Database Management System

Foundation Questions

1. The **database** is a collection of information or data, that is stored electronically in the computer system.
   The **database management** is software to create and manage databases and allows users to create, read, update, and delete data in a database.

2.

SQL	**NoSQL**
It uses SQL language to communicate.	It uses MQL, JSON(JavaScript Object Notation), XML, and YAML languages to communicate.
Fixed data scheme	More flexible
Table-based data like rows and columns.	Collection-based data like documents.
Example: MySQL, OSTACLE SQL, PostgreSQL, Microsoft SQL server.	Example: MongoDB, DynamoDB.

3. **SQL databases** are used for structured data, and **NoSQL databases** for structured, semi-structured, and unstructured data.

4. **Graph database** for shorting data. It uses nodes, edges, and properties instead of tables or documents to represent or store data. The edge represents the relationship between the nodes and it is commonly referred to as a NoSQL.

5. **API** stands for Application Programme Interface. It is used to integrate new applications with existing software systems and software is used as an intermediary that allows two applications to connect.

6.

7. **MongoDB Atlas** is a fully managed cloud database that handles all the complexities of deploying, managing, and healing your deployments on the cloud service provider of your choice.

8. storing, maintaining, and accessing any sort of data. We can make a website without a database. When we make a statistics website it does not need a database.

9. **Static websites** are made as fixed websites we can not change anything on the webpage, otherwise, if we want to change we have to change our code. But when it comes to a **dynamic website** we can make a change in the web page.

10.

Database Management System with Business Project

1. **SQL:** My SQL, Oracle SQL, Postgre SQL, Microsoft SQL server, MariaDB
   NoSQL: MongoDB, Amazon DynamoDB, Cassandra, CouchDB, HBase

2. **Problem**: In this cutting-edge era, there are difficult to find experienced doctors and medical equipment.
   Solution: Due to this, I am going to develop a platform between the public and the medical system to fill this gap with video and audio consultation and

3. NoSQL

4. **Model**: It shows the logical structure of a database.

5. Doctors, Users, Products

6. A **database instance** is a set of memory structures that manage database files. That changes over time when we add or delete from the database.

7.
   ```
   {
    “_id”: “246513542156743146854”,
   “Name”: “Janujah”,
   “Email”: “janujahsivarattinam@gmail.com”,
   “Age”: 22
   }
   ```

8. Lucidchart
   Dbdiagram
   DbSchema
   Dezign
   SqlDBM

9. Users’ passwords are not stored as plain text, instead, they should be stored by using a combination of hashing and salting techniques.

10. **Hashing** is a Database Management system technique for searching for needed data on the disc without using an index structure.

MongoDB

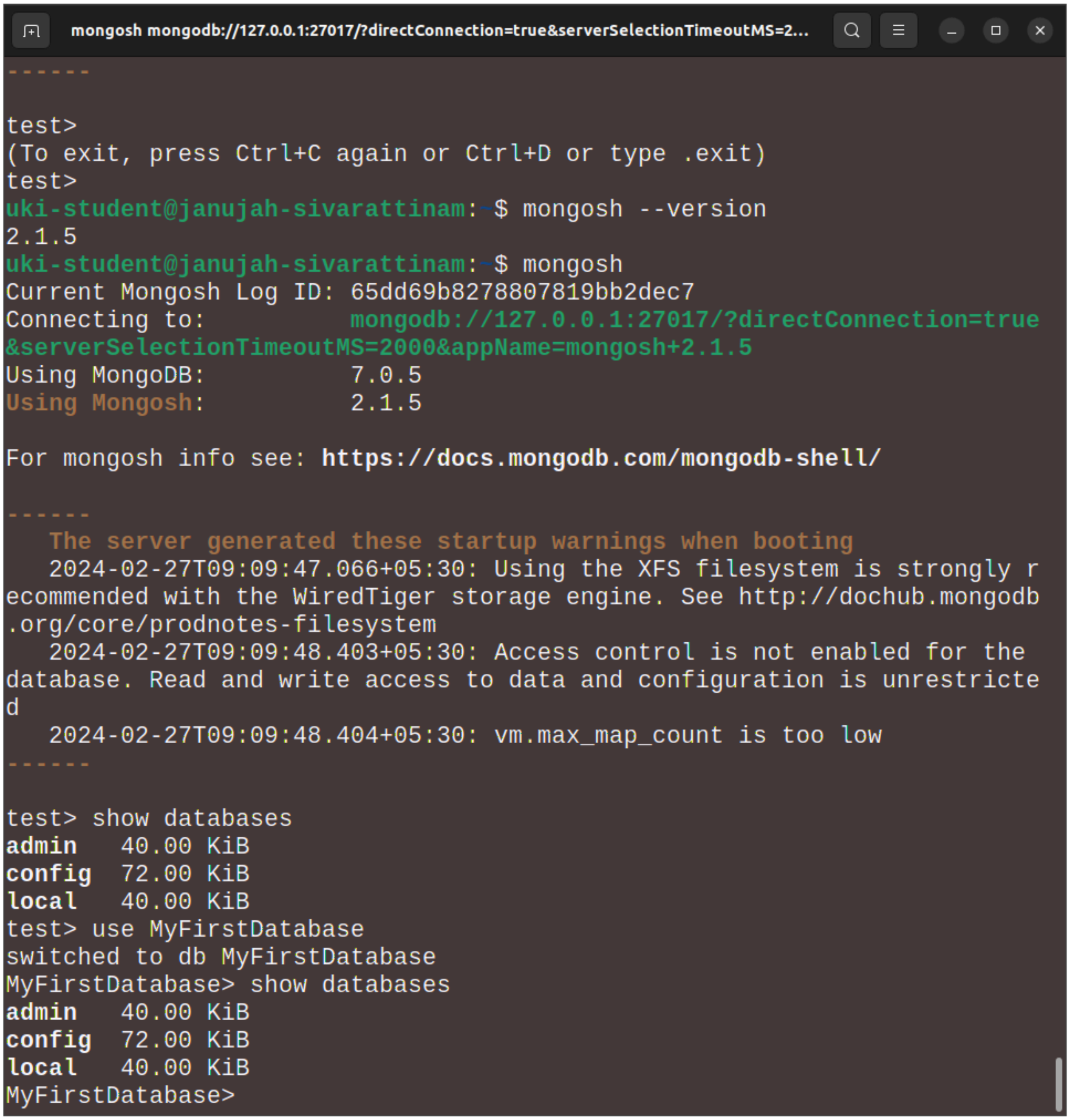
1. a.navigate to https://www.mongodb.com/cloud/atlas/register and register through the Email and confirm it.
   b.Setting up a cluster in MongoDB.
   c.Create a shared cluster.
   d.Access the MongoDB Atlas Cluster.
   e.Allow the access to your IP Address.
   f.Create a cluster user.
   g.Generating a Database Connection String

2. **MongoDB Community software:**
   Mongosh(MongoDB Shell): An interactive JavaScript interface to MongoDB.

3. **The MongoDB shell** is the quickest way to connect and work with MongoDB and Atlas.

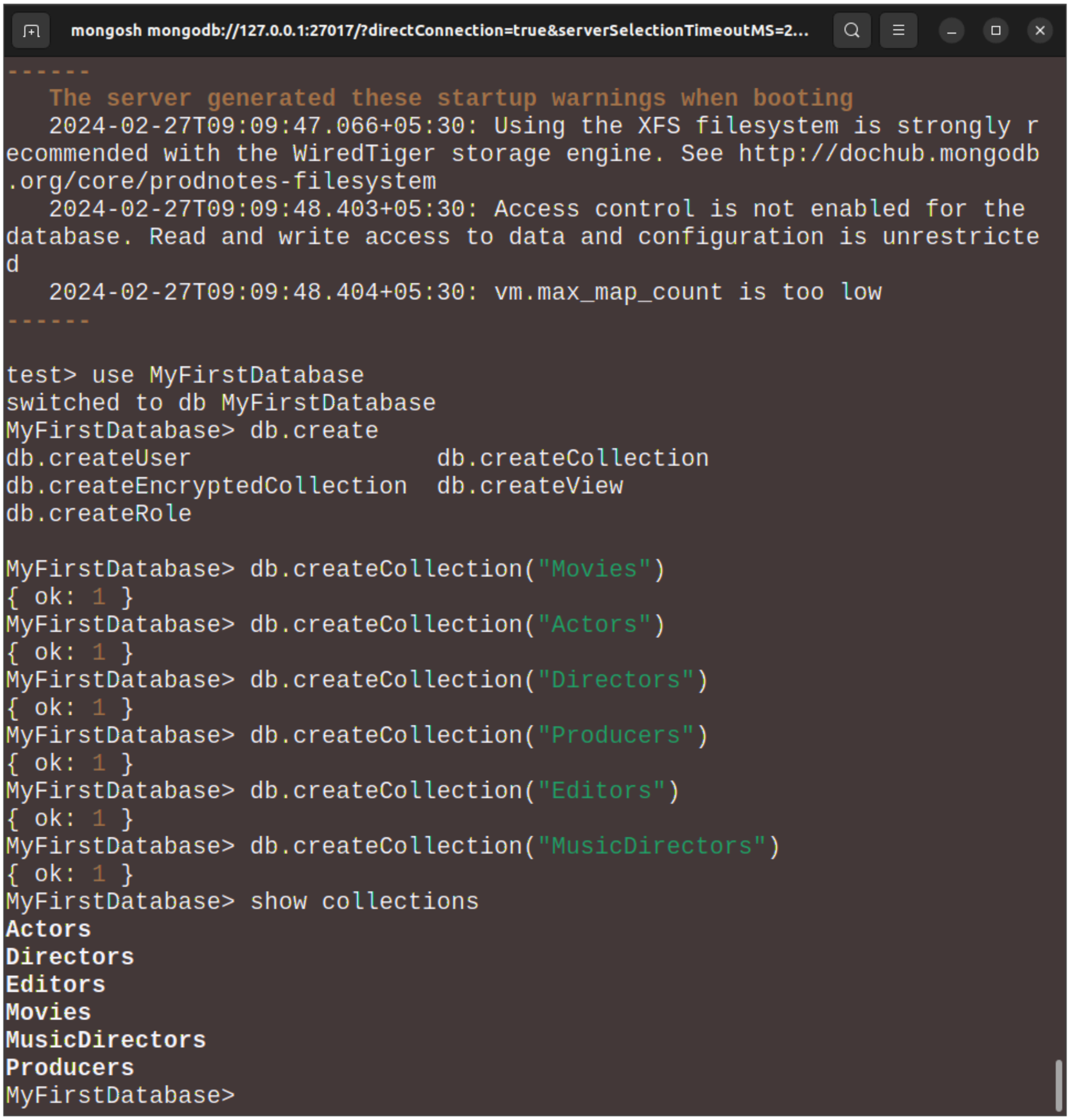
4. a.Go to the MongoDB official page(https://www.mongodb.com/)
   b.Go to the Docs.
   c.Click the Docs Menu.
   d.Click the Database Manual.
   e.Click the installation.
   f.Go to the Install MongoDB Community Edition.
   g.Select your OS.
   h.Follow the instructions.

