Roshan Shetty 13

Business Intelligence And Big Data Analytics

Mini Project

Aim: Implementation Of Mongo DB

Steps:

Steps to Install MongoDB:

1) Steps to install mongodb in debian based distro
2) Import the public key used by the package management system.
   - From a terminal, issue the following command to import the MongoDB public GPG Key
   - wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
   - The operation should respond with an `OK`.
   - However, if you receive an error indicating that `gnupg` is not installed, you can:
   - Install `gnupg` and its required libraries using the following command:
   - sudo apt-get install gnupg
   - Once installed, retry importing the key:
   - wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
3) Create a `/etc/apt/sources.list.d/mongodb-org-5.0.list` file for MongoDB.
   - Create the list file using the command appropriate for your version of Debian:
   - Debian 10 Buster:
   - echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/5.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list

4)Reload local package database.
   - Issue the following command to reload the local package database:
   - sudo apt-get update
5) Install the MongoDB packages.
   - You can install either the latest stable version of MongoDB or a specific version of MongoDB.
   - To install the latest stable version, issue the following
   - sudo apt-get install -y mongodb-org

Run Mongo DB:

1) Start MongoDB.
   - You can start the `mongod` process by issuing the following command:
   - sudo systemctl start mongod
   - If you receive an error similar to the following when starting `mongod`:
   - Failed to start mongod.service: Unit mongod.service not found.
   - Run the following command first:
   - sudo systemctl daemon-reload
   - Then run the start command above again.
2) Verify that MongoDB has started successfully.
   - sudo systemctl status mongod
   - You can optionally ensure that MongoDB will start following a system reboot by issuing the following command:

Roshan Shetty 13

- sudo systemctl enable mongod

3) Stop MongoDB.

- As needed, you can stop the mongod process by issuing the following command:
- sudo systemctl stop mongod

4) Restart MongoDB.

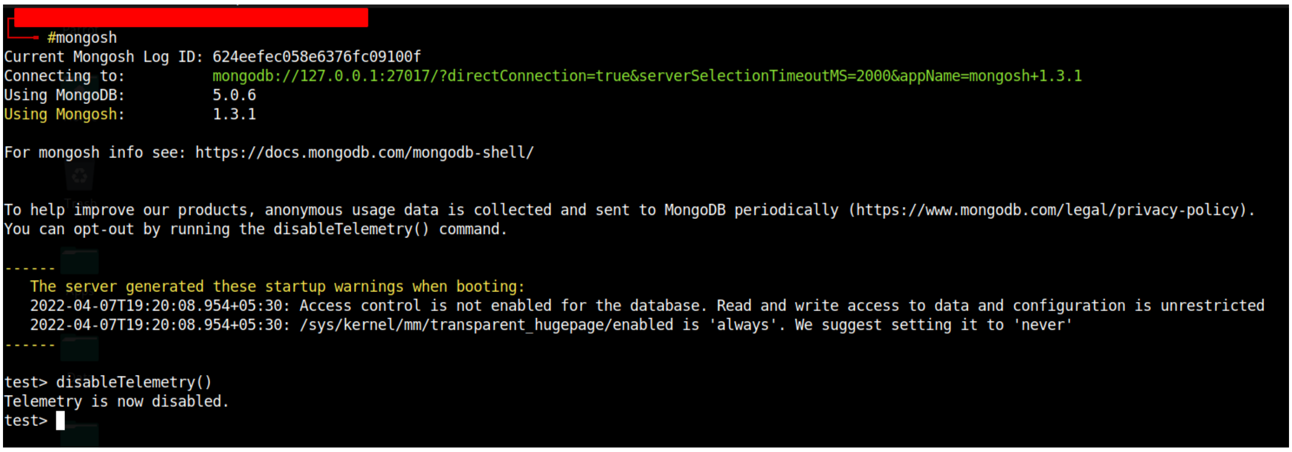
- You can restart the mongod process by issuing the following command:
- sudo systemctl restart mongod
- You can follow the state of the process for errors or important messages by watching the output in the /var/log/mongodb/mongod.log file.

5) Begin using MongoDB.

- Start a mongosh session on the same host machine as the mongod. You can run mongosh without any command-line options to connect to a mongod that is running on your localhost with default port 27017.
- mongosh

Start Mongosh:

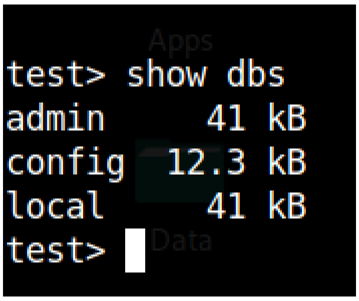
1) >mongosh



```
[redacted]
└──╼ #mongosh
Current Mongosh Log ID: 624eefec058e6376fc09100f
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.3.1
Using MongoDB:          5.0.6
Using Mongosh:          1.3.1

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:
   2022-04-07T19:20:08.954+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2022-04-07T19:20:08.954+05:30: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
------

test> disableTelemetry()
Telemetry is now disabled.
test> 
```

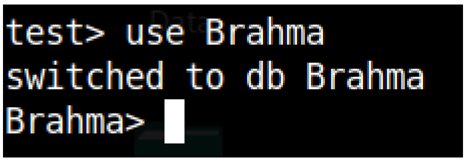
Roshan Shetty 13

2) Check for any existing databases.



