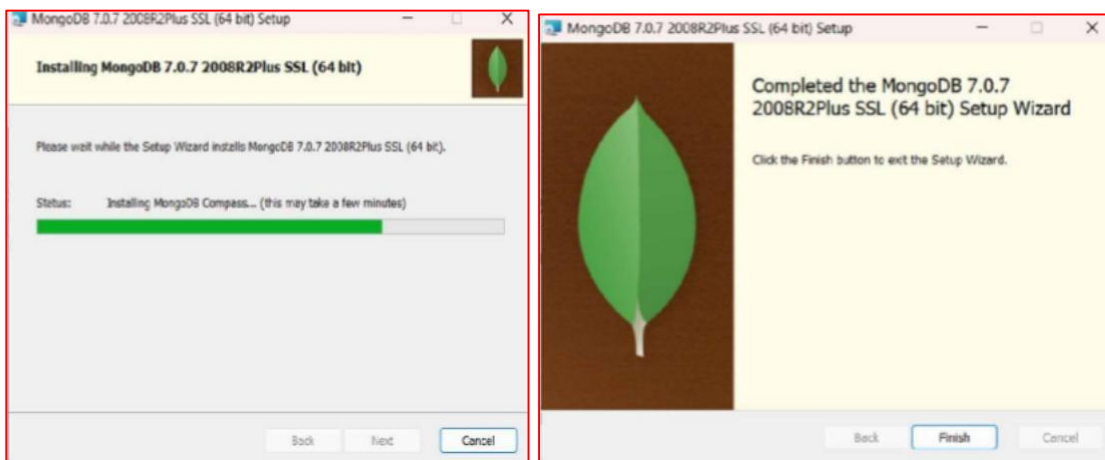
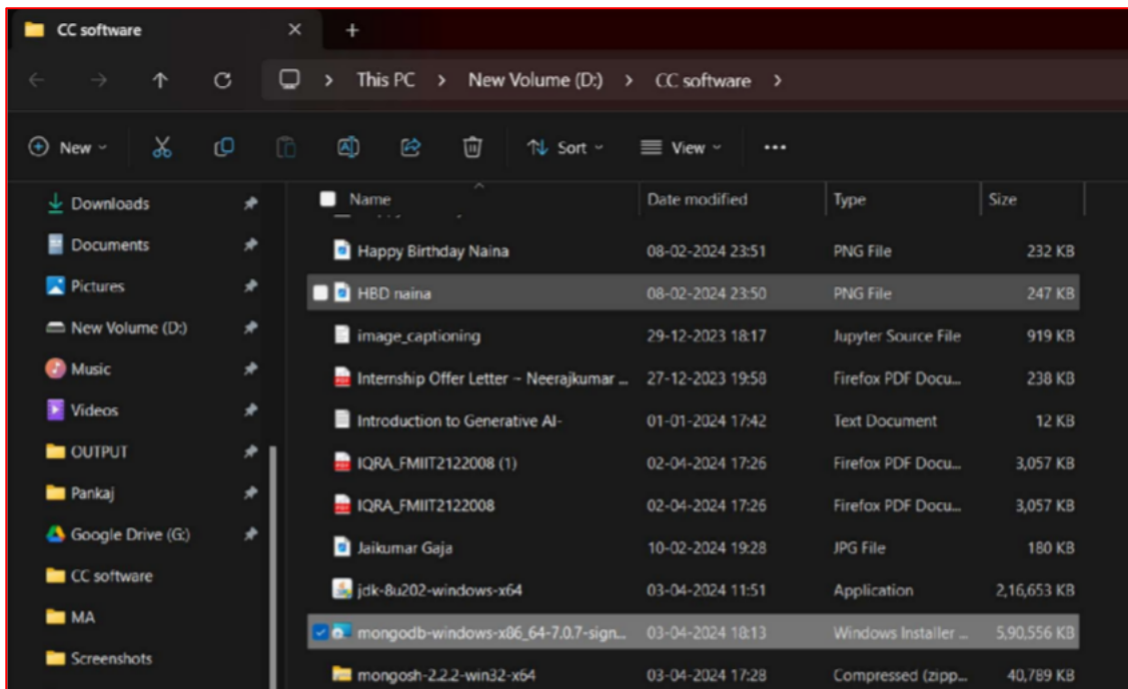
Steps:

Download MongoDB Server from the below given link as it is necessary in order to run the MongoDB Shell.