5.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

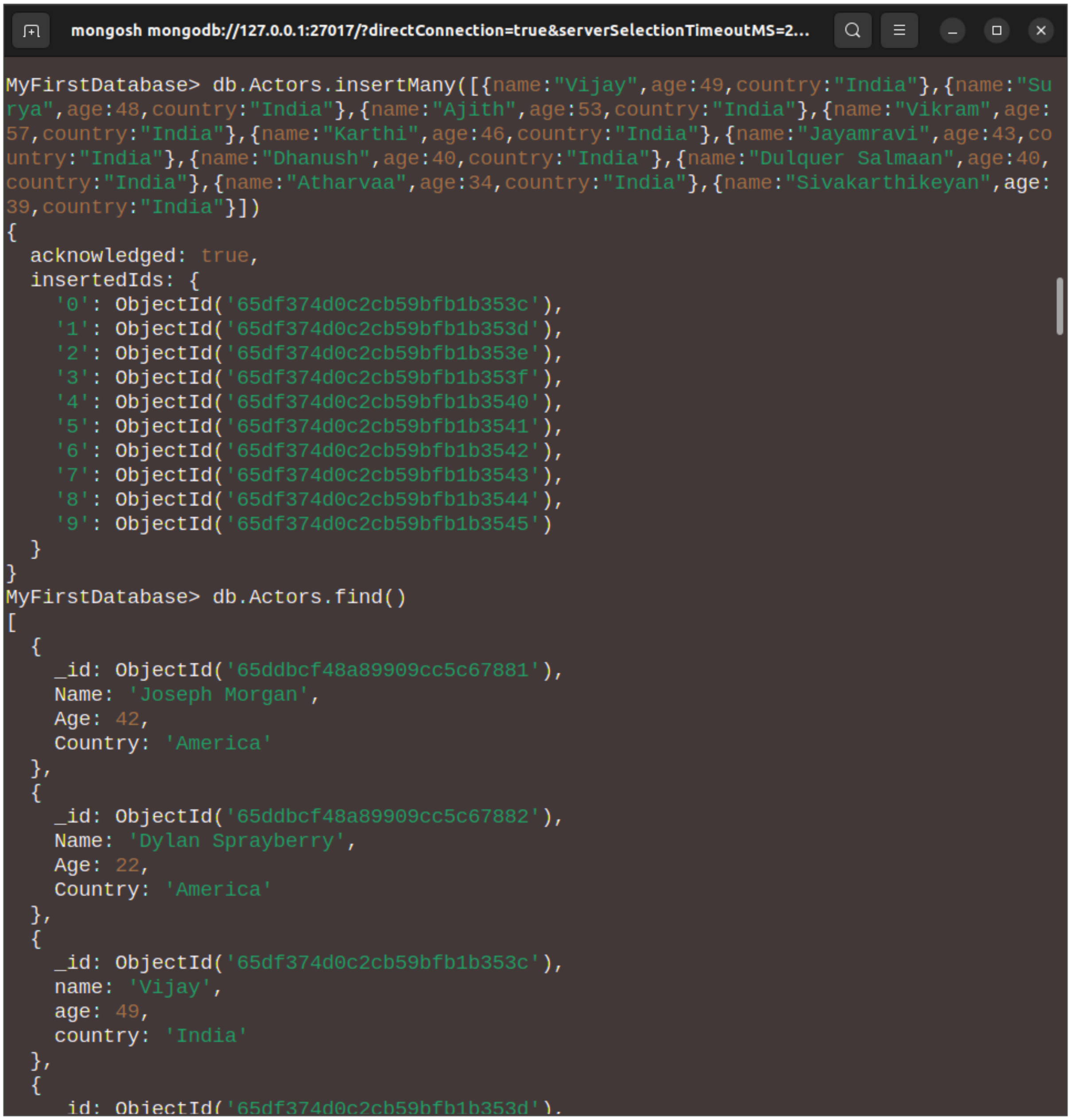
```
------

test>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
test>
uki-student@janujah-sivarattinam:~$ mongosh --version
2.1.5
uki-student@janujah-sivarattinam:~$ mongosh
Current Mongosh Log ID: 65dd69b8278807819bb2dec7
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true
&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:          7.0.5
Using Mongosh:          2.1.5

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

------
   The server generated these startup warnings when booting
   2024-02-27T09:09:47.066+05:30: Using the XFS filesystem is strongly r
ecommended with the WiredTiger storage engine. See http://dochub.mongodb
.org/core/prodnotes-filesystem
   2024-02-27T09:09:48.403+05:30: Access control is not enabled for the
database. Read and write access to data and configuration is unrestricte
d
   2024-02-27T09:09:48.404+05:30: vm.max_map_count is too low
------

test> show databases
admin    40.00 KiB
config   72.00 KiB
local    40.00 KiB
test> use MyFirstDatabase
switched to db MyFirstDatabase
MyFirstDatabase> show databases
admin    40.00 KiB
config   72.00 KiB
local    40.00 KiB
MyFirstDatabase>
```

6.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

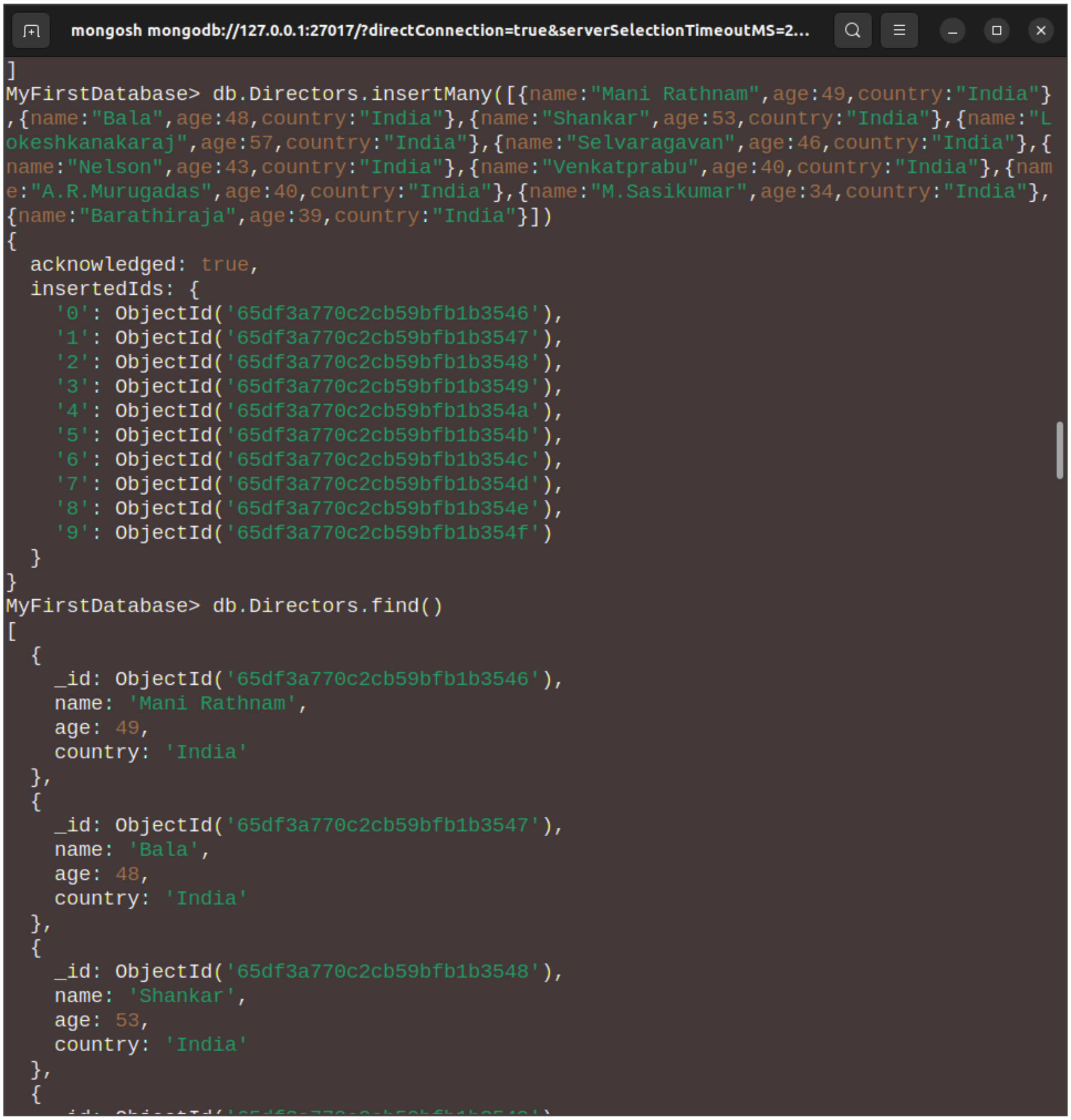
```
------
   The server generated these startup warnings when booting
   2024-02-27T09:09:47.066+05:30: Using the XFS filesystem is strongly r
ecommended with the WiredTiger storage engine. See http://dochub.mongodb
.org/core/prodnotes-filesystem
   2024-02-27T09:09:48.403+05:30: Access control is not enabled for the
database. Read and write access to data and configuration is unrestricte
d
   2024-02-27T09:09:48.404+05:30: vm.max_map_count is too low
------

test> use MyFirstDatabase
switched to db MyFirstDatabase
MyFirstDatabase> db.create
db.createUser                     db.createCollection
db.createEncryptedCollection  db.createView
db.createRole

MyFirstDatabase> db.createCollection("Movies")
{ ok: 1 }
MyFirstDatabase> db.createCollection("Actors")
{ ok: 1 }
MyFirstDatabase> db.createCollection("Directors")
{ ok: 1 }
MyFirstDatabase> db.createCollection("Producers")
{ ok: 1 }
MyFirstDatabase> db.createCollection("Editors")
{ ok: 1 }
MyFirstDatabase> db.createCollection("MusicDirectors")
{ ok: 1 }
MyFirstDatabase> show collections
Actors
Directors
Editors
Movies
MusicDirectors
Producers
MyFirstDatabase>
```