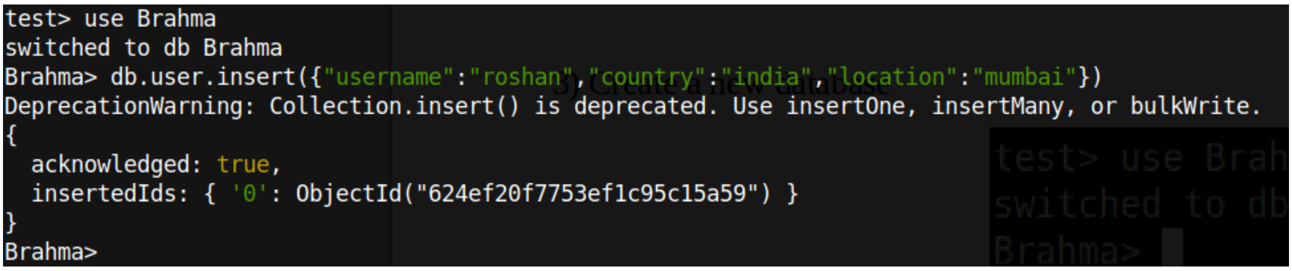
```
test> show dbs
admin      41 kB
config   12.3 kB
local      41 kB
test>
```

3) Create a new database



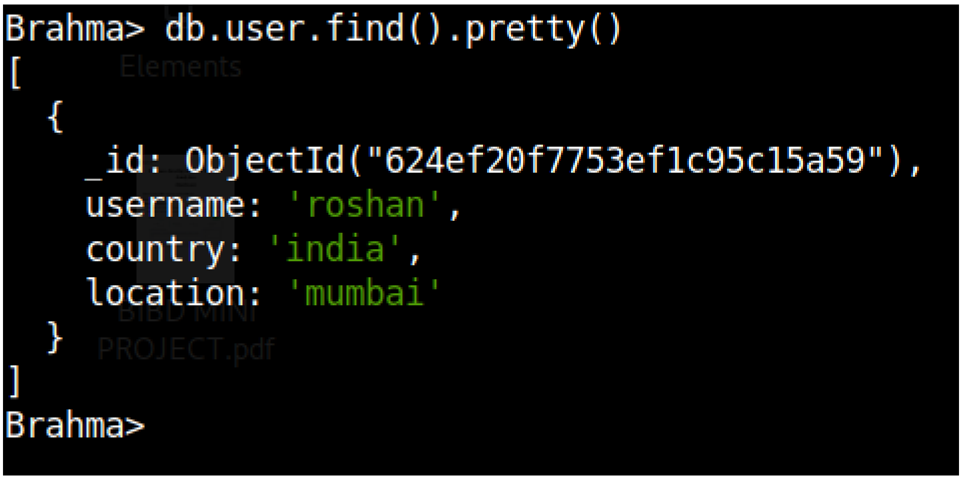
```
test> use Brahma
switched to db Brahma
Brahma>
```

4) Insert value to the DB



```
test> use Brahma
switched to db Brahma
Brahma> db.user.insert({"username":"roshan","country":"india","location":"mumbai"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("624ef20f7753ef1c95c15a59") }
}
Brahma>
```

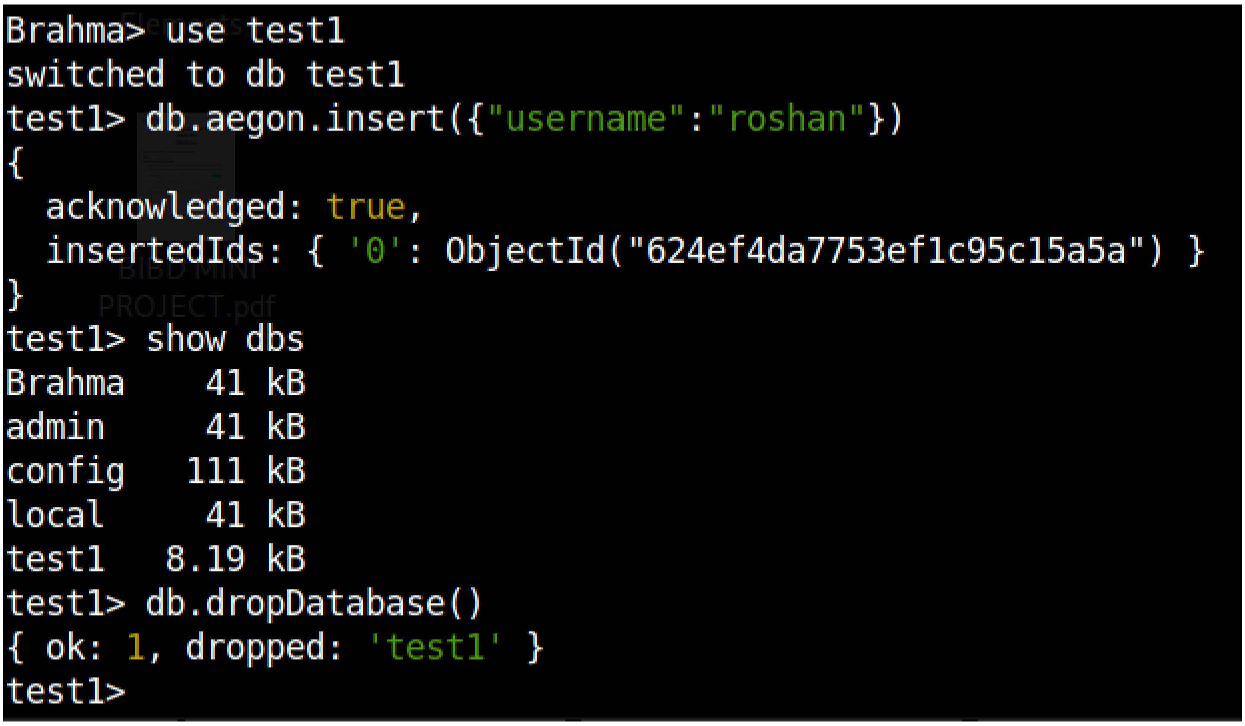
5) Show the inserted value



```
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'india',
    location: 'mumbai'
  }
]
Brahma>
```