(Link: <https://www.mongodb.com/try/download/community>)



Once downloaded open the location and click on the downloaded file. Give all the necessary permissions and install the complete version of mongodb server.



After complete installation add the directory to environment variable path by copying the mongodb bin path (C:\Program Files\MongoDB\Server\7.0\bin).

Now open the command prompt and type '**mongod --version**' to check if mongodb server is been setup properly or not.

Note:- Before running mongodb shell always run the '**mongod**' command in the command prompt.

```

C:\Users\acer>mongod --version
db version v7.0.7
Build Info: {
  "version": "7.0.7",
  "gitVersion": "c9b08e1ab7ef741b4abdd0638351b322514c45bd",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}

```

```

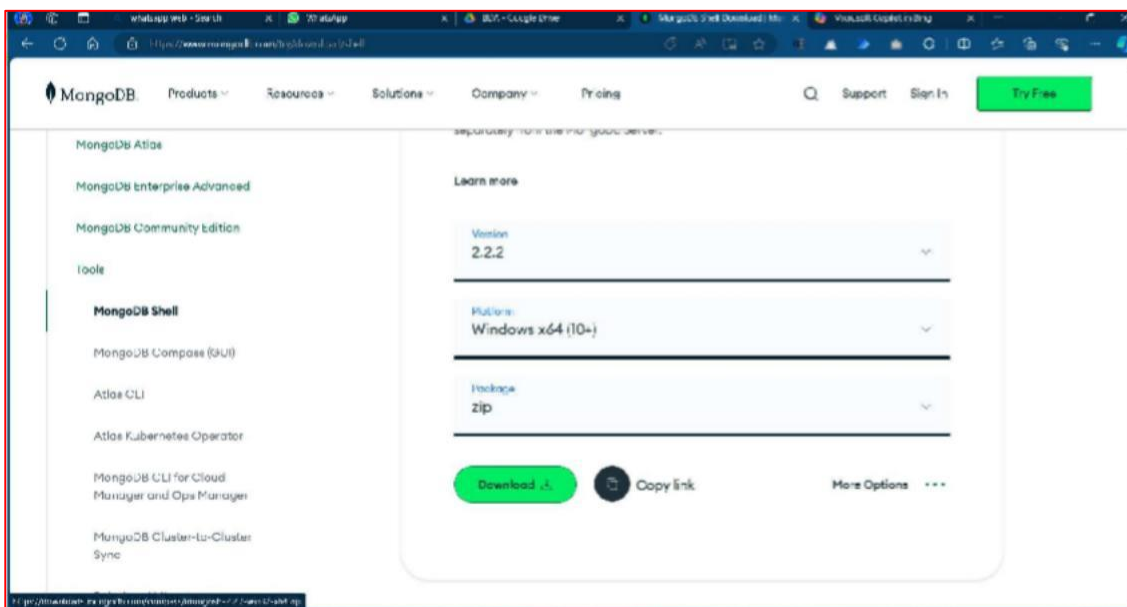
Microsoft Windows [Version 10.0.22621.2270]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>mongod
{"t":{"date":"2024-04-03T18:48:09.908+05:30"},"s":"I",  "c":"NETWORK",  "id":4915901, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient":true}}},"ts":{"date":"2024-04-03T18:48:09.914+05:30"},"s":"I",  "c":"CONTROL",  "id":22285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"},"t":{"date":"2024-04-03T18:48:09.994+05:30"},"s":"I",  "c":"NETWORK",  "id":4943602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."},"ts":{"date":"2024-04-03T18:48:09.997+05:30"},"s":"I",  "c":"REPL",       "id":5123603, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}},"t":{"date":"2024-04-03T18:48:09.997+05:30"},"s":"I",  "c":"REPL",       "id":5123603, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}},"ts":{"date":"2024-04-03T18:48:09.999+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi-threading initialized"},"t":{"date":"2024-04-03T18:48:09.999+05:30"},"s":"I",  "c":"TENANT_M",  "id":7901609, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}

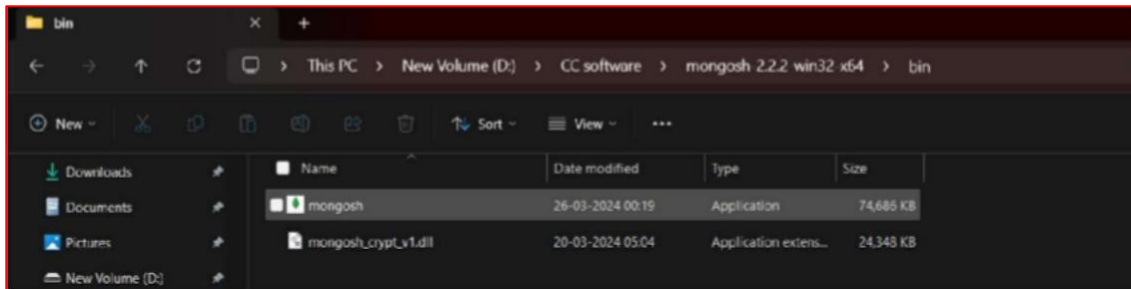
```