7. a. Actor’s


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
MyFirstDatabase> db.Actors.insertMany([{name:"Vijay",age:49,country:"India"},{name:"Su
rya",age:48,country:"India"},{name:"Ajith",age:53,country:"India"},{name:"Vikram",age:
57,country:"India"},{name:"Karthi",age:46,country:"India"},{name:"Jayamravi",age:43,co
untry:"India"},{name:"Dhanush",age:40,country:"India"},{name:"Dulquer Salmaan",age:40,
country:"India"},{name:"Atharvaa",age:34,country:"India"},{name:"Sivakarthikeyan",age:
39,country:"India"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df374d0c2cb59bfb1b353c'),
    '1': ObjectId('65df374d0c2cb59bfb1b353d'),
    '2': ObjectId('65df374d0c2cb59bfb1b353e'),
    '3': ObjectId('65df374d0c2cb59bfb1b353f'),
    '4': ObjectId('65df374d0c2cb59bfb1b3540'),
    '5': ObjectId('65df374d0c2cb59bfb1b3541'),
    '6': ObjectId('65df374d0c2cb59bfb1b3542'),
    '7': ObjectId('65df374d0c2cb59bfb1b3543'),
    '8': ObjectId('65df374d0c2cb59bfb1b3544'),
    '9': ObjectId('65df374d0c2cb59bfb1b3545')
  }
}
MyFirstDatabase> db.Actors.find()
[
  {
    _id: ObjectId('65ddbcf48a89909cc5c67881'),
    Name: 'Joseph Morgan',
    Age: 42,
    Country: 'America'
  },
  {
    _id: ObjectId('65ddbcf48a89909cc5c67882'),
    Name: 'Dylan Sprayberry',
    Age: 22,
    Country: 'America'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b353c'),
    name: 'Vijay',
    age: 49,
    country: 'India'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b353d'),
```

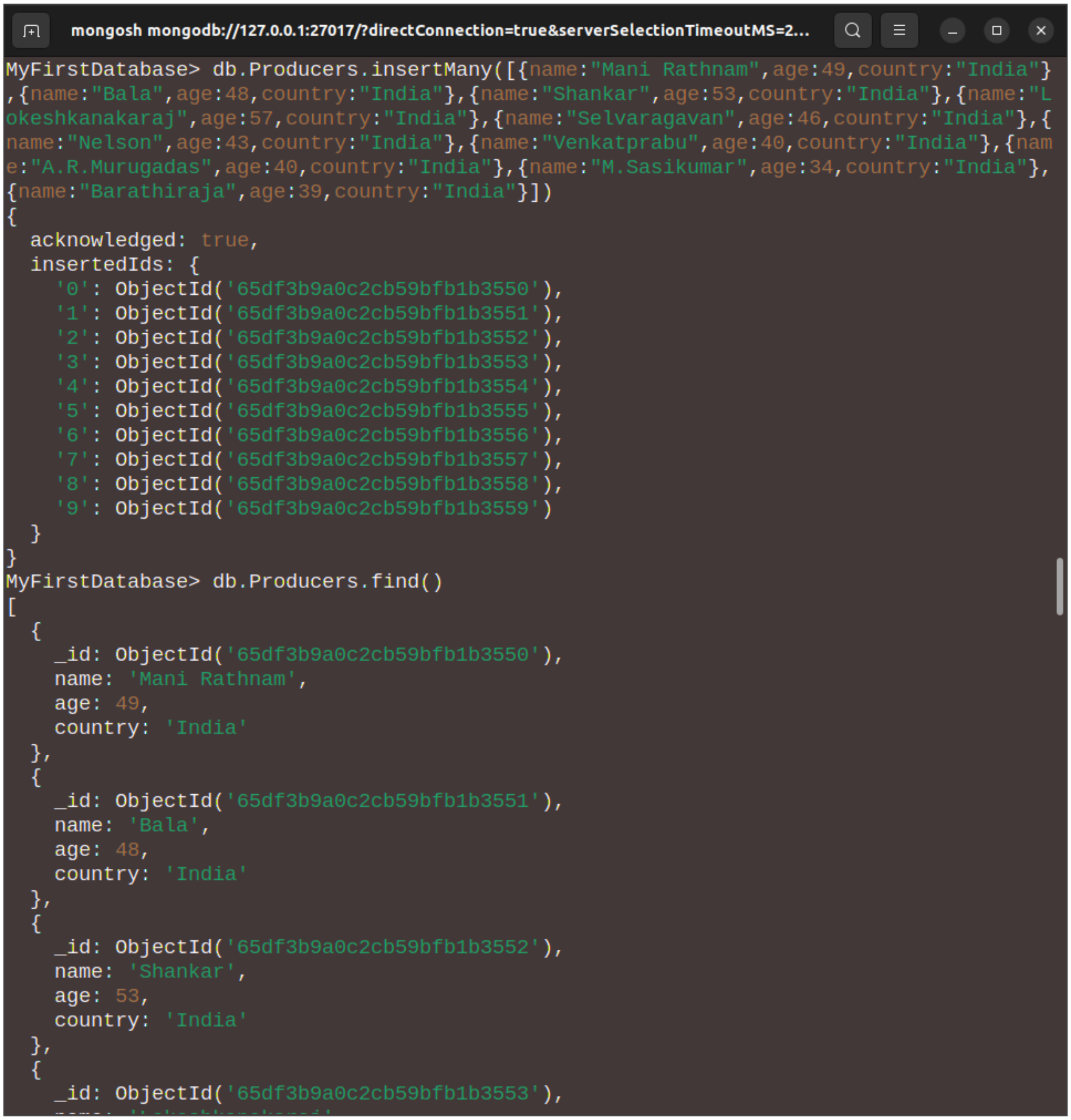
Director’s



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
]
MyFirstDatabase> db.Directors.insertMany([{name:"Mani Rathnam",age:49,country:"India"}
,{name:"Bala",age:48,country:"India"},{name:"Shankar",age:53,country:"India"},{name:"L
okeshkanakaraj",age:57,country:"India"},{name:"Selvaragavan",age:46,country:"India"},{
name:"Nelson",age:43,country:"India"},{name:"Venkatprabu",age:40,country:"India"},{nam
e:"A.R.Murugadas",age:40,country:"India"},{name:"M.Sasikumar",age:34,country:"India"},
{name:"Barathiraja",age:39,country:"India"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df3a770c2cb59bfb1b3546'),
    '1': ObjectId('65df3a770c2cb59bfb1b3547'),
    '2': ObjectId('65df3a770c2cb59bfb1b3548'),
    '3': ObjectId('65df3a770c2cb59bfb1b3549'),
    '4': ObjectId('65df3a770c2cb59bfb1b354a'),
    '5': ObjectId('65df3a770c2cb59bfb1b354b'),
    '6': ObjectId('65df3a770c2cb59bfb1b354c'),
    '7': ObjectId('65df3a770c2cb59bfb1b354d'),
    '8': ObjectId('65df3a770c2cb59bfb1b354e'),
    '9': ObjectId('65df3a770c2cb59bfb1b354f')
  }
}
MyFirstDatabase> db.Directors.find()
[
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3546'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India'
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3547'),
    name: 'Bala',
    age: 48,
    country: 'India'
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3548'),
    name: 'Shankar',
    age: 53,
    country: 'India'
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3549'),
```

