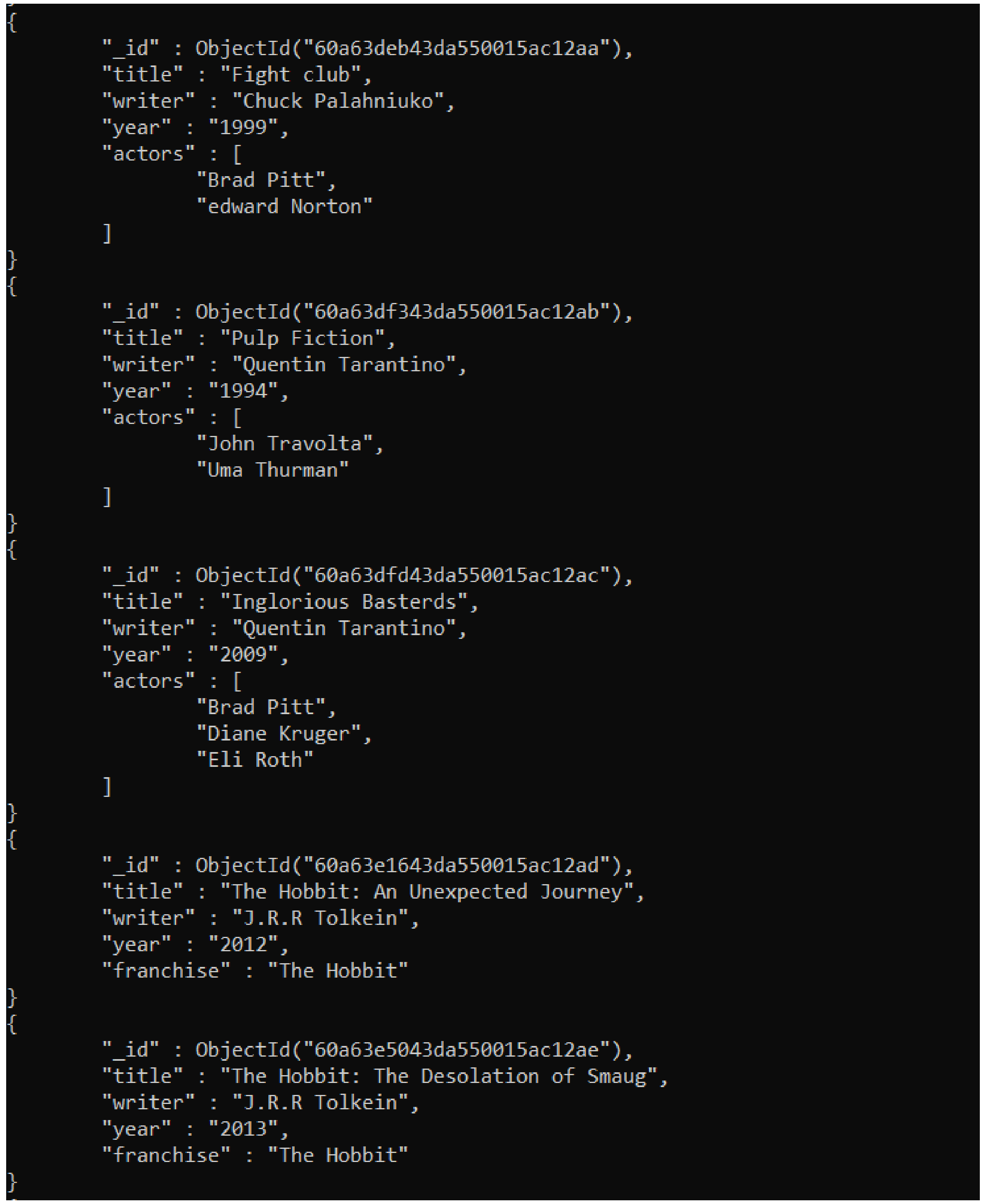
MongoDB Assignment1

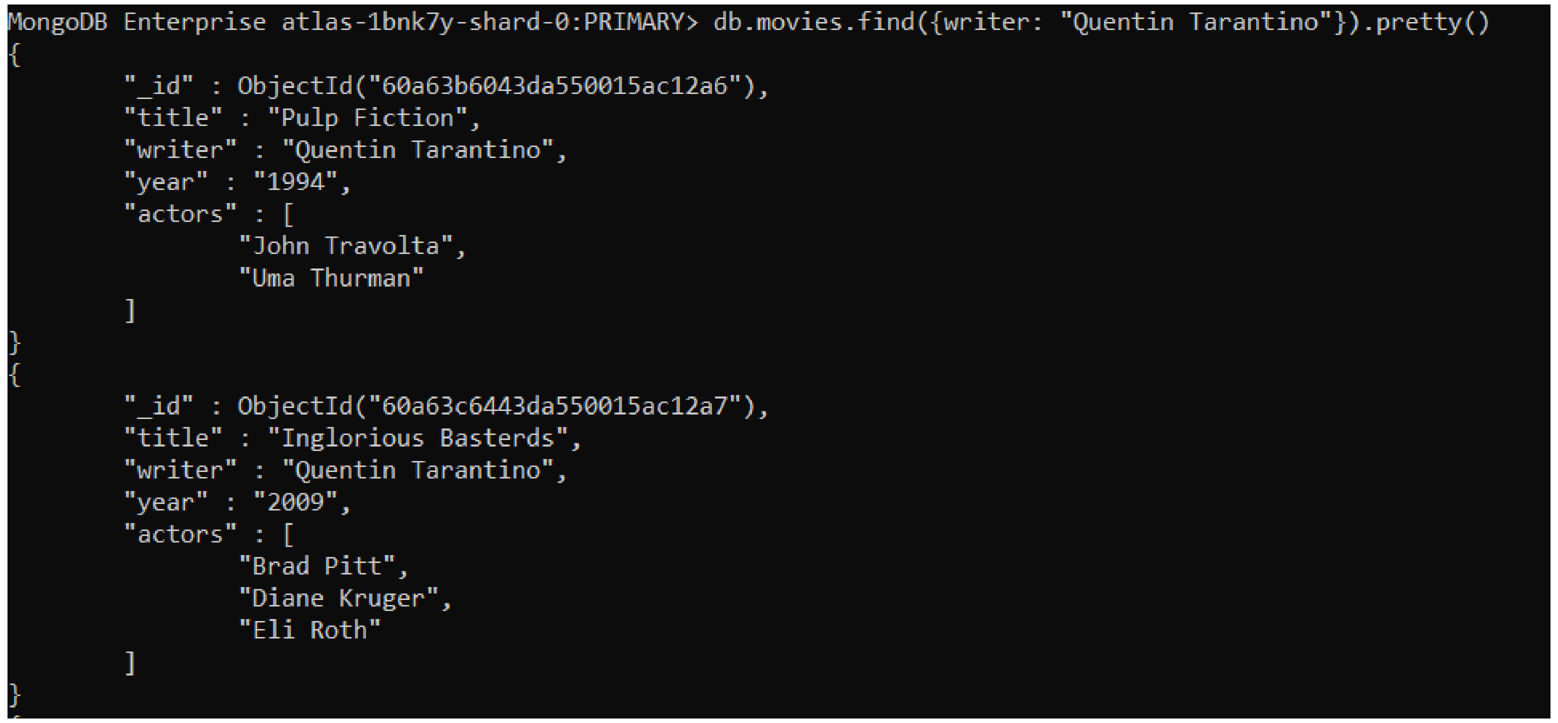
A) Query the Movies collection to

1.Get all document



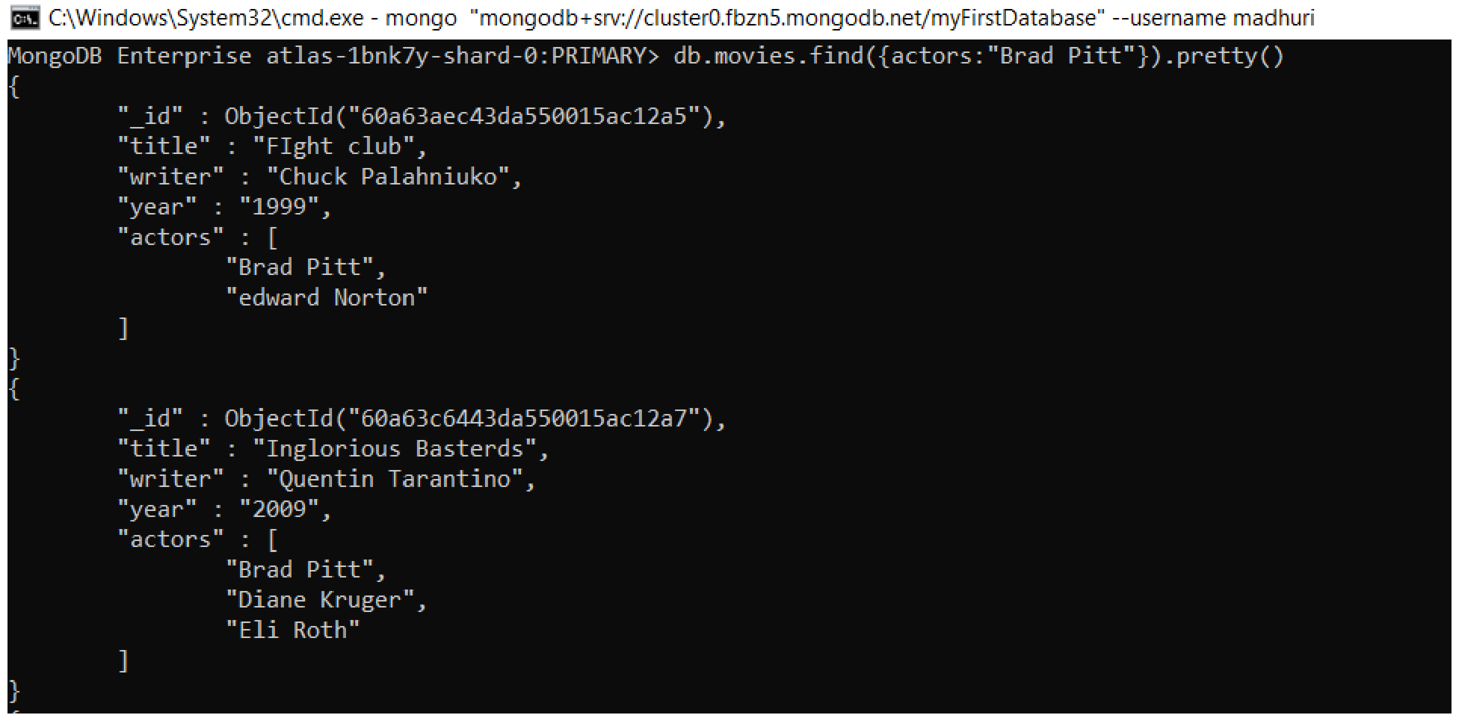
```
{
        "_id" : ObjectId("60a63deb43da550015ac12aa"),
        "title" : "Fight club",
        "writer" : "Chuck Palahniuko",
        "year" : "1999",
        "actors" : [
                "Brad Pitt",
                "edward Norton"
        ]
}
{
        "_id" : ObjectId("60a63df343da550015ac12ab"),
        "title" : "Pulp Fiction",
        "writer" : "Quentin Tarantino",
        "year" : "1994",
        "actors" : [
                "John Travolta",
                "Uma Thurman"
        ]
}
{
        "_id" : ObjectId("60a63dfd43da550015ac12ac"),
        "title" : "Inglorious Basterds",
        "writer" : "Quentin Tarantino",
        "year" : "2009",
        "actors" : [
                "Brad Pitt",
                "Diane Kruger",
                "Eli Roth"
        ]
}
{
        "_id" : ObjectId("60a63e1643da550015ac12ad"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit"
}
{
        "_id" : ObjectId("60a63e5043da550015ac12ae"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "year" : "2013",
        "franchise" : "The Hobbit"
}
```