1) Download MongoDB Shell

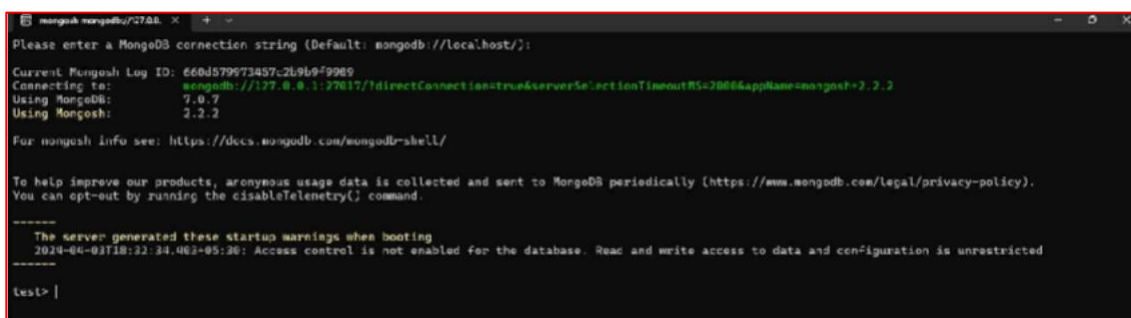
(Link: <https://www.mongodb.com/try/download/shell>)



2) Extract it where it is been downloaded and install the application. Click and open the shell application



MongoDB Shell.....



Note:- Sometime it will ask you to create database to store the data in that situation you need to create a data folder in the C: drive and inside data folder create a db folder.

4) Create Operations –

The create or insert operations is used to insert or add new documents in the collection. If a collection does not exist, then it will create a new collection in the database. You can perform, create operations using the following methods provided by the MongoDB:

It is used to insert a single document in the collection.

Code:-

use BDA

db.collection.insertOne()

Example 1: In this example, we are inserting details of a single student in the form of document in the student collection using db.collection.insertOne() method.

Output:

```
test> use BDA
switched to db BDA
BDA> db.student.insertOne({
... name : "Neeraj",
... age : 22,
... branch : "IT",
... course : "MSC",
... mode : "offline",
... paid : true,
... amount : 52000
... })
{
  acknowledged: true,
  insertedId: ObjectId('660d5cc431728aa2899f990a')
}
BDA>
```

5) Insert Many -

It is used to insert multiple documents in the collection

Code:

```
db.collection.insertMany().
```

Example 2:

In this example, we are inserting details of the multiple students in the form of documents in the student collection using db.collection.insertMany() method.

```
BDA> db.student.insertMany([
... {
... name : "Gautam",
... age : 21,
... branch : "IT",
... course : "HSC",
... mode : "offline",
... paid : true,
... amount : 52000
... },
... {
... name : "Kiran",
... age : 22,
... branch : "IT",
... course : "HSC",
... mode : "offline",
... paid : true,
... amount : 52000
... }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660d5de931728aa2899f990b'),
    '1': ObjectId('660d5de931728aa2899f990c')
  }
}
BDA> |
```

6) Read Operations –

The Read operations are used to retrieve documents from the collection, or in other words, read operations are used to query a collection for a document. You can perform read operation using the following method provided by the MongoDB:

It is used to retrieve documents from the collection.