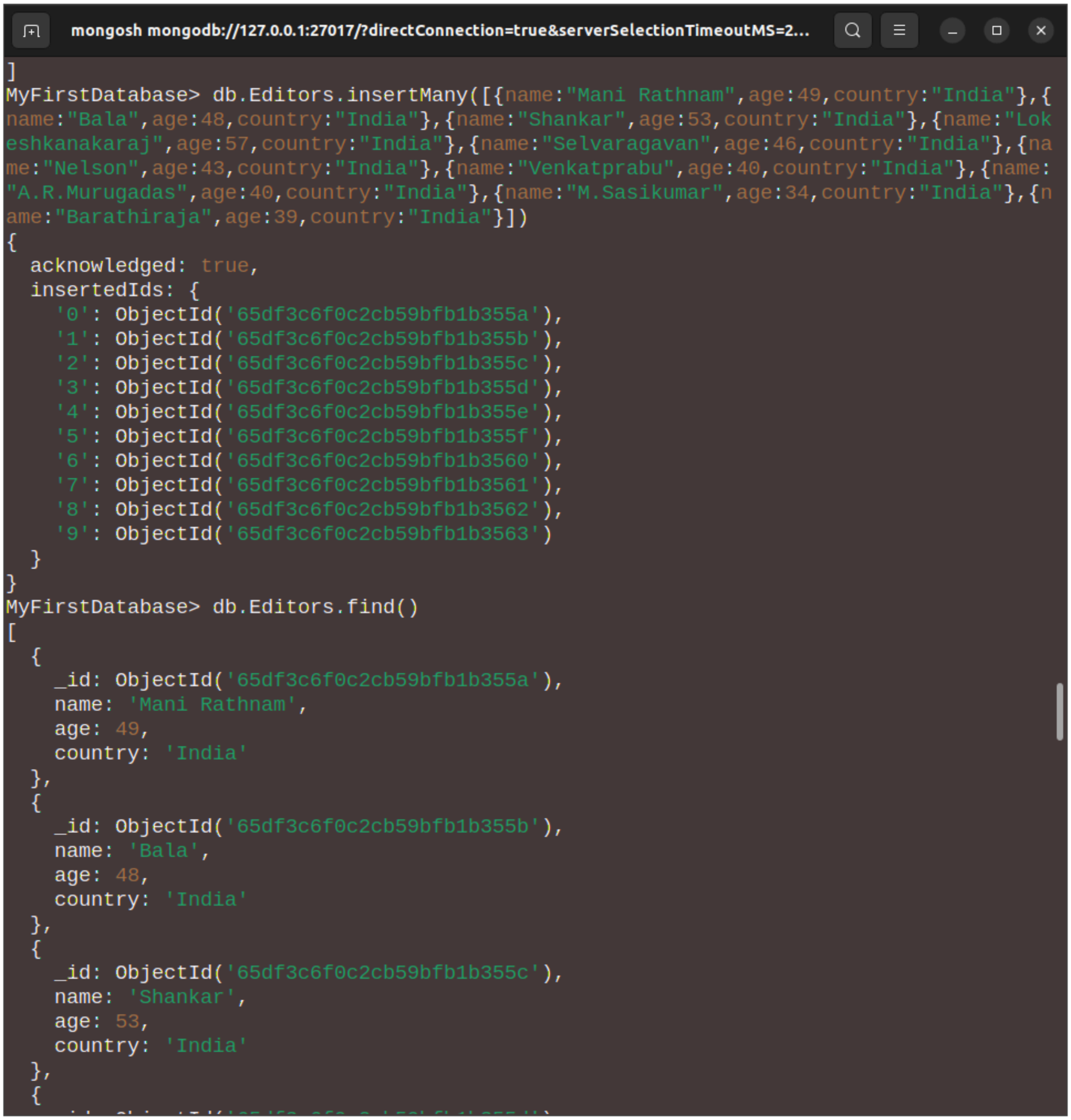
Producer’s



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

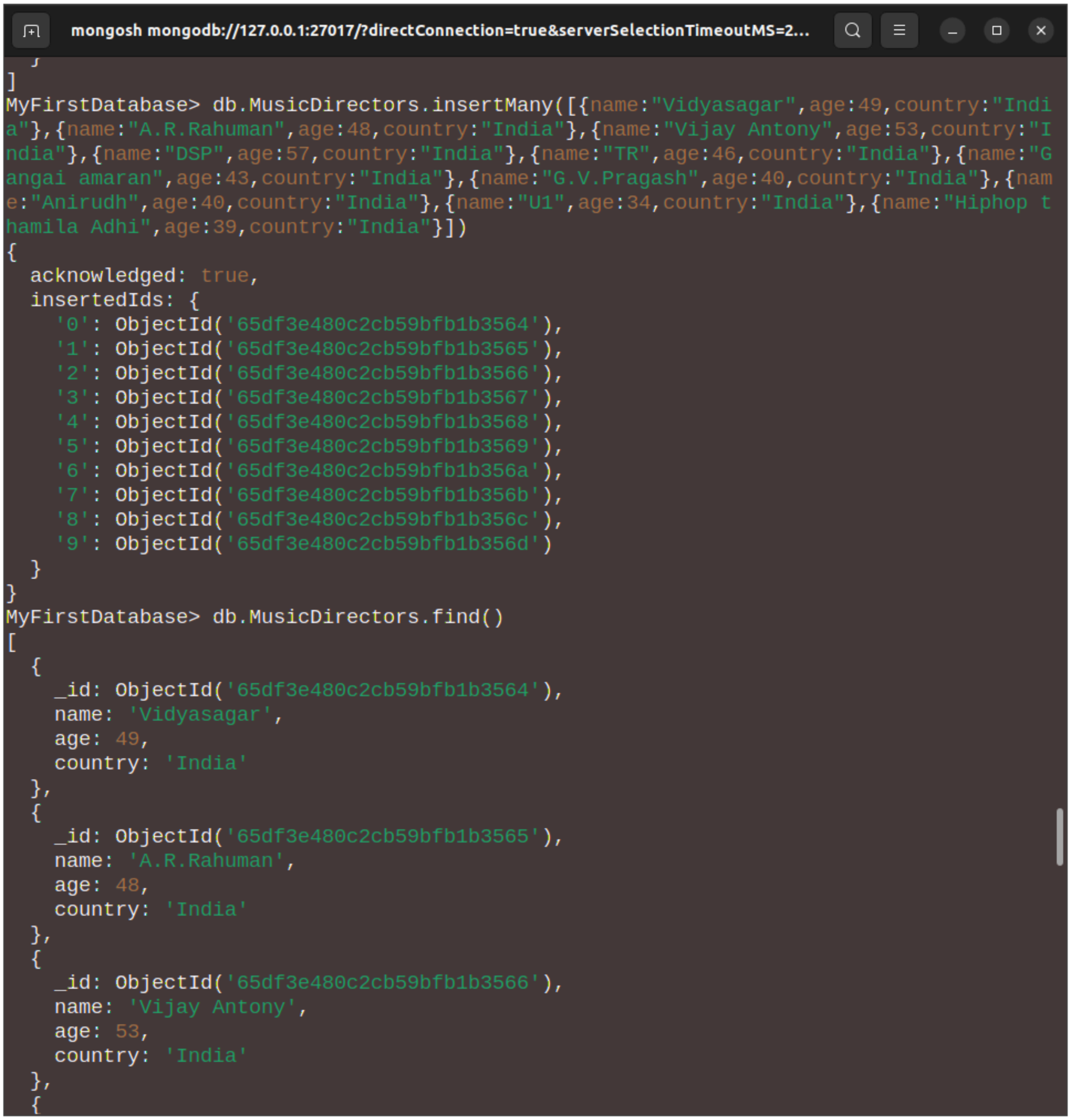
```
MyFirstDatabase> db.Producers.insertMany([{name:"Mani Rathnam",age:49,country:"India"}
,{name:"Bala",age:48,country:"India"},{name:"Shankar",age:53,country:"India"},{name:"L
okeshkanakaraj",age:57,country:"India"},{name:"Selvaragavan",age:46,country:"India"},{
name:"Nelson",age:43,country:"India"},{name:"Venkatprabu",age:40,country:"India"},{nam
e:"A.R.Murugadas",age:40,country:"India"},{name:"M.Sasikumar",age:34,country:"India"},
{name:"Barathiraja",age:39,country:"India"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df3b9a0c2cb59bfb1b3550'),
    '1': ObjectId('65df3b9a0c2cb59bfb1b3551'),
    '2': ObjectId('65df3b9a0c2cb59bfb1b3552'),
    '3': ObjectId('65df3b9a0c2cb59bfb1b3553'),
    '4': ObjectId('65df3b9a0c2cb59bfb1b3554'),
    '5': ObjectId('65df3b9a0c2cb59bfb1b3555'),
    '6': ObjectId('65df3b9a0c2cb59bfb1b3556'),
    '7': ObjectId('65df3b9a0c2cb59bfb1b3557'),
    '8': ObjectId('65df3b9a0c2cb59bfb1b3558'),
    '9': ObjectId('65df3b9a0c2cb59bfb1b3559')
  }
}
MyFirstDatabase> db.Producers.find()
[
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3550'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India'
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3551'),
    name: 'Bala',
    age: 48,
    country: 'India'
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3552'),
    name: 'Shankar',
    age: 53,
    country: 'India'
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3553'),
    name: 'Lokeshkanakaraj'
```

Editor’s


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

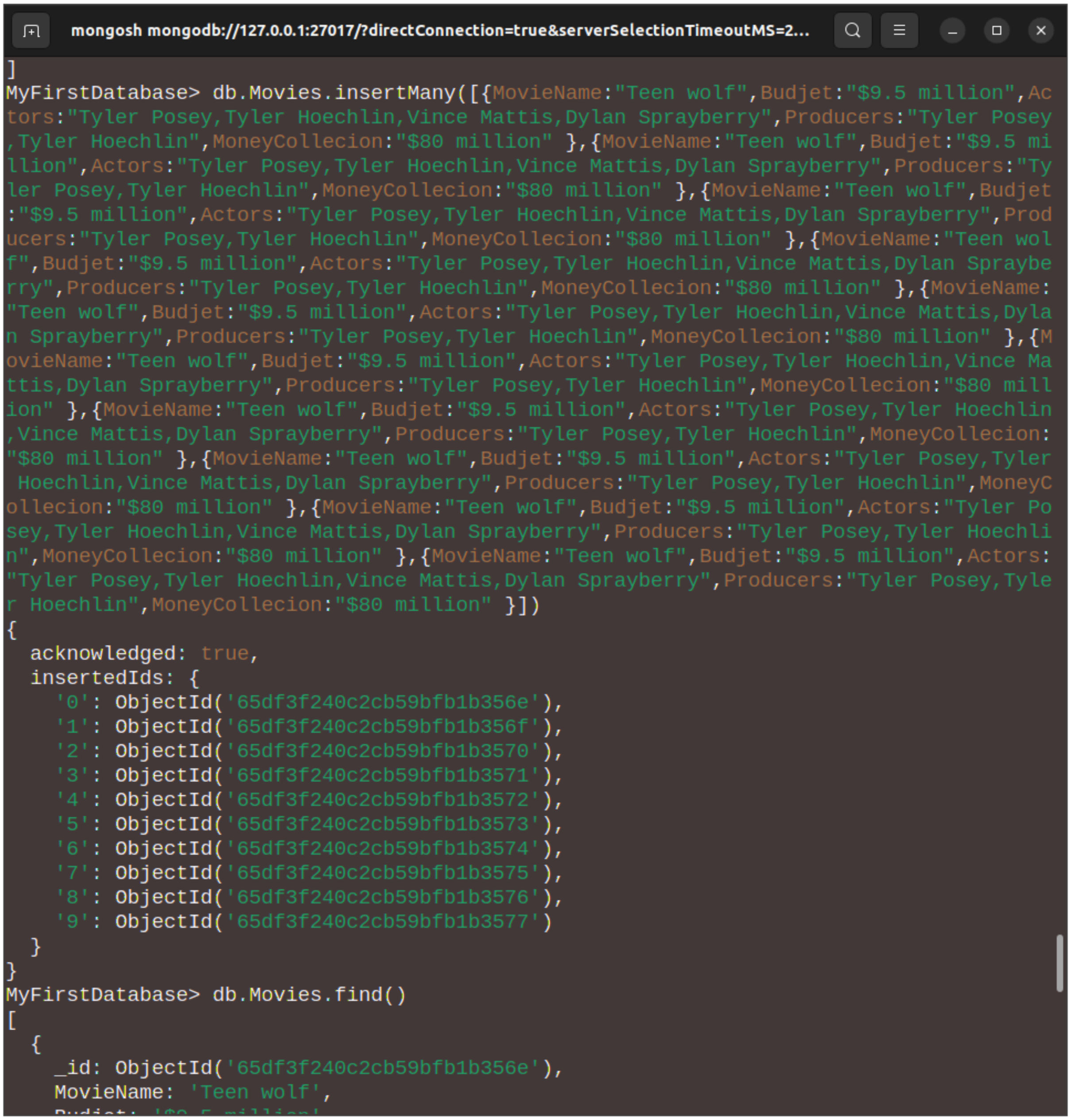
```
]
MyFirstDatabase> db.Editors.insertMany([{name:"Mani Rathnam",age:49,country:"India"},{
name:"Bala",age:48,country:"India"},{name:"Shankar",age:53,country:"India"},{name:"Lok
eshkanakaraj",age:57,country:"India"},{name:"Selvaragavan",age:46,country:"India"},{na
me:"Nelson",age:43,country:"India"},{name:"Venkatprabu",age:40,country:"India"},{name:
"A.R.Murugadas",age:40,country:"India"},{name:"M.Sasikumar",age:34,country:"India"},{n
ame:"Barathiraja",age:39,country:"India"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df3c6f0c2cb59bfb1b355a'),
    '1': ObjectId('65df3c6f0c2cb59bfb1b355b'),
    '2': ObjectId('65df3c6f0c2cb59bfb1b355c'),
    '3': ObjectId('65df3c6f0c2cb59bfb1b355d'),
    '4': ObjectId('65df3c6f0c2cb59bfb1b355e'),
    '5': ObjectId('65df3c6f0c2cb59bfb1b355f'),
    '6': ObjectId('65df3c6f0c2cb59bfb1b3560'),
    '7': ObjectId('65df3c6f0c2cb59bfb1b3561'),
    '8': ObjectId('65df3c6f0c2cb59bfb1b3562'),
    '9': ObjectId('65df3c6f0c2cb59bfb1b3563')
  }
}
MyFirstDatabase> db.Editors.find()
[
  {
    _id: ObjectId('65df3c6f0c2cb59bfb1b355a'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India'
  },
  {
    _id: ObjectId('65df3c6f0c2cb59bfb1b355b'),
    name: 'Bala',
    age: 48,
    country: 'India'
  },
  {
    _id: ObjectId('65df3c6f0c2cb59bfb1b355c'),
    name: 'Shankar',
    age: 53,
    country: 'India'
  },
  {
```