Roshan Shetty 13

6) To drop a database

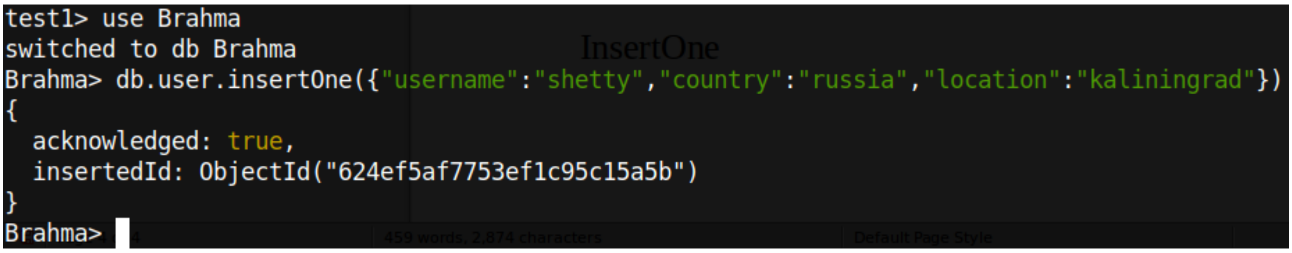


```
Brahma> use test1
switched to db test1
test1> db.aegon.insert({"username":"roshan"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("624ef4da7753ef1c95c15a5a") }
}
test1> show dbs
Brahma     41 kB
admin      41 kB
config    111 kB
local      41 kB
test1    8.19 kB
test1> db.dropDatabase()
{ ok: 1, dropped: 'test1' }
test1>
```

7) Create Value

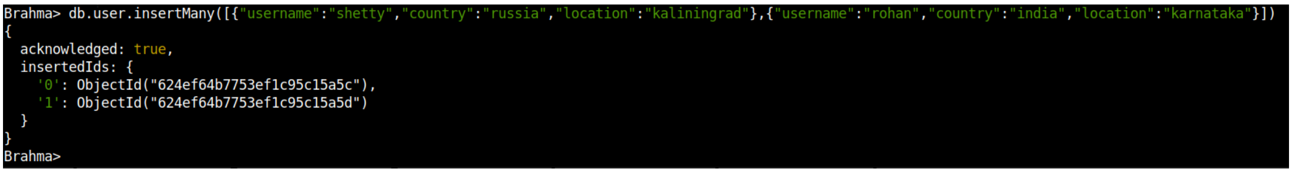
- insertOne
- insertMany

InsertOne



```
test1> use Brahma
switched to db Brahma
Brahma> db.user.insertOne({"username":"shetty","country":"russia","location":"kaliningrad"})
{
  acknowledged: true,
  insertedId: ObjectId("624ef5af7753ef1c95c15a5b")
}
Brahma>
```