2. Get all documents with writer set to "Quentin Tarantino”



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({writer: "Quentin Tarantino"}).pretty()
{
        "_id" : ObjectId("60a63b6043da550015ac12a6"),
        "title" : "Pulp Fiction",
        "writer" : "Quentin Tarantino",
        "year" : "1994",
        "actors" : [
                "John Travolta",
                "Uma Thurman"
        ]
}
{
        "_id" : ObjectId("60a63c6443da550015ac12a7"),
        "title" : "Inglorious Basterds",
        "writer" : "Quentin Tarantino",
        "year" : "2009",
        "actors" : [
                "Brad Pitt",
                "Diane Kruger",
                "Eli Roth"
        ]
}
```

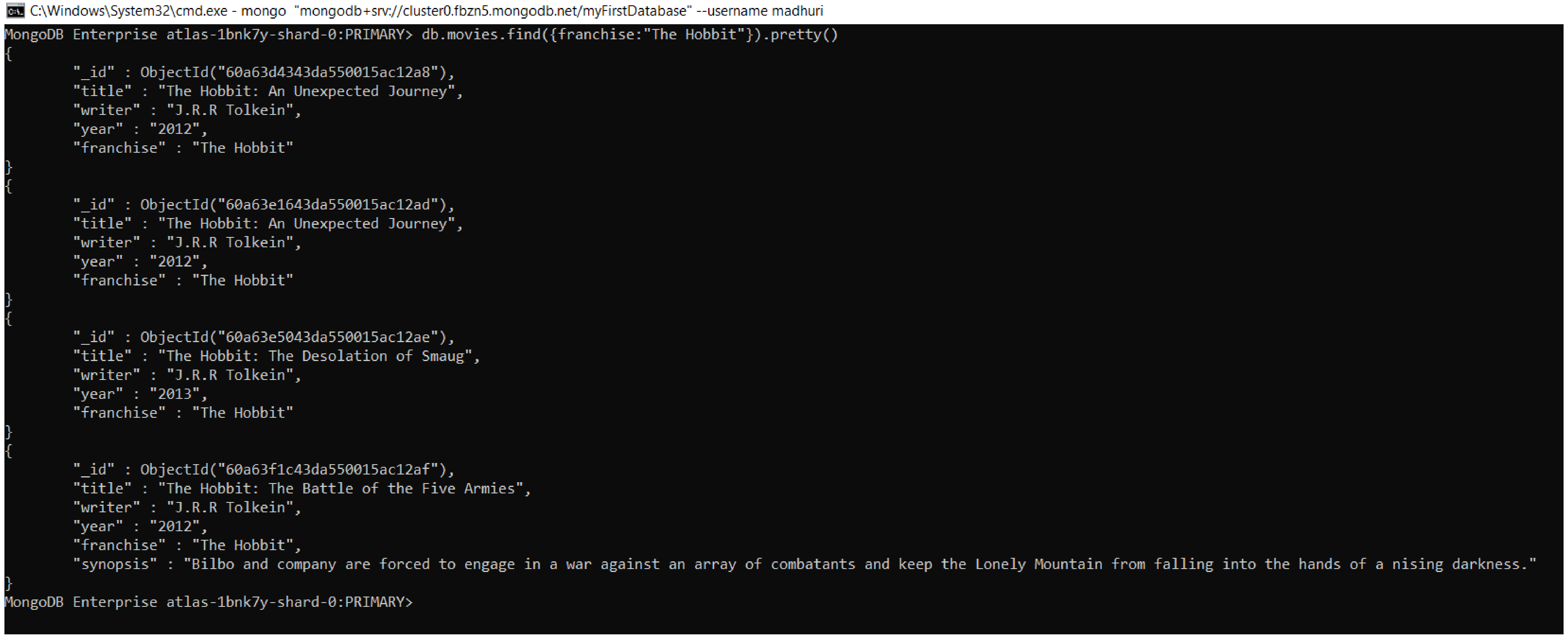
3. Get all documents where actors include "Brad Pitt”



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({actors:"Brad Pitt"}).pretty()
{
        "_id" : ObjectId("60a63aec43da550015ac12a5"),
        "title" : "FIght club",
        "writer" : "Chuck Palahniuko",
        "year" : "1999",
        "actors" : [
                "Brad Pitt",
                "edward Norton"
        ]
}
{
        "_id" : ObjectId("60a63c6443da550015ac12a7"),
        "title" : "Inglorious Basterds",
        "writer" : "Quentin Tarantino",
        "year" : "2009",
        "actors" : [
                "Brad Pitt",
                "Diane Kruger",
                "Eli Roth"
        ]
}
```