Music Director’s


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

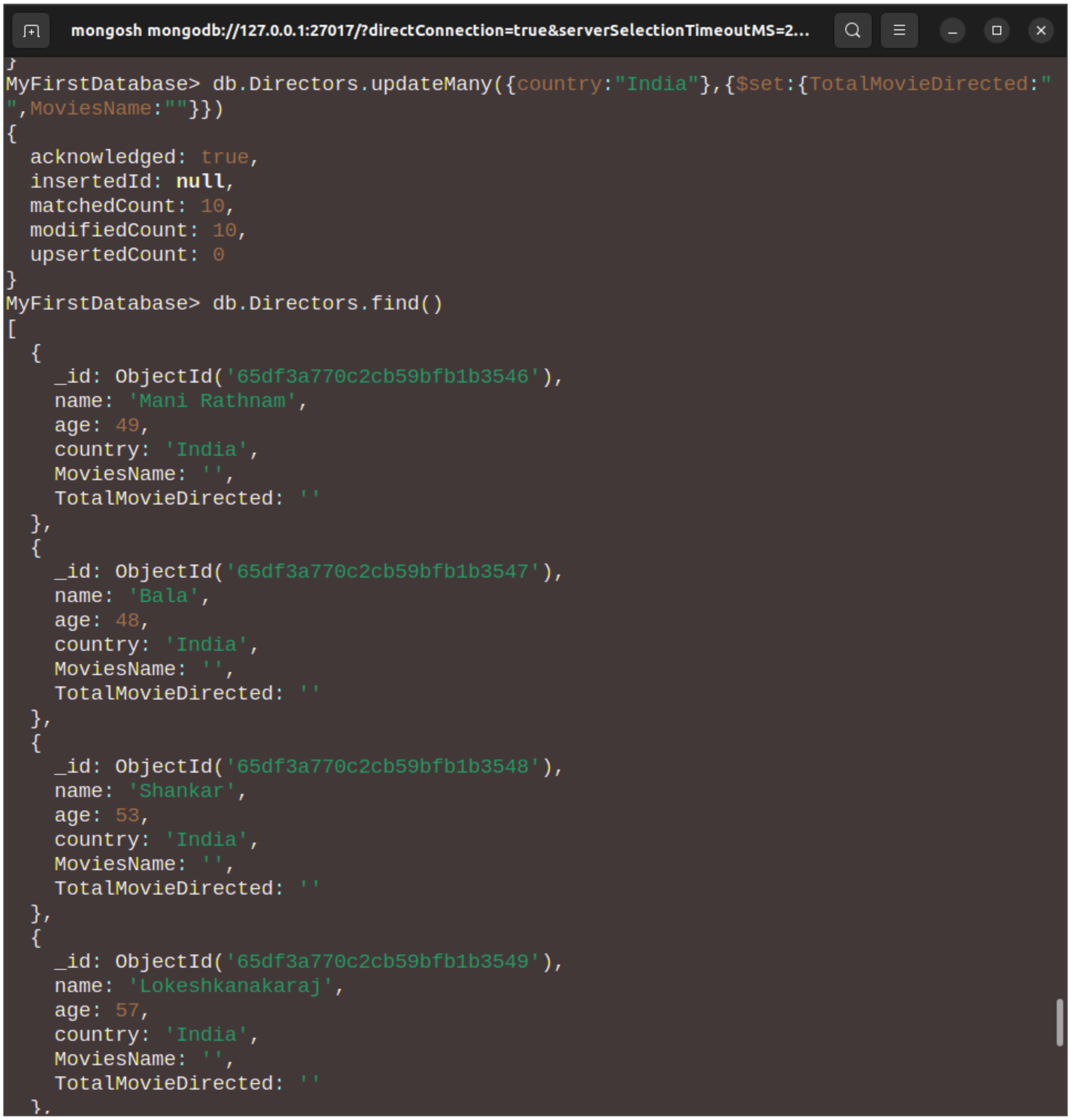
```
  }
]
MyFirstDatabase> db.MusicDirectors.insertMany([{name:"Vidyasagar",age:49,country:"Indi
a"},{name:"A.R.Rahuman",age:48,country:"India"},{name:"Vijay Antony",age:53,country:"I
ndia"},{name:"DSP",age:57,country:"India"},{name:"TR",age:46,country:"India"},{name:"G
angai amaran",age:43,country:"India"},{name:"G.V.Pragash",age:40,country:"India"},{nam
e:"Anirudh",age:40,country:"India"},{name:"U1",age:34,country:"India"},{name:"Hiphop t
hamila Adhi",age:39,country:"India"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df3e480c2cb59bfb1b3564'),
    '1': ObjectId('65df3e480c2cb59bfb1b3565'),
    '2': ObjectId('65df3e480c2cb59bfb1b3566'),
    '3': ObjectId('65df3e480c2cb59bfb1b3567'),
    '4': ObjectId('65df3e480c2cb59bfb1b3568'),
    '5': ObjectId('65df3e480c2cb59bfb1b3569'),
    '6': ObjectId('65df3e480c2cb59bfb1b356a'),
    '7': ObjectId('65df3e480c2cb59bfb1b356b'),
    '8': ObjectId('65df3e480c2cb59bfb1b356c'),
    '9': ObjectId('65df3e480c2cb59bfb1b356d')
  }
}
MyFirstDatabase> db.MusicDirectors.find()
[
  {
    _id: ObjectId('65df3e480c2cb59bfb1b3564'),
    name: 'Vidyasagar',
    age: 49,
    country: 'India'
  },
  {
    _id: ObjectId('65df3e480c2cb59bfb1b3565'),
    name: 'A.R.Rahuman',
    age: 48,
    country: 'India'
  },
  {
    _id: ObjectId('65df3e480c2cb59bfb1b3566'),
    name: 'Vijay Antony',
    age: 53,
    country: 'India'
  },
  {
```