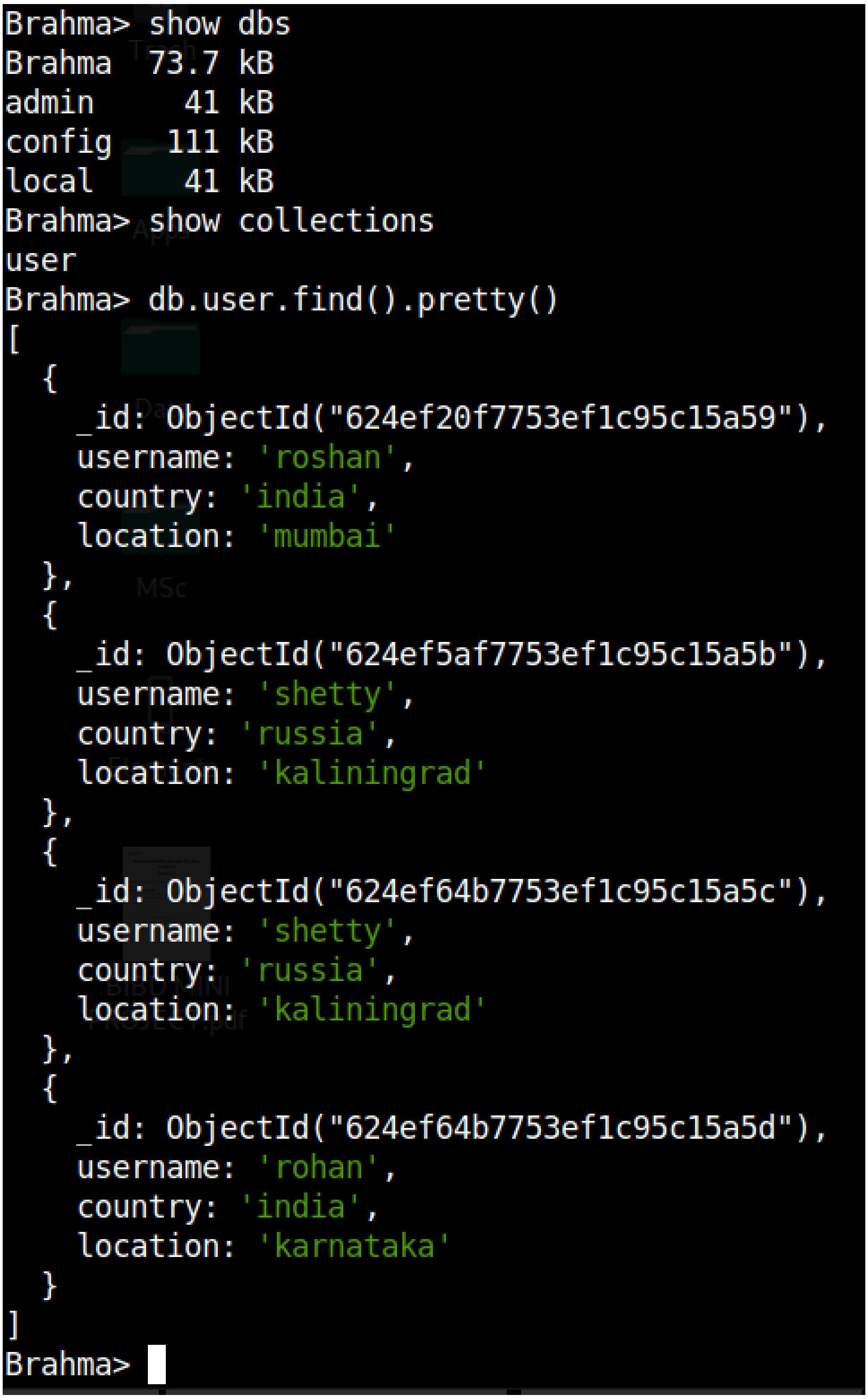
InsertMany



```
Brahma> db.user.insertMany([{"username":"shetty","country":"russia","location":"kaliningrad"},{"username":"rohan","country":"india","location":"karnataka"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("624ef64b7753ef1c95c15a5c"),
    '1': ObjectId("624ef64b7753ef1c95c15a5d")
  }
}
Brahma>
```