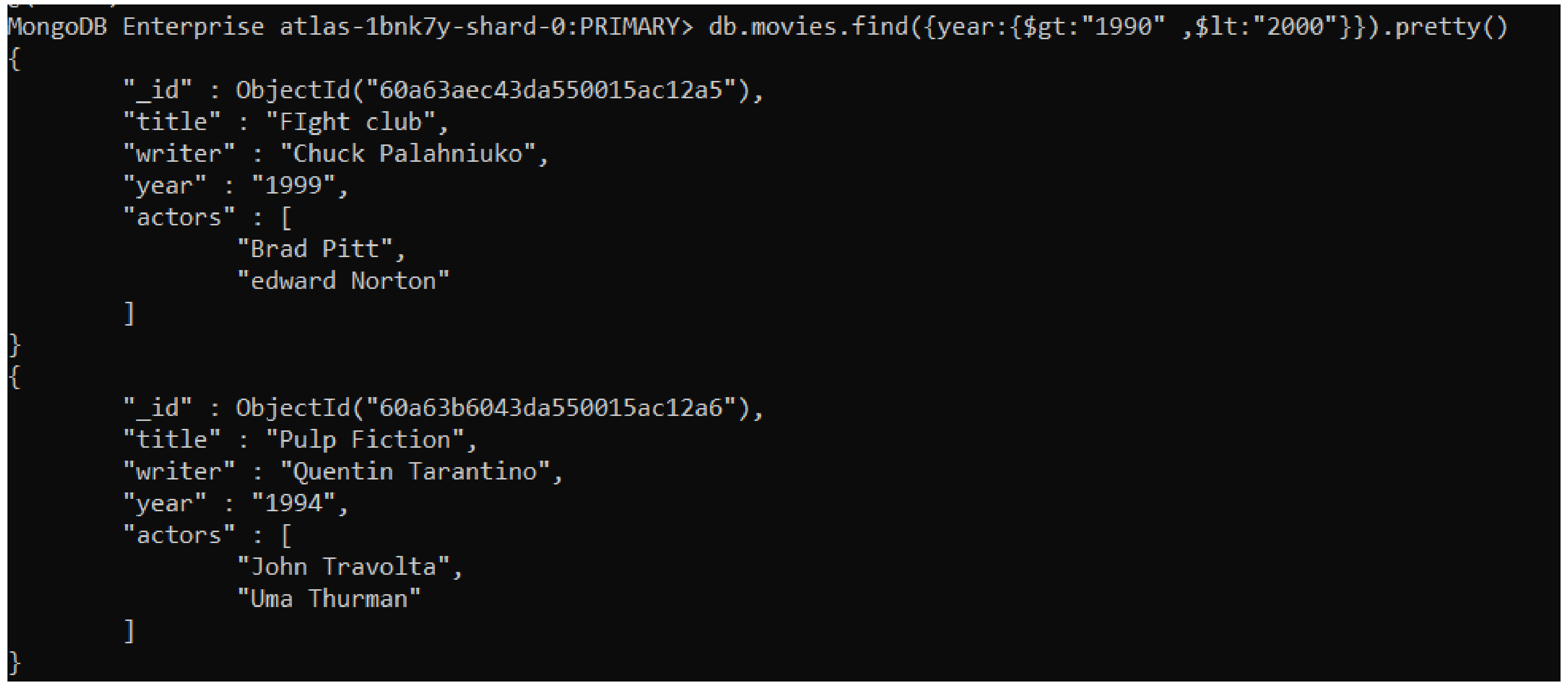
4. Get all documents with franchise set to "The Hobbit”