b. Movie’s


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
]
MyFirstDatabase> db.Movies.insertMany([{MovieName:"Teen wolf",Budjet:"$9.5 million",Ac
tors:"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dylan Sprayberry",Producers:"Tyler Posey
,Tyler Hoechlin",MoneyCollecion:"$80 million" },{MovieName:"Teen wolf",Budjet:"$9.5 mi
llion",Actors:"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dylan Sprayberry",Producers:"Ty
ler Posey,Tyler Hoechlin",MoneyCollecion:"$80 million" },{MovieName:"Teen wolf",Budjet
:"$9.5 million",Actors:"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dylan Sprayberry",Prod
ucers:"Tyler Posey,Tyler Hoechlin",MoneyCollecion:"$80 million" },{MovieName:"Teen wol
f",Budjet:"$9.5 million",Actors:"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dylan Spraybe
rry",Producers:"Tyler Posey,Tyler Hoechlin",MoneyCollecion:"$80 million" },{MovieName:
"Teen wolf",Budjet:"$9.5 million",Actors:"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dyla
n Sprayberry",Producers:"Tyler Posey,Tyler Hoechlin",MoneyCollecion:"$80 million" },{M
ovieName:"Teen wolf",Budjet:"$9.5 million",Actors:"Tyler Posey,Tyler Hoechlin,Vince Ma
ttis,Dylan Sprayberry",Producers:"Tyler Posey,Tyler Hoechlin",MoneyCollecion:"$80 mill
ion" },{MovieName:"Teen wolf",Budjet:"$9.5 million",Actors:"Tyler Posey,Tyler Hoechlin
,Vince Mattis,Dylan Sprayberry",Producers:"Tyler Posey,Tyler Hoechlin",MoneyCollecion:
"$80 million" },{MovieName:"Teen wolf",Budjet:"$9.5 million",Actors:"Tyler Posey,Tyler
 Hoechlin,Vince Mattis,Dylan Sprayberry",Producers:"Tyler Posey,Tyler Hoechlin",MoneyC
ollecion:"$80 million" },{MovieName:"Teen wolf",Budjet:"$9.5 million",Actors:"Tyler Po
sey,Tyler Hoechlin,Vince Mattis,Dylan Sprayberry",Producers:"Tyler Posey,Tyler Hoechli
n",MoneyCollecion:"$80 million" },{MovieName:"Teen wolf",Budjet:"$9.5 million",Actors:
"Tyler Posey,Tyler Hoechlin,Vince Mattis,Dylan Sprayberry",Producers:"Tyler Posey,Tyle
r Hoechlin",MoneyCollecion:"$80 million" }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65df3f240c2cb59bfb1b356e'),
    '1': ObjectId('65df3f240c2cb59bfb1b356f'),
    '2': ObjectId('65df3f240c2cb59bfb1b3570'),
    '3': ObjectId('65df3f240c2cb59bfb1b3571'),
    '4': ObjectId('65df3f240c2cb59bfb1b3572'),
    '5': ObjectId('65df3f240c2cb59bfb1b3573'),
    '6': ObjectId('65df3f240c2cb59bfb1b3574'),
    '7': ObjectId('65df3f240c2cb59bfb1b3575'),
    '8': ObjectId('65df3f240c2cb59bfb1b3576'),
    '9': ObjectId('65df3f240c2cb59bfb1b3577')
  }
}
MyFirstDatabase> db.Movies.find()
[
  {
    _id: ObjectId('65df3f240c2cb59bfb1b356e'),
    MovieName: 'Teen wolf',
    Budjet: '$9.5 million',
```

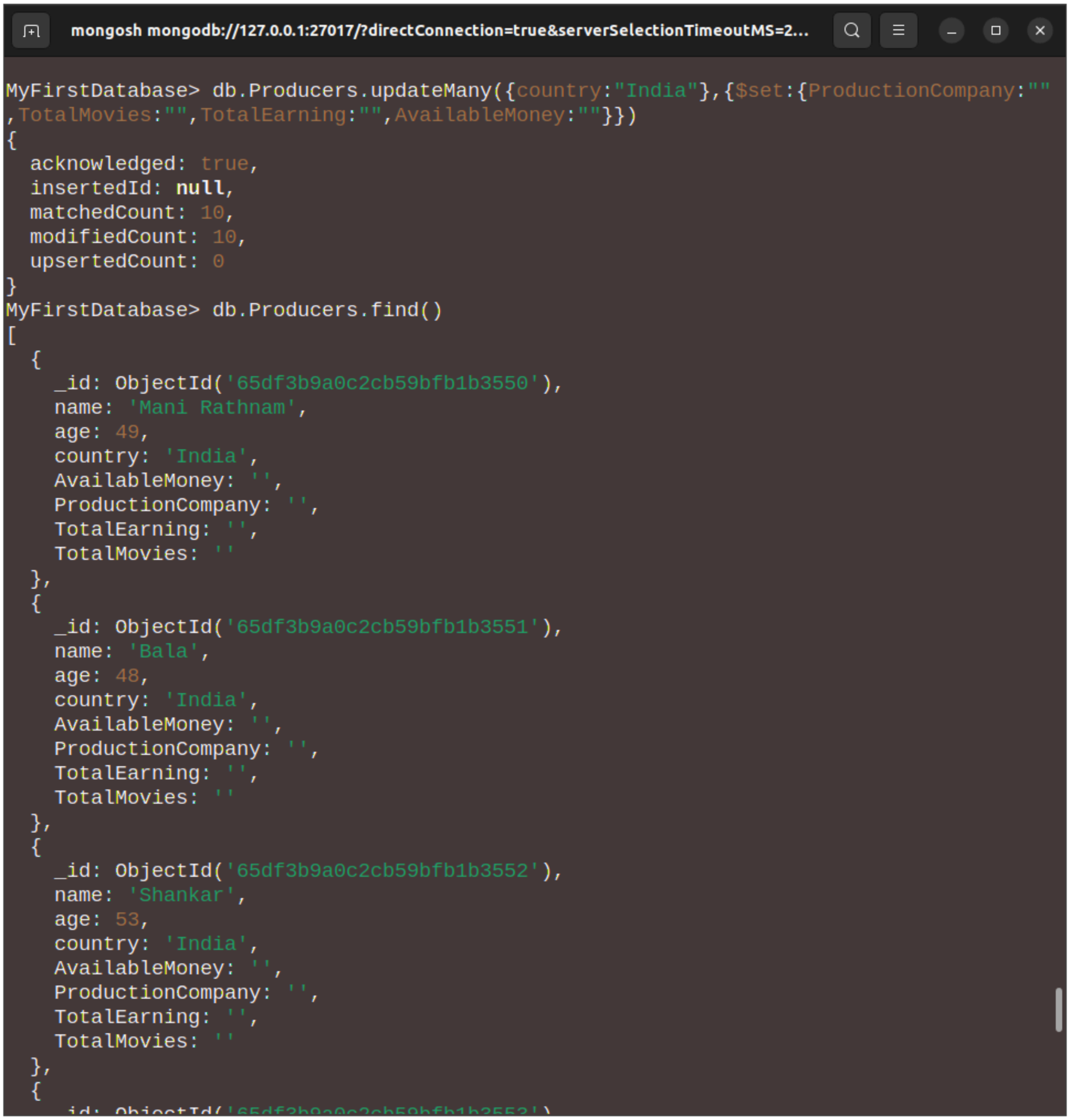
c.



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

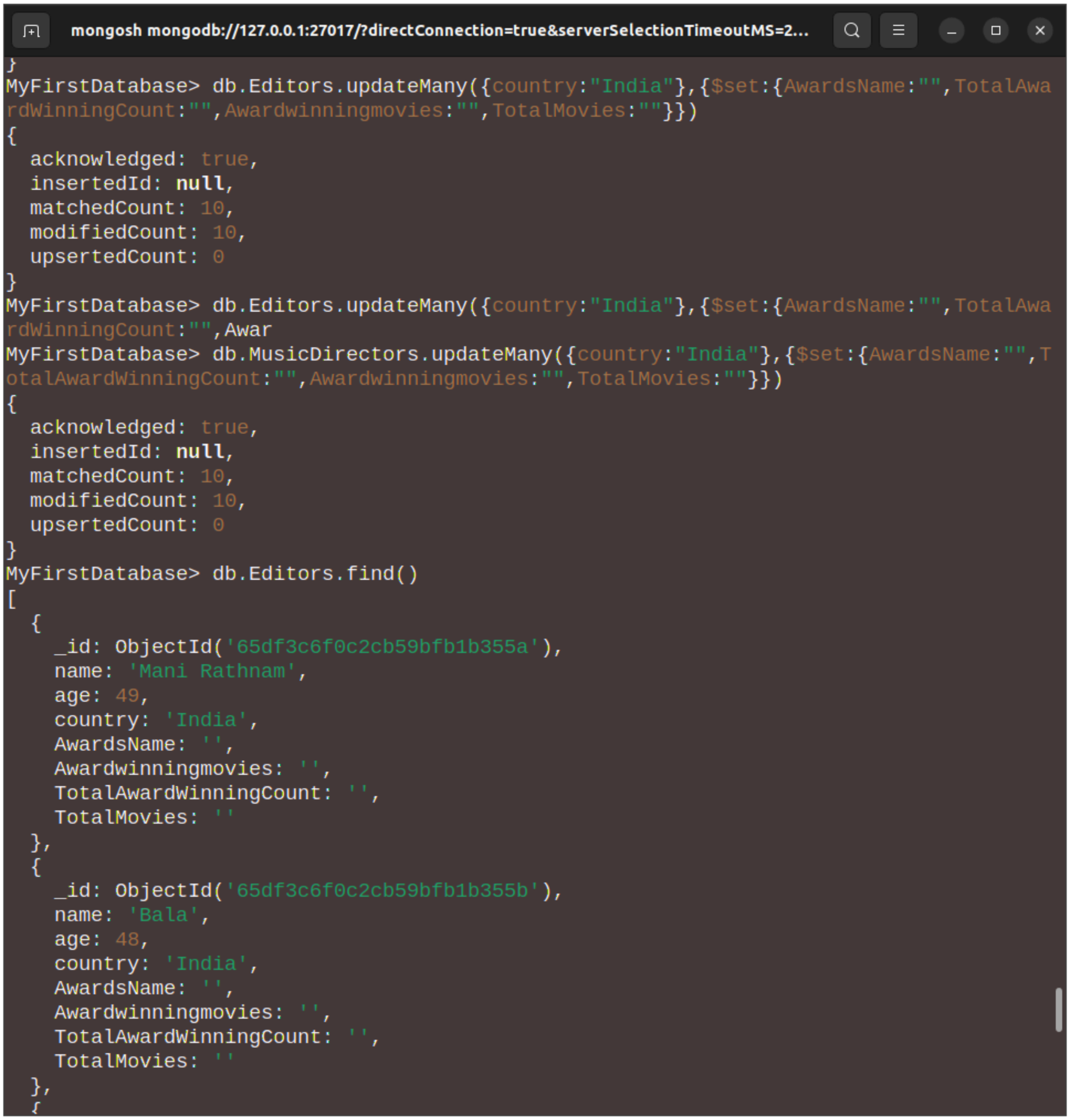
```
}
MyFirstDatabase> db.Directors.updateMany({country:"India"},{$set:{TotalMovieDirected:"
",MoviesName:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 10,
  upsertedCount: 0
}
MyFirstDatabase> db.Directors.find()
[
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3546'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India',
    MoviesName: '',
    TotalMovieDirected: ''
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3547'),
    name: 'Bala',
    age: 48,
    country: 'India',
    MoviesName: '',
    TotalMovieDirected: ''
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3548'),
    name: 'Shankar',
    age: 53,
    country: 'India',
    MoviesName: '',
    TotalMovieDirected: ''
  },
  {
    _id: ObjectId('65df3a770c2cb59bfb1b3549'),
    name: 'Lokeshkanakaraj',
    age: 57,
    country: 'India',
    MoviesName: '',
    TotalMovieDirected: ''
  },
```

d.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

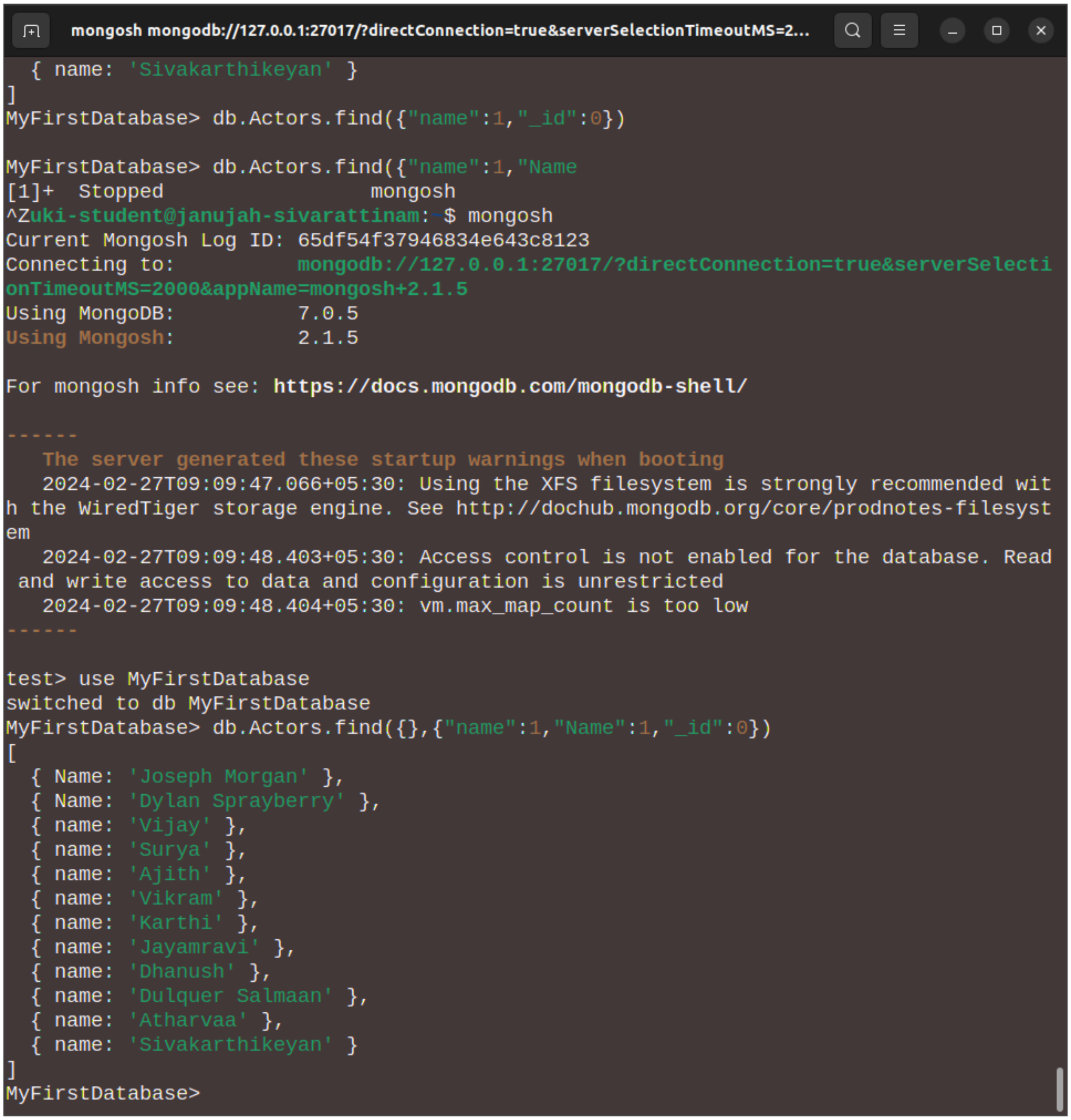
```
MyFirstDatabase> db.Producers.updateMany({country:"India"},{$set:{ProductionCompany:""
,TotalMovies:"",TotalEarning:"",AvailableMoney:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 10,
  upsertedCount: 0
}
MyFirstDatabase> db.Producers.find()
[
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3550'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India',
    AvailableMoney: '',
    ProductionCompany: '',
    TotalEarning: '',
    TotalMovies: ''
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3551'),
    name: 'Bala',
    age: 48,
    country: 'India',
    AvailableMoney: '',
    ProductionCompany: '',
    TotalEarning: '',
    TotalMovies: ''
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3552'),
    name: 'Shankar',
    age: 53,
    country: 'India',
    AvailableMoney: '',
    ProductionCompany: '',
    TotalEarning: '',
    TotalMovies: ''
  },
  {
    _id: ObjectId('65df3b9a0c2cb59bfb1b3553')
```

e.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

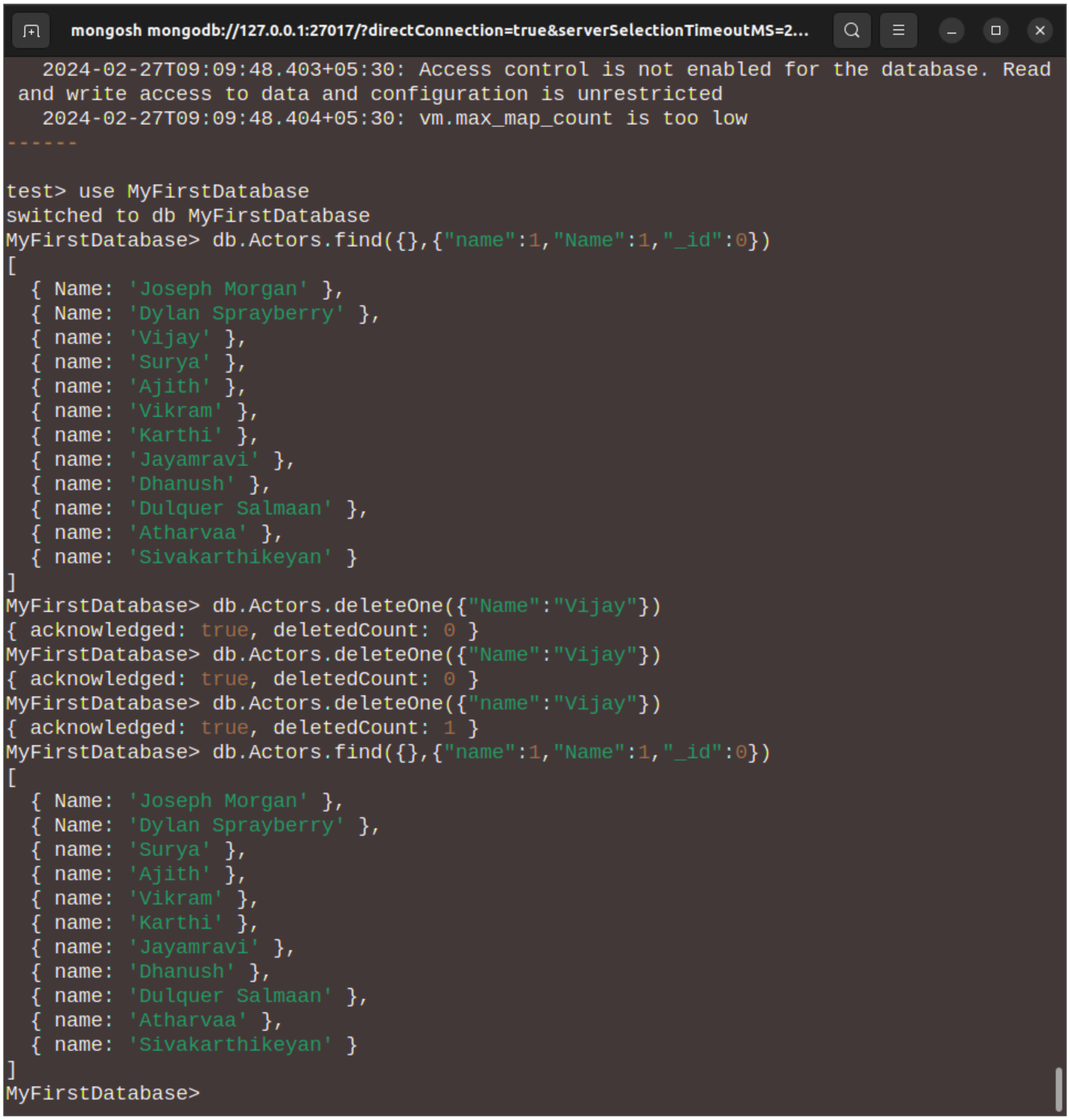
```
}
MyFirstDatabase> db.Editors.updateMany({country:"India"},{$set:{AwardsName:"",TotalAwardWinningCount:"",Awardwinningmovies:"",TotalMovies:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 10,
  upsertedCount: 0
}
MyFirstDatabase> db.Editors.updateMany({country:"India"},{$set:{AwardsName:"",TotalAwardWinningCount:"",Awar
MyFirstDatabase> db.MusicDirectors.updateMany({country:"India"},{$set:{AwardsName:"",TotalAwardWinningCount:"",Awardwinningmovies:"",TotalMovies:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 10,
  upsertedCount: 0
}
MyFirstDatabase> db.Editors.find()
[
  {
    _id: ObjectId('65df3c6f0c2cb59bfb1b355a'),
    name: 'Mani Rathnam',
    age: 49,
    country: 'India',
    AwardsName: '',
    Awardwinningmovies: '',
    TotalAwardWinningCount: '',
    TotalMovies: ''
  },
  {
    _id: ObjectId('65df3c6f0c2cb59bfb1b355b'),
    name: 'Bala',
    age: 48,
    country: 'India',
    AwardsName: '',
    Awardwinningmovies: '',
    TotalAwardWinningCount: '',
    TotalMovies: ''
  },
  {
```

7.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
  { name: 'Sivakarthikeyan' }
]
MyFirstDatabase> db.Actors.find({"name":1,"_id":0})

MyFirstDatabase> db.Actors.find({"name":1,"Name
[1]+  Stopped                 mongosh
^Zuki-student@janujah-sivarattinam:~$ mongosh
Current Mongosh Log ID: 65df54f37946834e643c8123
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelecti
onTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:          7.0.5
Using Mongosh:          2.1.5

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

------
   The server generated these startup warnings when booting
   2024-02-27T09:09:47.066+05:30: Using the XFS filesystem is strongly recommended wit
h the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesyst
em
   2024-02-27T09:09:48.403+05:30: Access control is not enabled for the database. Read
 and write access to data and configuration is unrestricted
   2024-02-27T09:09:48.404+05:30: vm.max_map_count is too low
------

test> use MyFirstDatabase
switched to db MyFirstDatabase
MyFirstDatabase> db.Actors.find({},{"name":1,"Name":1,"_id":0})
[
  { Name: 'Joseph Morgan' },
  { Name: 'Dylan Sprayberry' },
  { name: 'Vijay' },
  { name: 'Surya' },
  { name: 'Ajith' },
  { name: 'Vikram' },
  { name: 'Karthi' },
  { name: 'Jayamravi' },
  { name: 'Dhanush' },
  { name: 'Dulquer Salmaan' },
  { name: 'Atharvaa' },
  { name: 'Sivakarthikeyan' }
]
MyFirstDatabase>
```