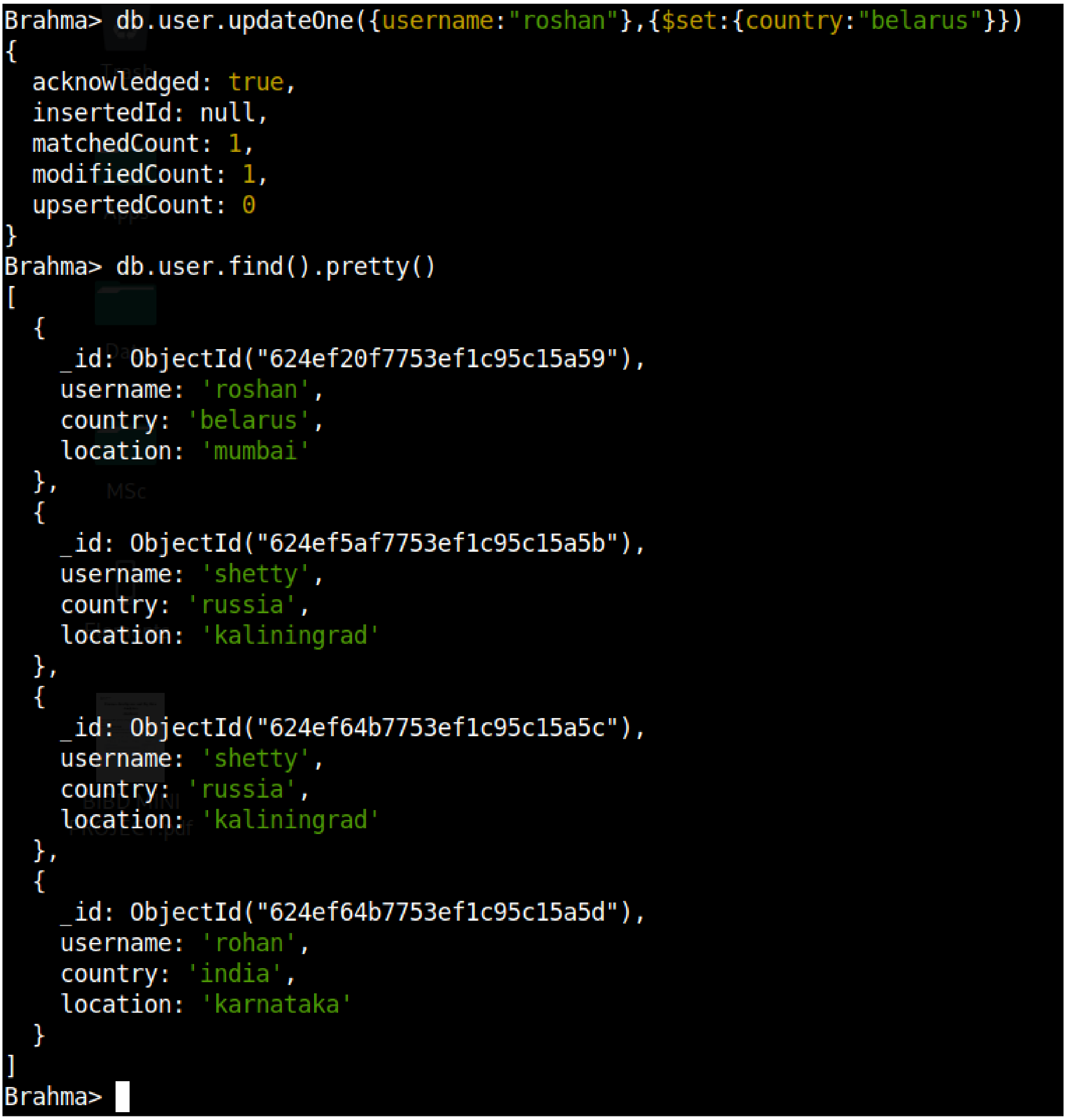
Check the Database

Roshan Shetty 13



```
Brahma> show dbs
Brahma  73.7 kB
admin     41 kB
config   111 kB
local     41 kB
Brahma> show collections
user
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'india',
    location: 'mumbai'
  },
  {
    _id: ObjectId("624ef5af7753ef1c95c15a5b"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5c"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5d"),
    username: 'rohan',
    country: 'india',
    location: 'karnataka'
  }
]
Brahma> 
```

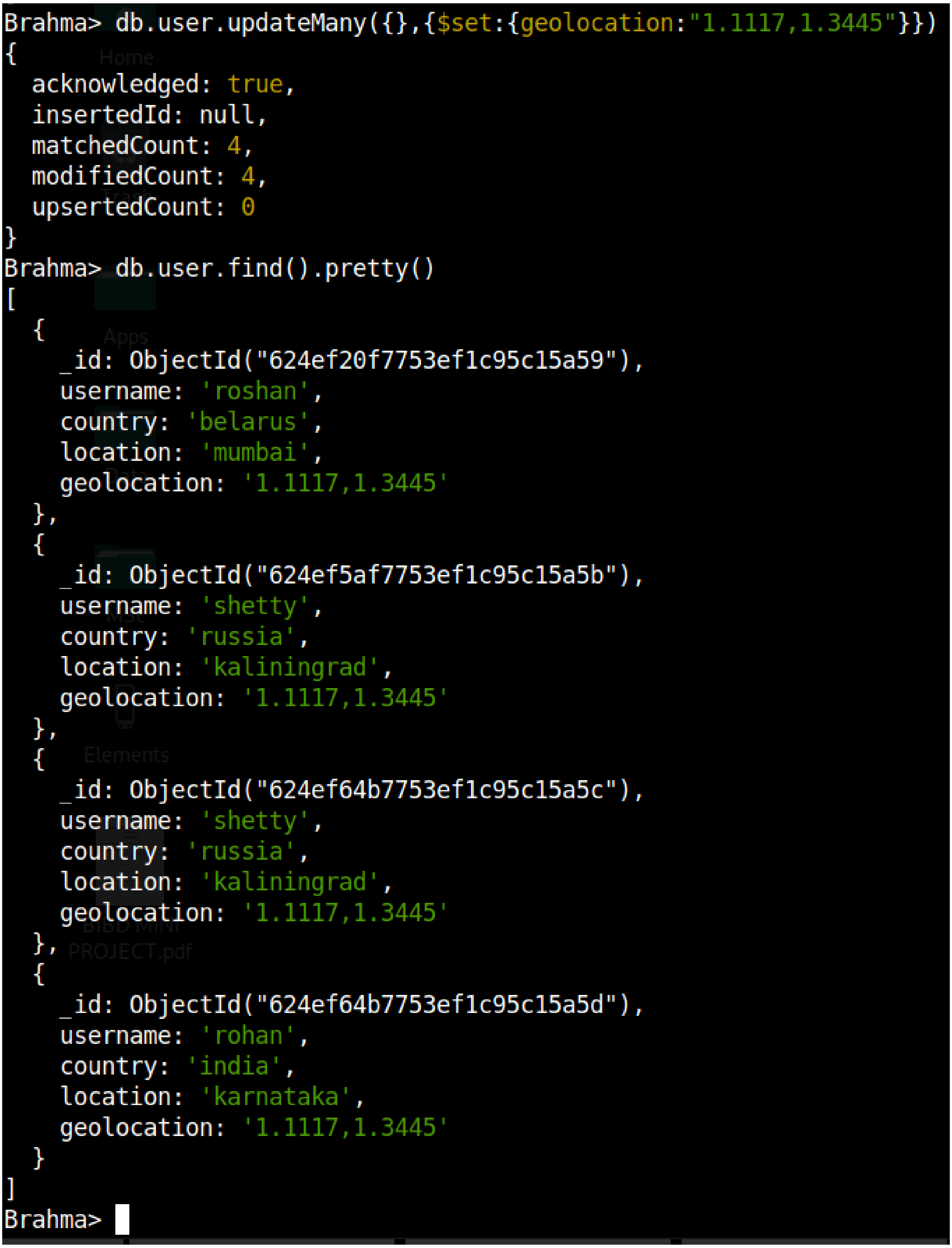
8) Updating a value



```
Brahma> db.user.updateOne({username:"roshan"},{$set:{country:"belarus"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'belarus',
    location: 'mumbai'
  },
  {
    _id: ObjectId("624ef5af7753ef1c95c15a5b"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5c"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5d"),
    username: 'rohan',
    country: 'india',
    location: 'karnataka'
  }
]
Brahma> 
```