Code:-

```
db.collection.find().pretty()
```

Example :

In this example, we are retrieving the details of students from the student collection using db.collection.find() method.

```
BDA> db.student.find().pretty()
[
  {
    _id: ObjectId('660d5cc431728aa2899f990a'),
    name: 'Neeraj',
    age: 22,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  },
  {
    _id: ObjectId('660d5de931728aa2899f990b'),
    name: 'Gautam',
    age: 21,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  },
  {
    _id: ObjectId('660d5de931728aa2899f990c'),
    name: 'Kiran',
    age: 22,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  }
]
BDA> |
```

7) Update Operations –

The update operations are used to update or modify the existing document in the collection. You can perform update operations using the following methods provided by the MongoDB:

It is used to update a single document in the collection that satisfy the given criteria.

Code:

```
db.collection.updateOne()
```

Example 1:

In this example, we are updating the age of Sumit in the student collection using `db.collection.updateOne()` method.

```
BDA> db.student.updateOne({name: "Neeraj"},{$set:{age:30}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
BDA> db.student.find().pretty()
[
  {
    _id: ObjectId('660d5cc431728aa2899f990a'),
    name: 'Neeraj',
    age: 30,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  },
  {
    _id: ObjectId('660d5de931728aa2899f990b'),
    name: 'Gautam',
    age: 21,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  },
  {
    _id: ObjectId('660d5de931728aa2899f990c'),
    name: 'Kiran',
    age: 22,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000
  }
]
```


It is used to update multiple documents in the collection that satisfy the given criteria.

Code:

```
db.collection.updateMany()
```

Example 2:

In this example, we are updating the year of course in all the documents in the student collection using db.collection.updateMany() method.

```
BDA> db.student.updateMany({},{$set:{year:2024}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
BDA> db.student.find().pretty()
[
  {
    _id: ObjectId('660d5cc431728aa2899f990a'),
    name: 'Neeraj',
    age: 30,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000,
    year: 2024
  },
  {
    _id: ObjectId('660d5de931728aa2899f990b'),
    name: 'Gautam',
    age: 21,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000,
    year: 2024
  },
  {
    _id: ObjectId('660d5de931728aa2899f990c'),
    name: 'Kiran',
    age: 22,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000,
    year: 2024
  }
]
BDA> |
```


8) Delete Operations –

The delete operation are used to delete or remove the documents from a collection. You can perform delete operations using the following methods provided by the MongoDB:

It is used to delete a single document from the collection that satisfy the given criteria.

Code:

```
db.collection.deleteOne()
```

Example 1:

In this example, we are deleting a document from the student collection using db.collection.deleteOne() method.

```
BDA> db.student.deleteOne({name : "Neeraj"})
{ acknowledged: true, deletedCount: 1 }
BDA> db.student.find().pretty()
[
  {
    _id: ObjectId('660d5de931728aa2899f990b'),
    name: 'Gautam',
    age: 21,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000,
    year: 2024
  },
  {
    _id: ObjectId('660d5de931728aa2899f990c'),
    name: 'Kiran',
    age: 22,
    branch: 'IT',
    course: 'MSC',
    mode: 'offline',
    paid: true,
    amount: 52000,
    year: 2024
  }
]
BDA> |
```

It is used to delete multiple documents from the collection that satisfy the given criteria.

Code:

```
db.collection.deleteMany()
```

Example 2:

In this example, we are deleting all the documents from the student collection using `db.collection.deleteMany()` method.

```
BDA> db.student.deleteMany({})  
{ acknowledged: true, deletedCount: 2 }  
BDA> db.student.find().pretty()  
  
BDA> |
```