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

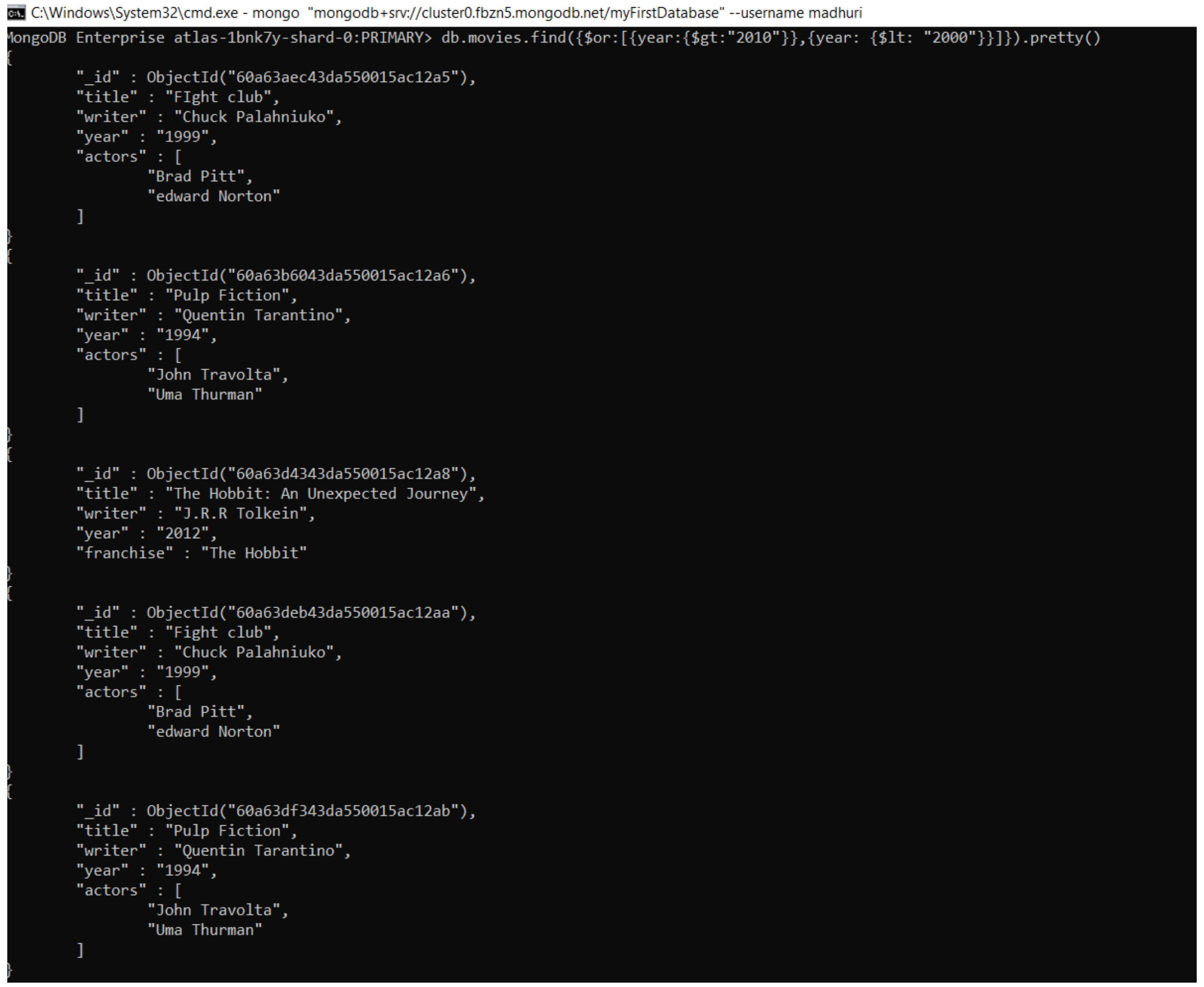
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({franchise:"The Hobbit"}).pretty()
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit"
}
{
        "_id" : ObjectId("60a63e1643da550015ac12ad"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit"
}
{
        "_id" : ObjectId("60a63e5043da550015ac12ae"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "year" : "2013",
        "franchise" : "The Hobbit"
}
{
        "_id" : ObjectId("60a63f1c43da550015ac12af"),
        "title" : "The Hobbit: The Battle of the Five Armies",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "Bilbo and company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a nising darkness."
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