Roshan Shetty 13

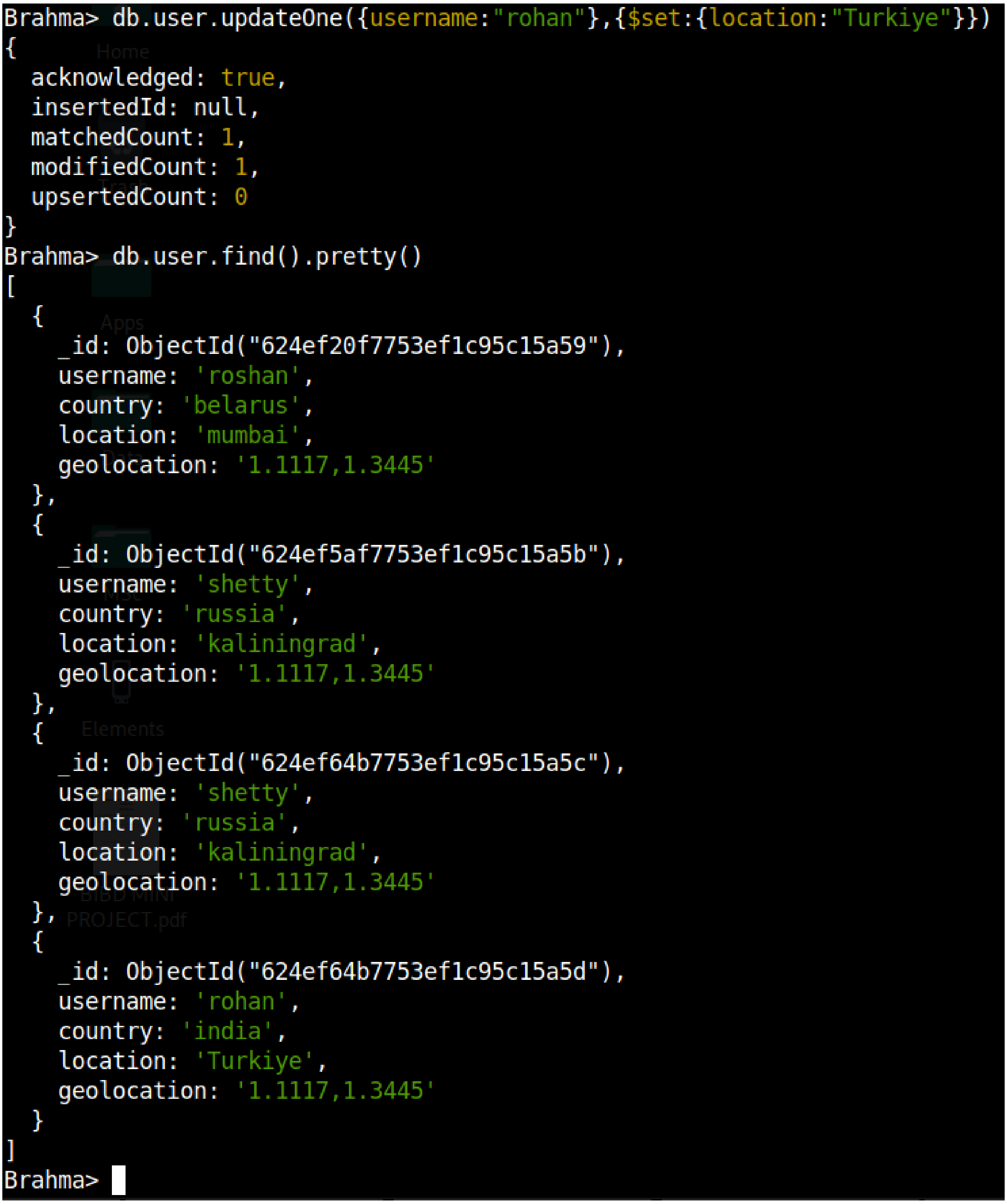
9) updateMany



```
Brahma> db.user.updateMany({},{$set:{geolocation:"1.1117,1.3445"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'belarus',
    location: 'mumbai',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef5af7753ef1c95c15a5b"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5c"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5d"),
    username: 'rohan',
    country: 'india',
    location: 'karnataka',
    geolocation: '1.1117,1.3445'
  }
]
Brahma> 
```

