**Advance Database Management System**

**Practical 3: CRUD Operations using MongoDB**

**I. Setup & Installation**

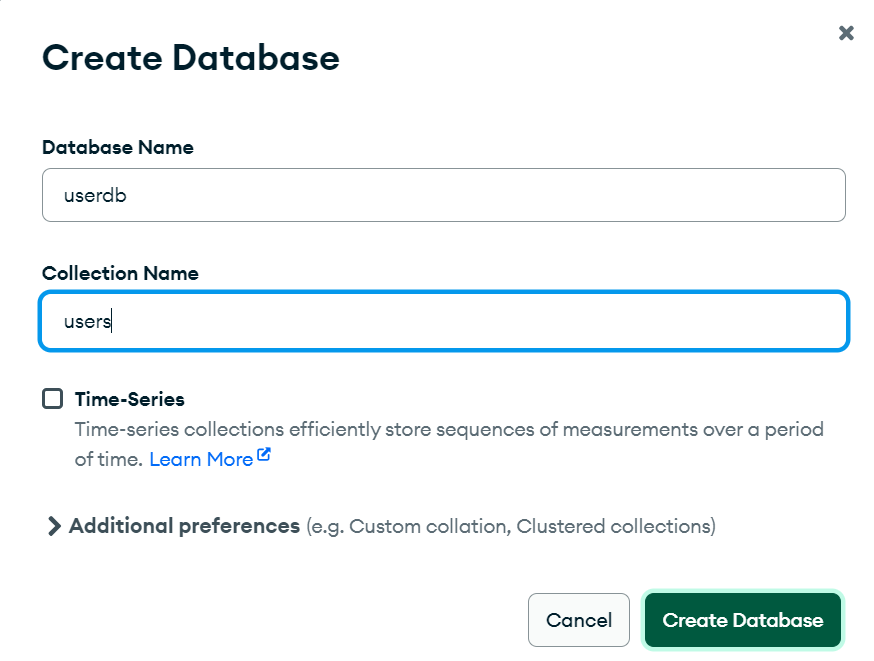
1. **Install MongoDB Compass:**
   * Download from: <https://www.mongodb.com/try/download/compass>
   * Install the downloaded package. MongoDB Compass provides a GUI to interact with your MongoDB database.
2. **Set Environment Variable (Optional but Recommended):**
   * This makes it easier to access MongoDB commands from your command line.
   * Find the bin directory within your MongoDB installation (e.g., C:\Program Files\MongoDB\Server\8.0\bin).
   * Add this path to your system's PATH environment variable.
3. **Install MongoDB Shell (mongosh):**
   * Download from: <https://www.mongodb.com/try/download/shell>
   * Install the downloaded package. This provides a command-line interface to MongoDB.

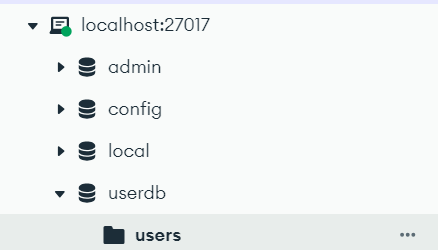
**II. Accessing MongoDB**

1. **Open MongoDB Compass:** Launch the MongoDB Compass application.
2. **Connect to your MongoDB instance:**
   * Typically, you can connect to a local MongoDB instance using the default connection string: mongodb://localhost:27017. Enter this in the connection field and click "Connect."

