Step 1: Install MongoDB

If you haven't installed MongoDB yet, you can download it from the MongoDB official website and follow the installation instructions for your operating system.

Step 2: Start MongoDB

If you've installed MongoDB locally, start the MongoDB server by running:

This starts the MongoDB server on the default port (27017).

Step 3: Connect to MongoDB

Open a new terminal window and connect to your MongoDB server using:

bash

Copy code

mongo

This connects you to the MongoDB shell where you can execute commands.

Step 4: Create a Database

To create a new database, use the use command:

use myDatabase

Here, myDatabase is the name of the database you want to create. If the database does not already exist, MongoDB will create it when you first store data in it.

Step 5: Create a Collection

A collection is like a table in relational databases. Create a collection within your database:

db.createCollection("myCollection")

Here, myCollection is the name of the collection you want to create.

Step 6: Perform CRUD Operations

Create (Insert Data)

To insert a document (record) into your collection:

db.myCollection.insertOne({ name: "Alice", age: 25, city: "New York" })

To insert multiple documents:

```
db.myCollection.insertMany([
  { name: "Bob", age: 30, city: "San Francisco" },
  { name: "Charlie", age: 35, city: "Los Angeles" }
])
```

Read (Query Data)

To find a single document:

db.myCollection.findOne({ name: "Alice" })

To find all documents:

db.myCollection.find()

You can also use pretty formatting to make the output more readable:

db.myCollection.find().pretty()

Update (Modify Data)

To update a single document:

```
db.myCollection.updateOne(
  { name: "Alice" },
  { $set: { age: 26 } }
)
```

To update multiple documents:

```
db.myCollection.updateMany(
  { city: "New York" },
  { $set: { city: "Boston" } }
)
```

Delete (Remove Data)

To delete a single document:

db.myCollection.deleteOne({ name: "Alice" })

To delete multiple documents:

db.myCollection.deleteMany({ city: "Boston" })

Step 7: Drop a Collection or Database

To drop (delete) a collection:

db.myCollection.drop()

To drop (delete) a database:

javascript

Copy code

use myDatabase

db.dropDatabase()

Example Summary

Here's a quick summary of the commands:

1. **Create Database:**

use myDatabase

2. **Create Collection:**

db.createCollection("myCollection")

3. **Insert Data:**

db.myCollection.insertOne({ name: "Alice", age: 25, city: "New York" })

db.myCollection.insertMany([{ name: "Bob", age: 30, city: "San Francisco" }, { name: "Charlie", age: 35, city: "Los Angeles" }])

4. **Find Data:**

db.myCollection.findOne({ name: "Alice" })

db.myCollection.find().pretty()

5. **Update Data:**

db.myCollection.updateOne({ name: "Alice" }, { $set: { age: 26 } })

db.myCollection.updateMany({ city: "New York" }, { $set: { city: "Boston" } })

6. **Delete Data:**

db.myCollection.deleteOne({ name: "Alice" })

db.myCollection.deleteMany({ city: "Boston" })

7. **Drop Collection or Database:**

db.myCollection.drop()

use myDatabase

db.dropDatabase()