Roshan Shetty 13

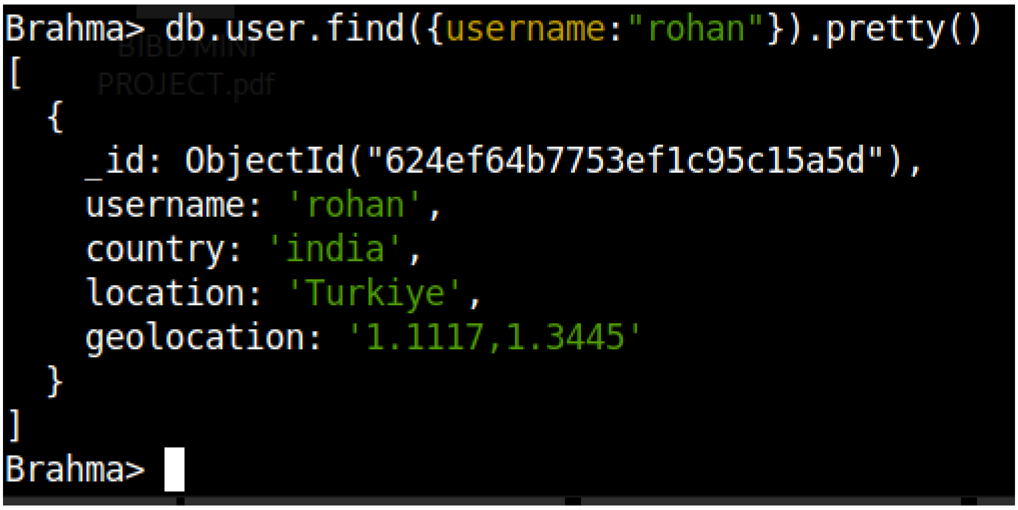
10) Update single value



```
Brahma> db.user.updateOne({username:"rohan"},{$set:{location:"Turkiye"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'belarus',
    location: 'mumbai',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef5af7753ef1c95c15a5b"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5c"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5d"),
    username: 'rohan',
    country: 'india',
    location: 'Turkiye',
    geolocation: '1.1117,1.3445'
  }
]
Brahma> 
```