5. Get all movies released in the 90s



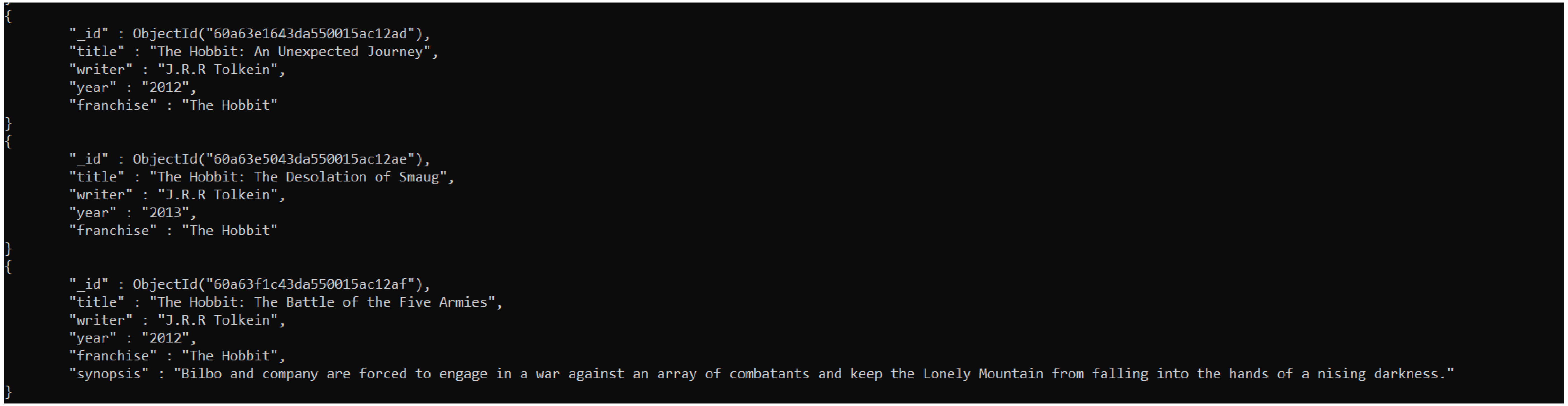
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({year:{$gt:"1990" ,$lt:"2000"}}).pretty()
{
        "_id" : ObjectId("60a63aec43da550015ac12a5"),
        "title" : "FIght club",
        "writer" : "Chuck Palahniuko",
        "year" : "1999",
        "actors" : [
                "Brad Pitt",
                "edward Norton"
        ]
}
{
        "_id" : ObjectId("60a63b6043da550015ac12a6"),
        "title" : "Pulp Fiction",
        "writer" : "Quentin Tarantino",
        "year" : "1994",
        "actors" : [
                "John Travolta",
                "Uma Thurman"
        ]
}
```

6. Get all movies released before the year 2000 or after 2010



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

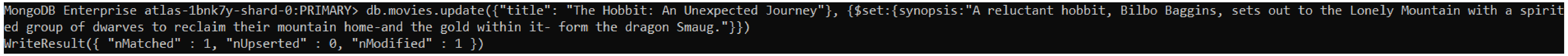
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt: "2000"}}]}).pretty()
{
        "_id" : ObjectId("60a63aec43da550015ac12a5"),
        "title" : "FIght club",
        "writer" : "Chuck Palahniuko",
        "year" : "1999",
        "actors" : [
                "Brad Pitt",
                "edward Norton"
        ]
}
{
        "_id" : ObjectId("60a63b6043da550015ac12a6"),
        "title" : "Pulp Fiction",
        "writer" : "Quentin Tarantino",
        "year" : "1994",
        "actors" : [
                "John Travolta",
                "Uma Thurman"
        ]
}
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit"
}
{
        "_id" : ObjectId("60a63deb43da550015ac12aa"),
        "title" : "Fight club",
        "writer" : "Chuck Palahniuko",
        "year" : "1999",
        "actors" : [
                "Brad Pitt",
                "edward Norton"
        ]
}
{
        "_id" : ObjectId("60a63df343da550015ac12ab"),
        "title" : "Pulp Fiction",
        "writer" : "Quentin Tarantino",
        "year" : "1994",
        "actors" : [
                "John Travolta",
                "Uma Thurman"
        ]
}
```



```
{

        "_id" : ObjectId("60a63e1643da550015ac12ad"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit"
}
{

        "_id" : ObjectId("60a63e5043da550015ac12ae"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "year" : "2013",
        "franchise" : "The Hobbit"
}
{

        "_id" : ObjectId("60a63f1c43da550015ac12af"),
        "title" : "The Hobbit: The Battle of the Five Armies",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "Bilbo and company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a nising darkness."
}
```

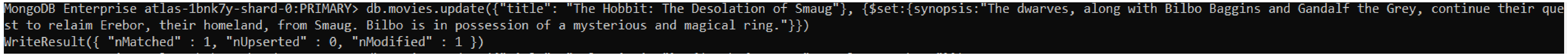
Update Documents

7. Add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."



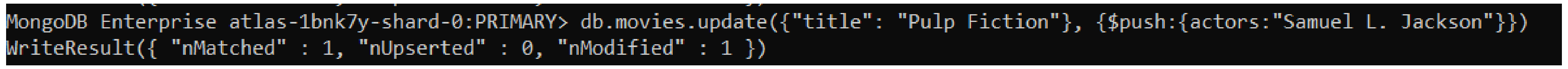
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.update({"title": "The Hobbit: An Unexpected Journey"}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirit
ed group of dwarves to reclaim their mountain home-and the gold within it- form the dragon Smaug."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

8. Add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.update({"title": "The Hobbit: The Desolation of Smaug"}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their que
st to relaim Erebor, their homeland, from Smaug. Bilbo is in possession of a mysterious and magical ring."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