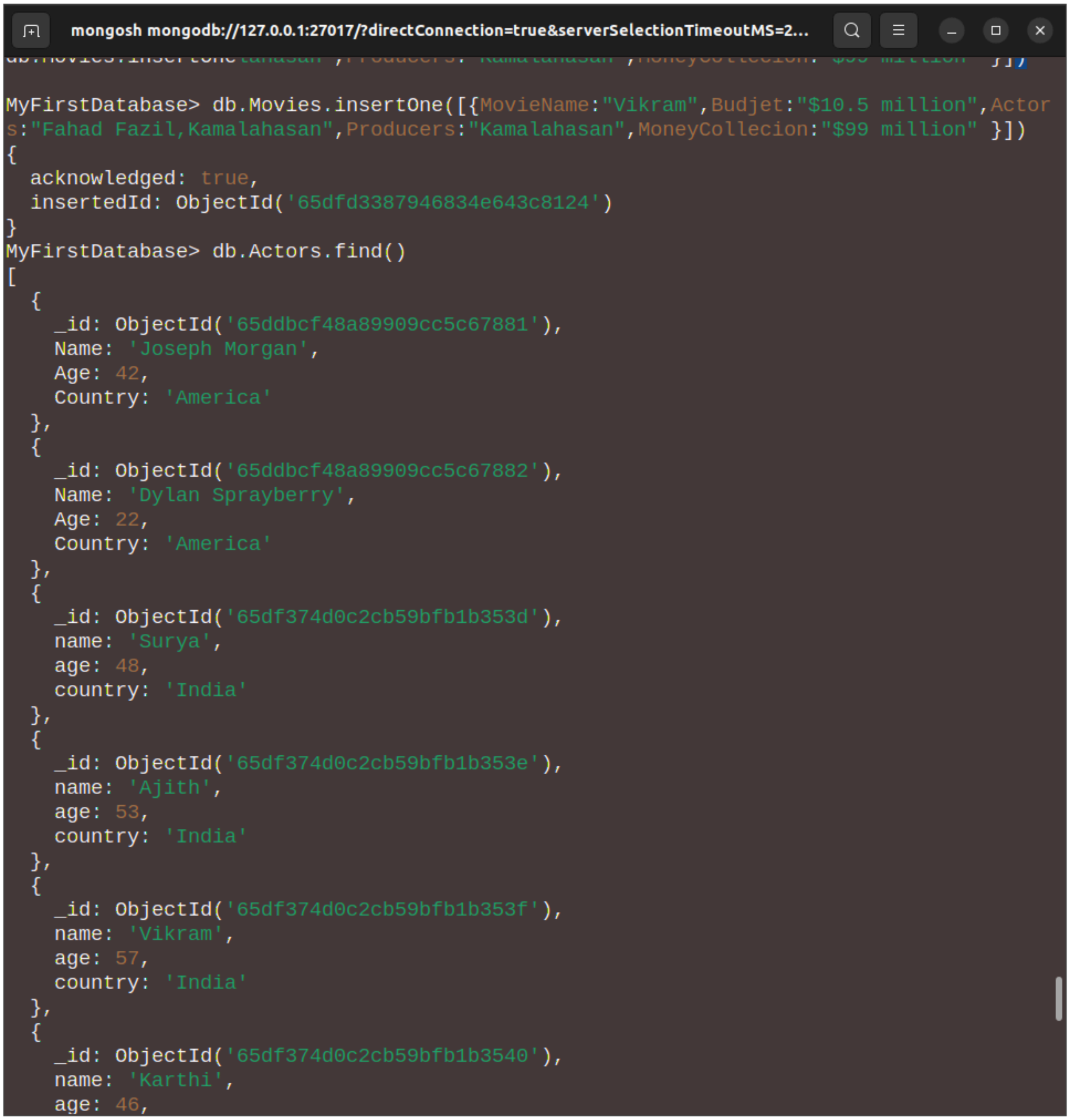
8.