Roshan Shetty 13

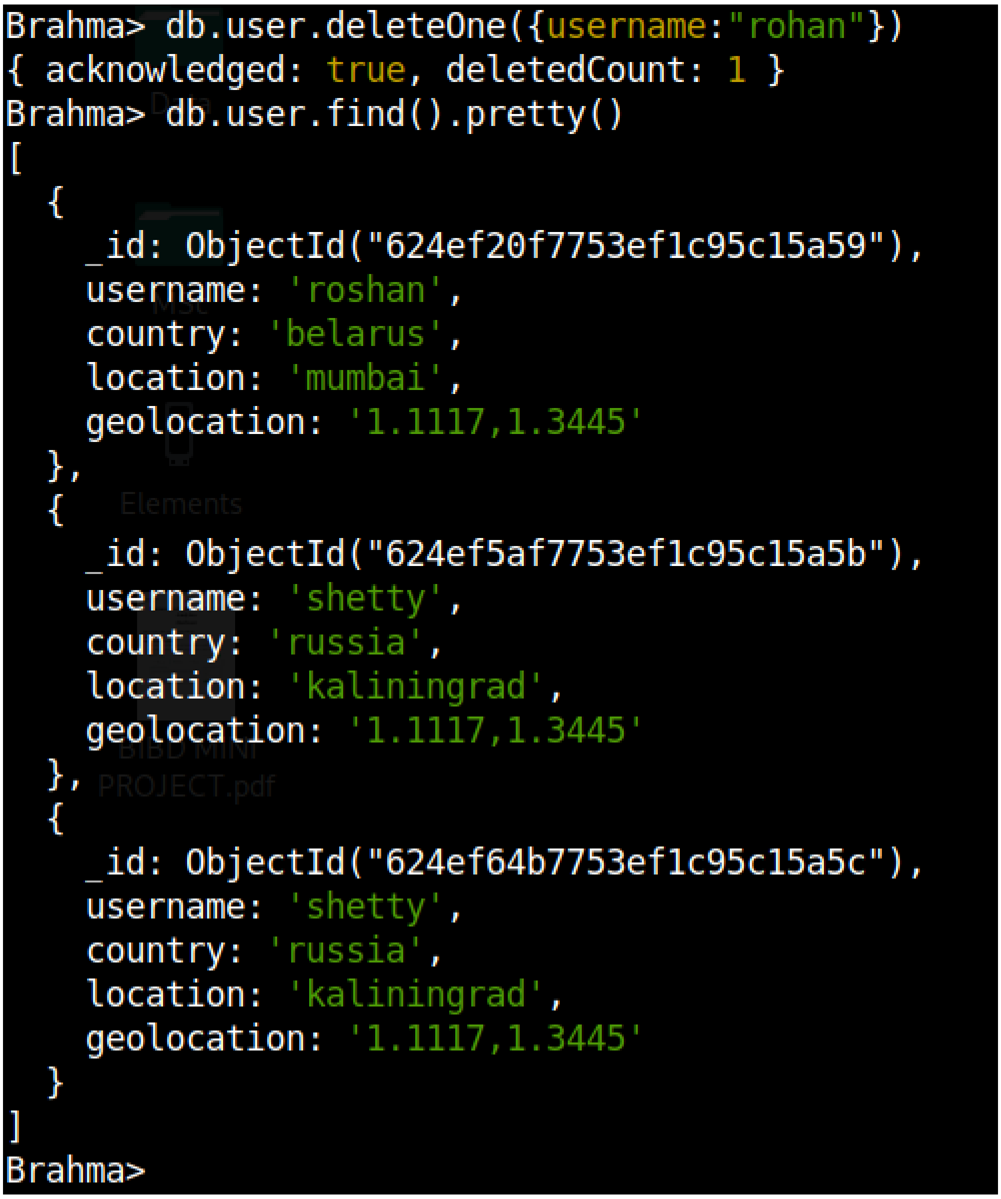
11) Find command to find particular tag



```
Brahma> db.user.find({username:"rohan"}).pretty()
[
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5d"),
    username: 'rohan',
    country: 'india',
    location: 'Turkiye',
    geolocation: '1.1117,1.3445'
  }
]
Brahma> 
```

Roshan Shetty 13

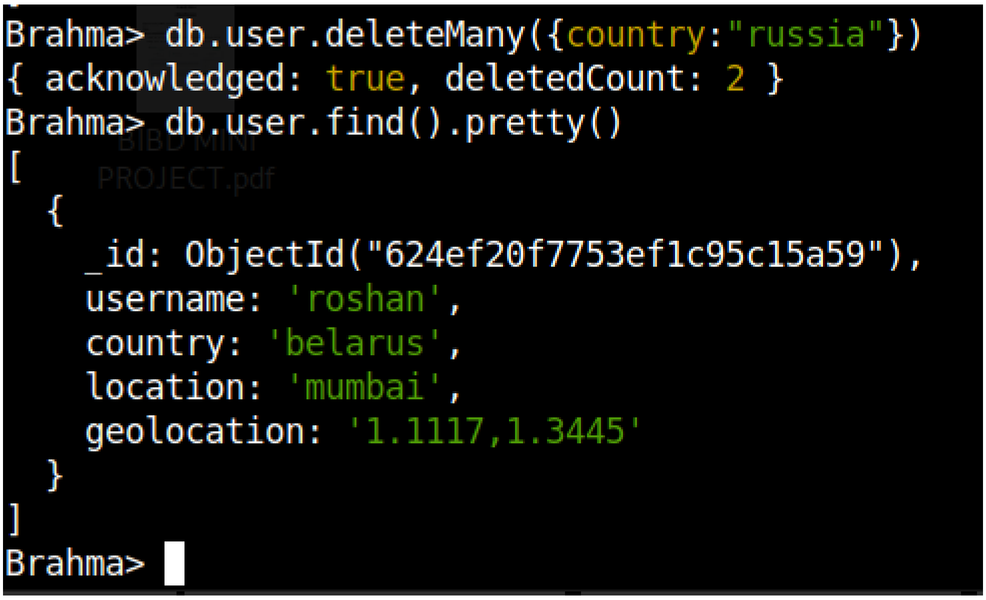
12) deleteOne



```
Brahma> db.user.deleteOne({username:"rohan"})
{ acknowledged: true, deletedCount: 1 }
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'belarus',
    location: 'mumbai',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef5af7753ef1c95c15a5b"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  },
  {
    _id: ObjectId("624ef64b7753ef1c95c15a5c"),
    username: 'shetty',
    country: 'russia',
    location: 'kaliningrad',
    geolocation: '1.1117,1.3445'
  }
]
Brahma>
```

Roshan Shetty 13

13) DeleteMany



```
Brahma> db.user.deleteMany({country:"russia"})
{ acknowledged: true, deletedCount: 2 }
Brahma> db.user.find().pretty()
[
  {
    _id: ObjectId("624ef20f7753ef1c95c15a59"),
    username: 'roshan',
    country: 'belarus',
    location: 'mumbai',
    geolocation: '1.1117,1.3445'
  }
]
Brahma> 
```