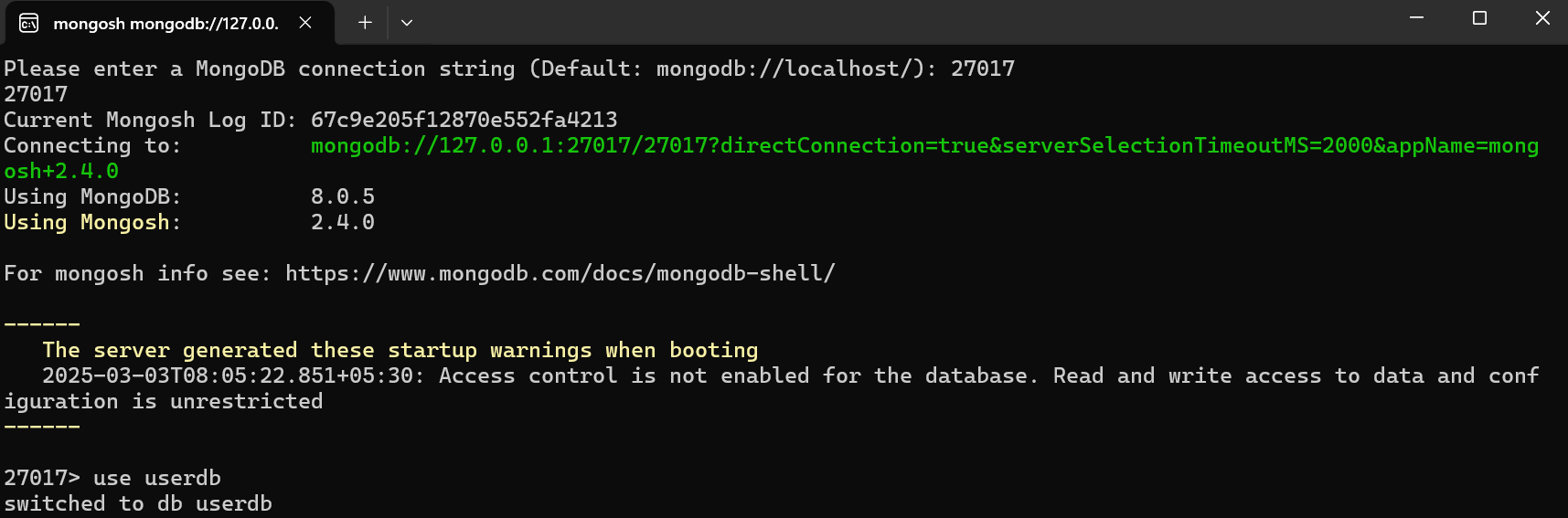
**III. CRUD Operations**

1. **Creating a Database:**
   * In MongoDB Compass, click on "Create Database".
   * Enter the database name (e.g., userdb) and a collection name (e.g., users). Click "Create Database".





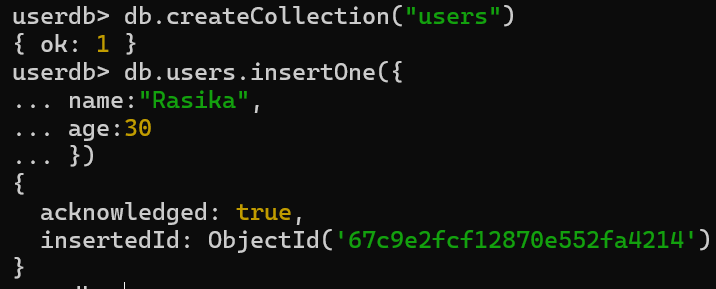
* + Alternatively, you can use the mongosh shell. Open your command line or terminal and type mongosh. Then switch to the “userdb” database.



1. **Creating a Collection (If you didn't do it when creating the database):**

* In MongoDB Compass, a collection is created when you insert the first document.

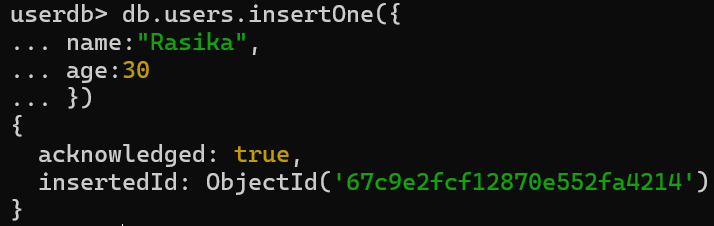
db.createCollection("users")



1. **Create (Insert) Operations:**

* **insertOne():** Adds a single document.

db.users.insertOne({ name: "Rasika", age: 21 })



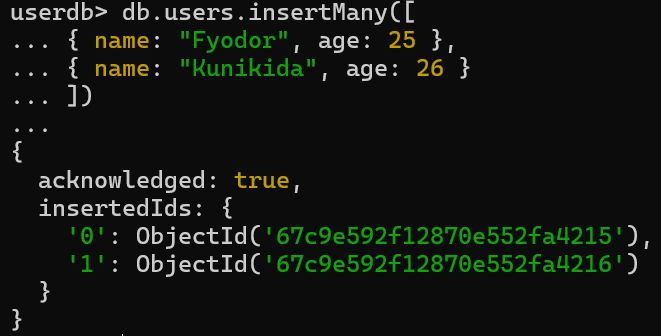
1. **insertMany():** Adds multiple documents at once.

