MONGODB ASSIGNMENT

Screenshot 1

```
madhurgupta@Madhurs-MacBook-Air ~ % brew tap mongodb/brew
==> Tapping mongodb/brew
Cloning into '/opt/homebrew/Library/Taps/mongodb/homebrew-brew'...
remote: Enumerating objects: 1256, done.
remote: Counting objects: 100% (541/541), done.
remote: Compressing objects: 100% (185/185), done.
remote: Total 1256 (delta 412), reused 441 (delta 355), pack-reused 715
Receiving objects: 100% (1256/1256), 273.16 KiB | 4.48 MiB/s, done.
Resolving deltas: 100% (707/707), done.
Tapped 17 formulae (36 files, 358.6KB).
madhurgupta@Madhurs-MacBook-Air ~ % brew update
Installing from the API is now the default behaviour!
You can save space and time by running:
  brew untap homebrew/core
Already up-to-date.
madhurgupta@Madhurs-MacBook-Air ~ % brew install mongodb-community@7.0
==> Fetching dependencies for mongodb/brew/mongodb-community: mongodb/brew/mongodb-database-tools, c-ares, libuv, node and mongosh
==> Fetching mongodb/brew/mongodb-database-tools
==> Downloading https://fastdl.mongodb.org/tools/db/mongodb-database-tools-macos-arm64-100.8.0.zip
######################################################################################################################## 100.0%
==> Fetching c-ares
==> Downloading https://ghcr.io/v2/homebrew/core/c-ares/manifests/1.19.1
######################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/c-ares/blobs/sha256:ba35cc0962beaea7ae345ee1818297c40d5653649e563dc9493b93924b87ae41
######################################################################################################################## 100.0%
==> Fetching libuv
==> Downloading https://ghcr.io/v2/homebrew/core/libuv/manifests/1.46.0
######################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libuv/blobs/sha256:8c3beb4d11ed0d45cf0b7e07d280ff815eab9f9c138eec90a2f824168aed039e
######################################################################################################################## 100.0%
==> Fetching node
==> Downloading https://ghcr.io/v2/homebrew/core/node/manifests/20.7.0
######################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/node/blobs/sha256:a278eb1ace9cc5b780a96fc3a3dc5bd7625ec1af61962b4842c702b2260c9aba
######################################################################################################################## 100.0%
==> Fetching mongosh
```

Screenshot 2

```
~ — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000 — -zsh            +
madhurgupta@Madhurs-MacBook-Air ~ % brew --prefix
/opt/homebrew
madhurgupta@Madhurs-MacBook-Air ~ % brew services start mongodb-community@7.0
==> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-community)
madhurgupta@Madhurs-MacBook-Air ~ % mongosh
Current Mongosh Log ID: 650c3591451d6b1b4174c5be
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.0
Using Mongosh:          2.0.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2023-09-21T17:46:36.869+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> db.help

  Database Class:

    getMongo                          Returns the current database connection
    getName                           Returns the name of the DB
    getCollectionNames                Returns an array containing the names of all collections in the current database.
    getCollectionInfos                Returns an array of documents with collection information, i.e. collection name and options, for the current database.
    runCommand                        Runs an arbitrary command on the database.
    adminCommand                      Runs an arbitrary command against the admin database.
    aggregate                         Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
    getSiblingDB                      Returns another database without modifying the db variable in the shell environment.
    getCollection                     Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
    dropDatabase                      Removes the current database, deleting the associated data files.
    createUser                        Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.
    updateUser                        Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.
    changeUserPassword                Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.
    logout                            Ends the current authentication session. This function has no effect if the current session is not authenticated.
    dropUser                          Removes the user from the current database.
```

Screenshot 3

```
test> show dbs;
admin    40.00 KiB
config   60.00 KiB
local    40.00 KiB
test> use CDAC
switched to db CDAC
CDAC> db
CDAC
CDAC> show collections;

CDAC> db.createCollection("Library");
{ ok: 1 }
```

Screenshot 4

```
CDAC> show collections
Library
CDAC> db.library.insert({title:'MongoDB Programming',author:'Madhur',likes:100})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("650c3ac9451d6b1b4174c5bf") }
}
CDAC> db.library.find()
[
  {
    _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
    title: 'MongoDB Programming',
    author: 'Madhur',
    likes: 100
  }
]
CDAC> db.library.insert({title:'MongoDB programming',authors:['Jack','Jill'],likes:200})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("650c3ba1451d6b1b4174c5c0") }
}
CDAC> db.library.find()
[
  {
    _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
    title: 'MongoDB Programming',
    author: 'Madhur',
    likes: 100
  },
  {
    _id: ObjectId("650c3ba1451d6b1b4174c5c0"),
    title: 'MongoDB programming',
    authors: [ 'Jack', 'Jill' ],
    likes: 200
  }
]
```

Screenshot 5

```
CDAC> db.library.findOne()
{
  _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
  title: 'MongoDB Programming',
  author: 'Madhur',
  likes: 100
}
CDAC> db.library.find().pretty();
[
  {
    _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
    title: 'MongoDB Programming',
    author: 'Madhur',
    likes: 100
  },
  {
    _id: ObjectId("650c3ba1451d6b1b4174c5c0"),
    title: 'MongoDB programming',
    authors: [ 'Jack', 'Jill' ],
    likes: 200
  }
]
```

Screenshot 6

```
CDAC> db.library.update({author:'Madhur'},{$set:{author:'Madhur Gupta'}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
CDAC> db.library.find()
[
  {
    _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
    title: 'MongoDB Programming',
    author: 'Madhur Gupta',
    likes: 100
  },
  {
    _id: ObjectId("650c3ba1451d6b1b4174c5c0"),
    title: 'MongoDB programming',
    authors: [ 'Jack', 'Jill' ],
    likes: 200
  }
]
CDAC> db.library.remove({likes:118})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 0 }
CDAC> db.library.find()
[
  {
    _id: ObjectId("650c3ac9451d6b1b4174c5bf"),
    title: 'MongoDB Programming',
    author: 'Madhur Gupta',
    likes: 100
  },
  {
    _id: ObjectId("650c3ba1451d6b1b4174c5c0"),
    title: 'MongoDB programming',
    authors: [ 'Jack', 'Jill' ],
    likes: 200
  }
]
CDAC> db.library.remove({likes:100})
{ acknowledged: true, deletedCount: 1 }
```

Screenshot 7

```
[CDAC> db.library.remove({likes:100})
{ acknowledged: true, deletedCount: 1 }
[CDAC> db.library.find()
[
  {
    _id: ObjectId("650c3ba1451d6b1b4174c5c0"),
    title: 'MongoDB programming',
    authors: [ 'Jack', 'Jill' ],
    likes: 200
  }
]
[CDAC> db.library.drop()
true
[CDAC> show collections
Library
[CDAC> db.Library.drop()
true
[CDAC> show collections

[CDAC> db.dropDatabase('CDAC')
{ ok: 1, dropped: 'CDAC' }
[CDAC> show dbs
admin    40.00 KiB
config  108.00 KiB
local    40.00 KiB
[CDAC> exit
[madhurgupta@Madhurs-MacBook-Air ~ % brew services stop mongodb-community@7.0
```