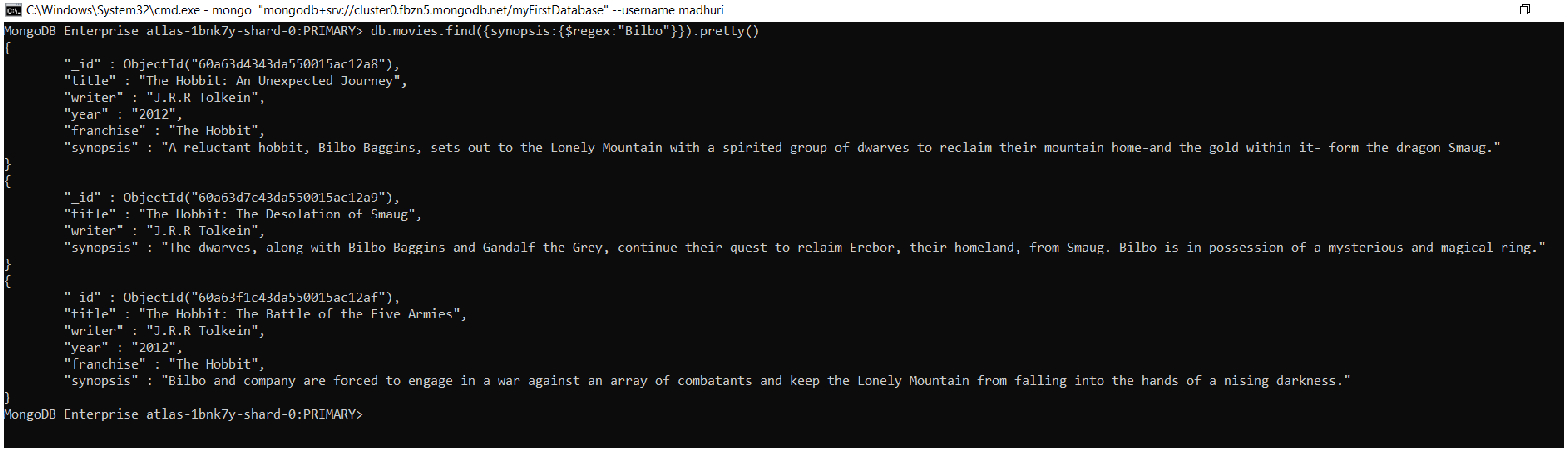
9. Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.update({"title": "Pulp Fiction"}, {$push:{actors:"Samuel L. Jackson"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

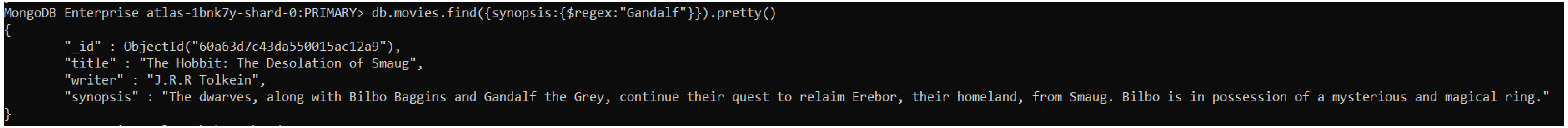
B) Text Search

10. Find all movies that have a synopsis that contains the word "Bilbo"


C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

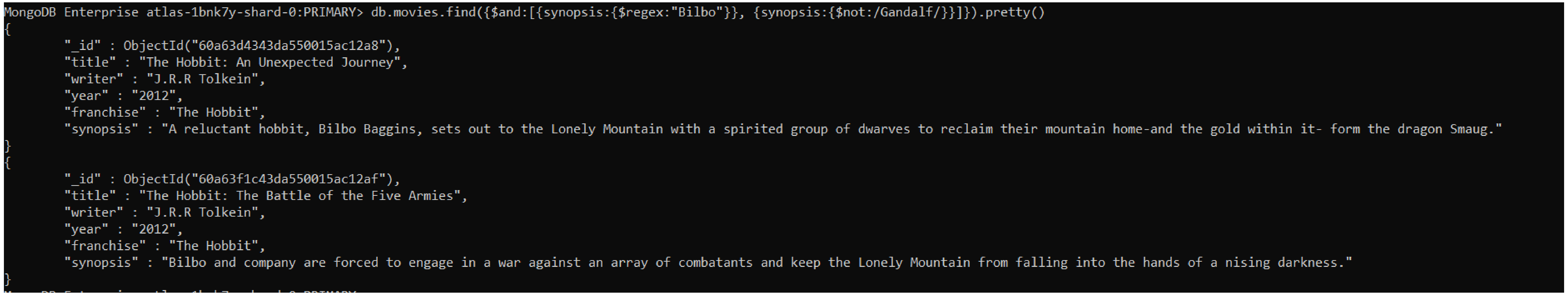
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({synopsis:{$regex:"Bilbo"}}).pretty()
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home-and the gold within it- form the dragon Smaug."
}
{
        "_id" : ObjectId("60a63d7c43da550015ac12a9"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to relaim Erebor, their homeland, from Smaug. Bilbo is in possession of a mysterious and magical ring."
}
{
        "_id" : ObjectId("60a63f1c43da550015ac12af"),
        "title" : "The Hobbit: The Battle of the Five Armies",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "Bilbo and company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a nising darkness."
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