db.users.insertMany([

{ name: "Fyodor", age: 25 },

{ name: "Kunikida", age: 26 }

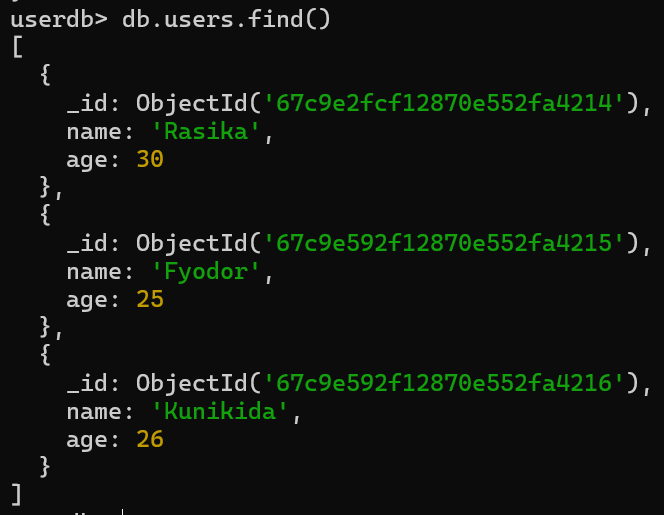
])



1. **Read (Find) Operations:**

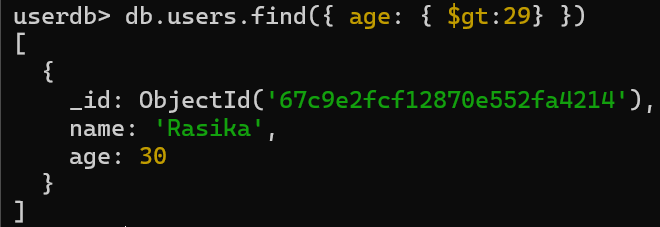
* **find():** Retrieves documents.
  + To retrieve all documents in the users collection:

db.users.find()



* + To retrieve documents with a specific condition (e.g., age greater than 29):

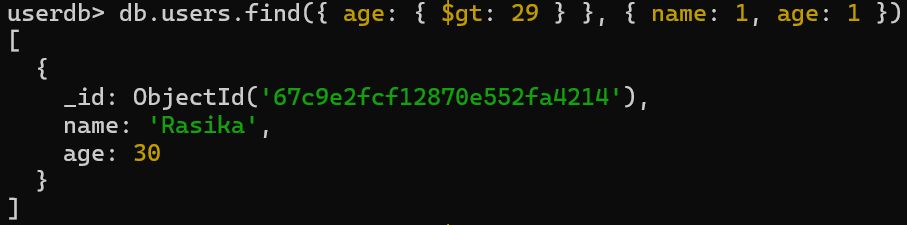
db.users.find({ age: { $gt: 29 } })



* + To retrieve documents with specific fields (name and age):

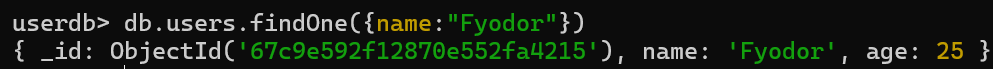
db.users.find({ age: { $gt: 29 } }, { name: 1, age: 1 })

*( 1 includes the field, 0 excludes it. \_id is included by default)*



* + **findOne():** Retrieves a single document.

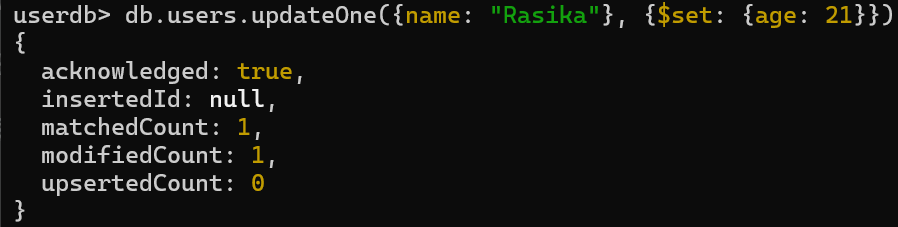
db.users.findOne({ name: "Fyodor" })