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
   2024-02-27T09:09:48.403+05:30: Access control is not enabled for the database. Read
 and write access to data and configuration is unrestricted
   2024-02-27T09:09:48.404+05:30: vm.max_map_count is too low
------

test> use MyFirstDatabase
switched to db MyFirstDatabase
MyFirstDatabase> db.Actors.find({},{"name":1,"Name":1,"_id":0})
[
  { Name: 'Joseph Morgan' },
  { Name: 'Dylan Sprayberry' },
  { name: 'Vijay' },
  { name: 'Surya' },
  { name: 'Ajith' },
  { name: 'Vikram' },
  { name: 'Karthi' },
  { name: 'Jayamravi' },
  { name: 'Dhanush' },
  { name: 'Dulquer Salmaan' },
  { name: 'Atharvaa' },
  { name: 'Sivakarthikeyan' }
]
MyFirstDatabase> db.Actors.deleteOne({"Name":"Vijay"})
{ acknowledged: true, deletedCount: 0 }
MyFirstDatabase> db.Actors.deleteOne({"Name":"Vijay"})
{ acknowledged: true, deletedCount: 0 }
MyFirstDatabase> db.Actors.deleteOne({"name":"Vijay"})
{ acknowledged: true, deletedCount: 1 }
MyFirstDatabase> db.Actors.find({},{"name":1,"Name":1,"_id":0})
[
  { Name: 'Joseph Morgan' },
  { Name: 'Dylan Sprayberry' },
  { name: 'Surya' },
  { name: 'Ajith' },
  { name: 'Vikram' },
  { name: 'Karthi' },
  { name: 'Jayamravi' },
  { name: 'Dhanush' },
  { name: 'Dulquer Salmaan' },
  { name: 'Atharvaa' },
  { name: 'Sivakarthikeyan' }
]
MyFirstDatabase>
```

9.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
db.Movies.insertOnelahasan",Producers:"Kamalahasan",MoneyCollecion:"$99 million" }])

MyFirstDatabase> db.Movies.insertOne([{MovieName:"Vikram",Budjet:"$10.5 million",Actor
s:"Fahad Fazil,Kamalahasan",Producers:"Kamalahasan",MoneyCollecion:"$99 million" }])
{
  acknowledged: true,
  insertedId: ObjectId('65dfd3387946834e643c8124')
}
MyFirstDatabase> db.Actors.find()
[
  {
    _id: ObjectId('65ddbcf48a89909cc5c67881'),
    Name: 'Joseph Morgan',
    Age: 42,
    Country: 'America'
  },
  {
    _id: ObjectId('65ddbcf48a89909cc5c67882'),
    Name: 'Dylan Sprayberry',
    Age: 22,
    Country: 'America'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b353d'),
    name: 'Surya',
    age: 48,
    country: 'India'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b353e'),
    name: 'Ajith',
    age: 53,
    country: 'India'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b353f'),
    name: 'Vikram',
    age: 57,
    country: 'India'
  },
  {
    _id: ObjectId('65df374d0c2cb59bfb1b3540'),
    name: 'Karthi',
    age: 46,
```

Answering Questions from Data

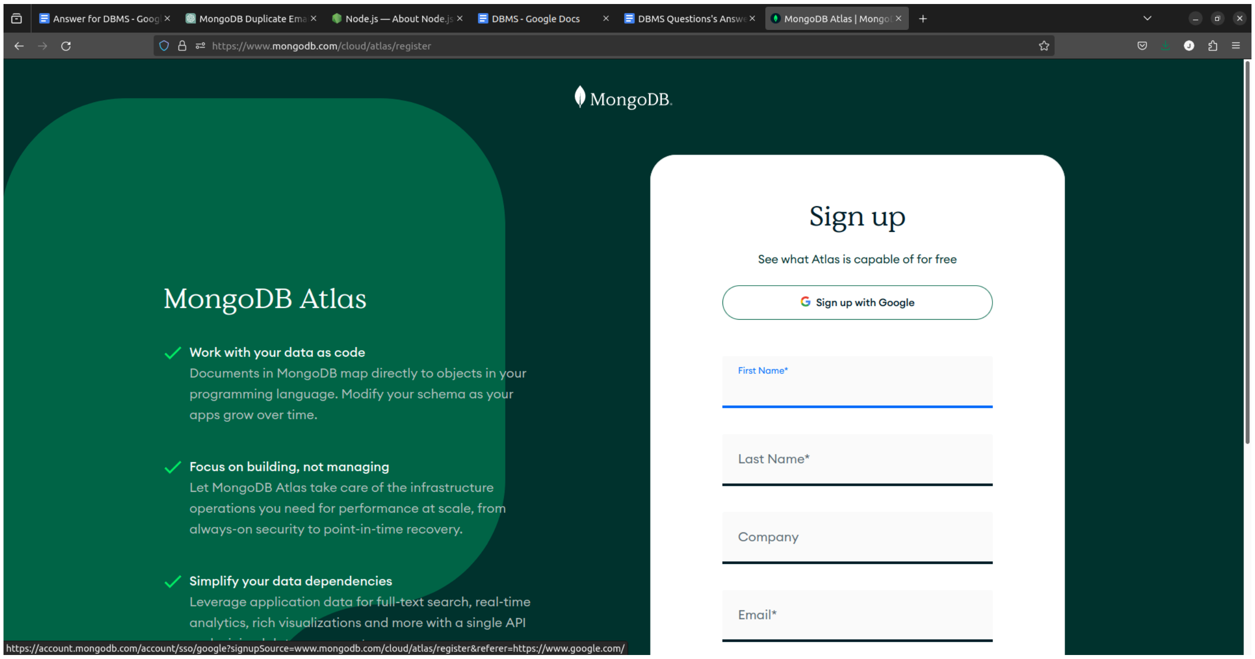
1. inventory
2. An **insertMany** used to add more than one data into a collection with unique Id.
3.

MongoDB Atlas

1. The **runtime environment** is the environment in which a program or application is executed. It's the hardware and software infrastructure that supports the running of a particular codebase in real time.

2. It is a open source. It is a JavaScript run time environment.

3.



MongoDB Atlas

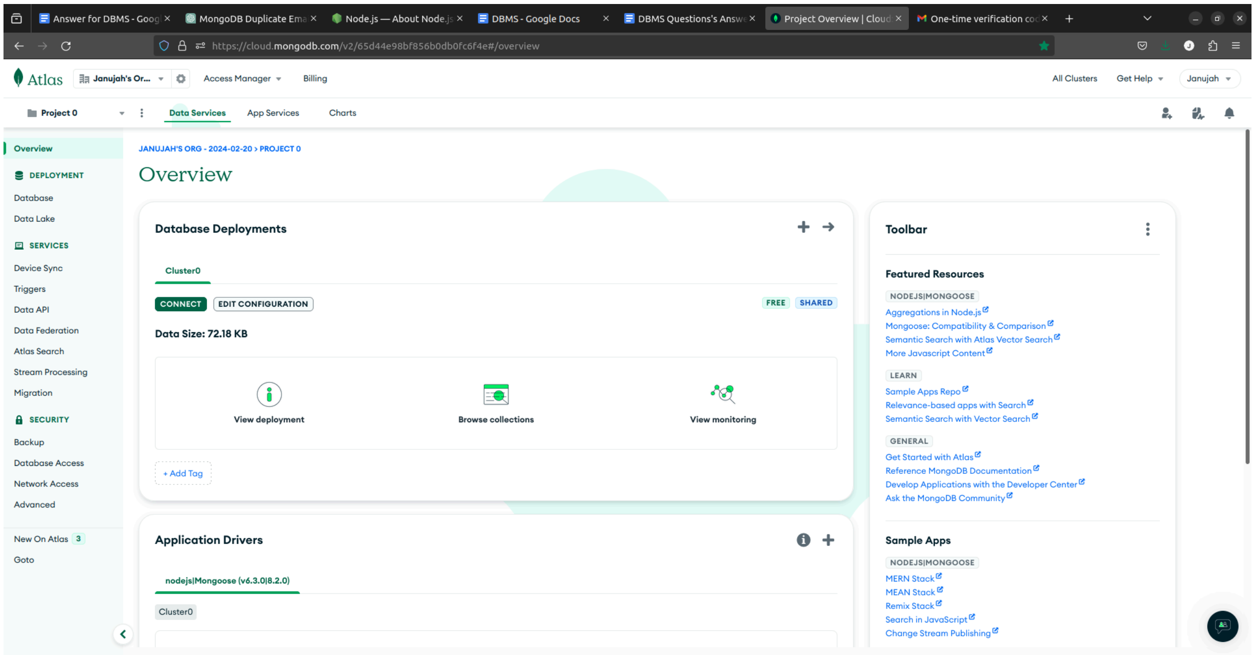
✓ **Work with your data as code**
Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.

✓ **Focus on building, not managing**
Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.

✓ **Simplify your data dependencies**
Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API

Sign up

See what Atlas is capable of for free



4. Log in your MongoDB atlas.
   Create a cluster.
   Change the IP address.
   Download the Node.js Driver - npm install mongodb
   Go to the MongoDB Atlas.
   Navigate to the database.
   Click the connect.
   Navigate to the Drivers.
   Get the command - npm install mongodb. and insert in your project folder.
   Coppy the URL and insert it in your JS file.
   URL=mongodb+srv://janujahsivarattinam:<password>@cluster0.lcosh5o.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0;
   Connect the mongoose
   mongoose.connect(URL);