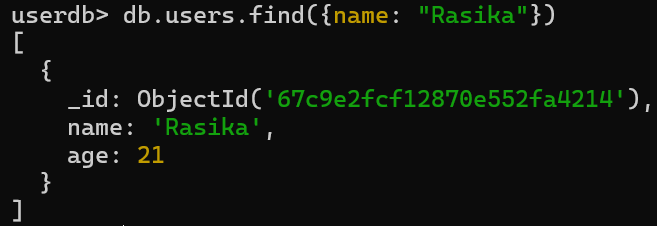


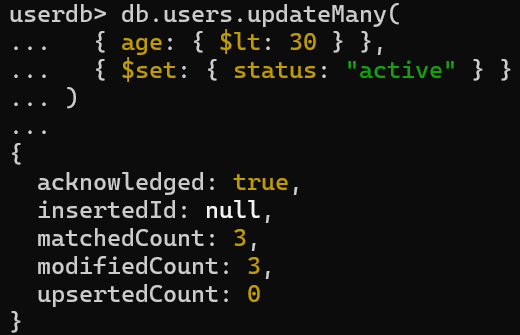
1. **Update Operations:**

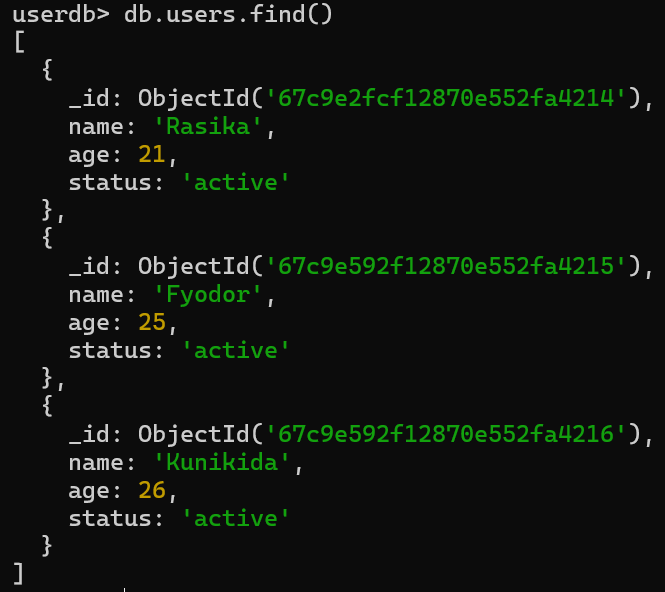
* **updateOne():** Updates a single document.

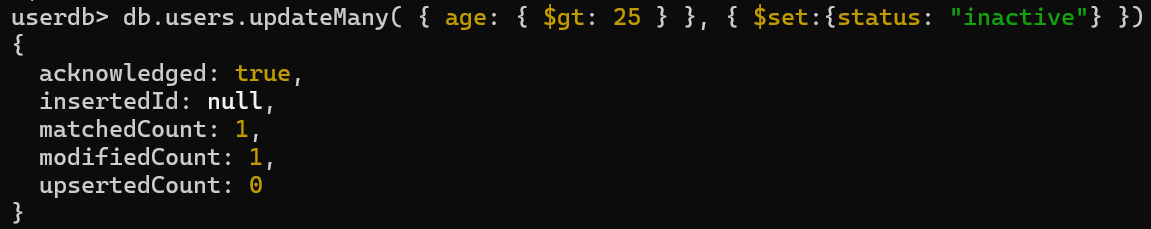
db.users.updateOne({name: "Rasika"}, {$set: {age: 21}})

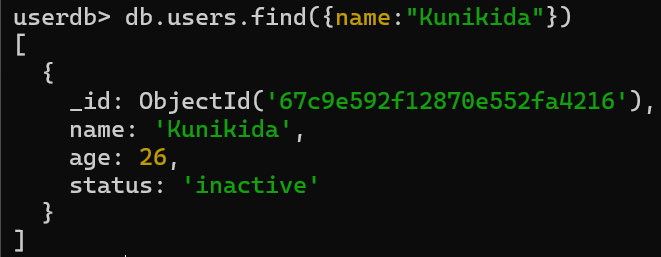








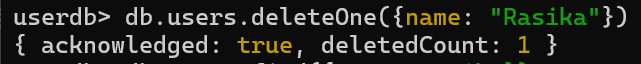


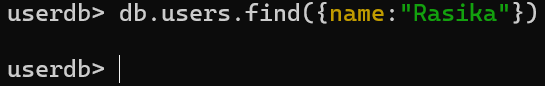


* **Update Operators:**
  + $set: Updates or creates a field.
  + $unset: Removes a field.
  + $inc: Increments a field.
  + $push: Adds an element to an array.
  + $pull: Removes an element from an array.

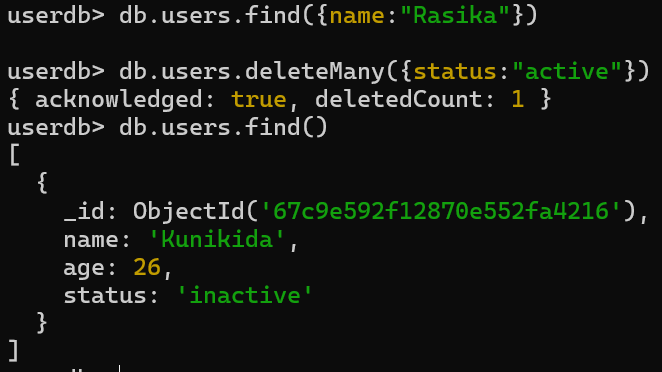
1. **Delete Operations:**

* **deleteOne():** Deletes a single document.





* **deleteMany():** Deletes multiple documents.



* **drop():** Deletes an entire collection.