11. Find all movies that have a synopsis that contains the word "Gandalf”



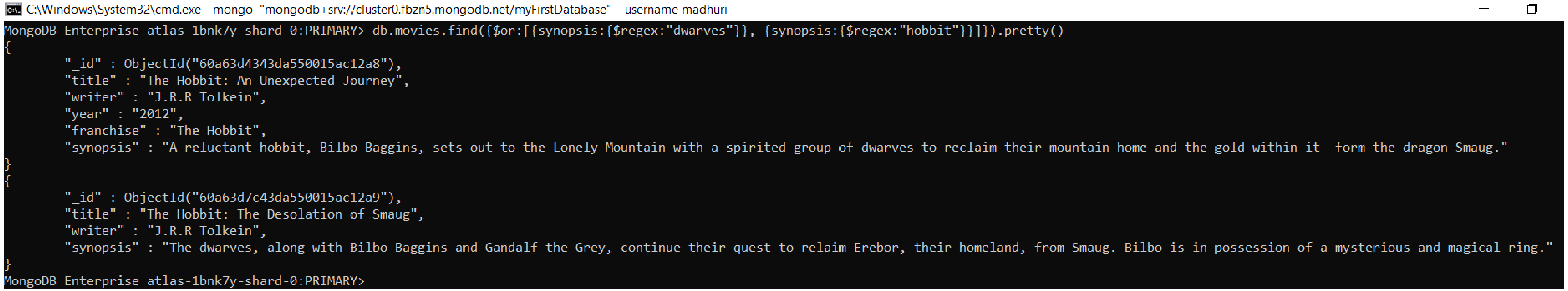
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({synopsis:{$regex:"Gandalf"}}).pretty()
{
        "_id" : ObjectId("60a63d7c43da550015ac12a9"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to relaim Erebor, their homeland, from Smaug. Bilbo is in possession of a mysterious and magical ring."
}
```

12. Find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf".



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]}).pretty()
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home-and the gold within it- form the dragon Smaug."
}
{
        "_id" : ObjectId("60a63f1c43da550015ac12af"),
        "title" : "The Hobbit: The Battle of the Five Armies",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "Bilbo and company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a nising darkness."
}
```

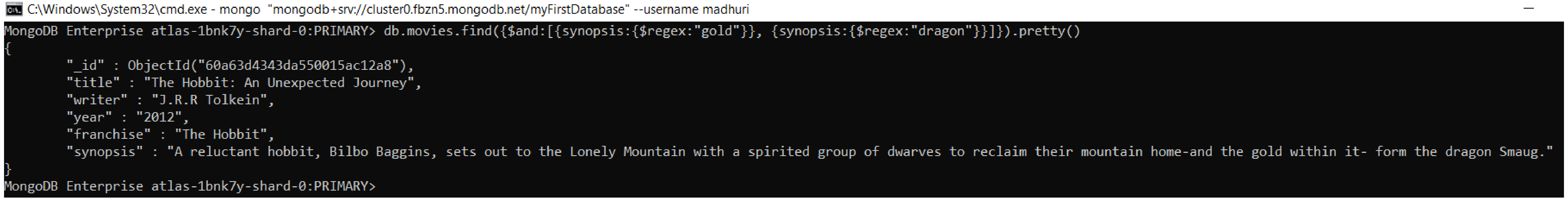
13. Find all movies that have a synopsis that contains the word "dwarves" or "hobbit".



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]}).pretty()
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home-and the gold within it- form the dragon Smaug."
}
{
        "_id" : ObjectId("60a63d7c43da550015ac12a9"),
        "title" : "The Hobbit: The Desolation of Smaug",
        "writer" : "J.R.R Tolkein",
        "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to relaim Erebor, their homeland, from Smaug. Bilbo is in possession of a mysterious and magical ring."
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

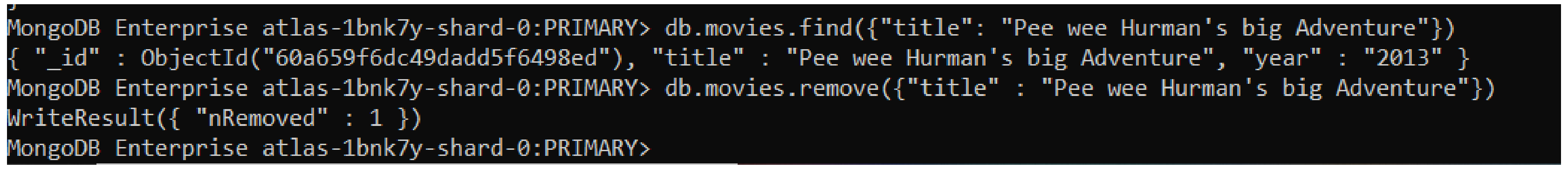
14. Find all movies that have a synopsis that contains the word "gold" and "dragon".


C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]}).pretty()
{
        "_id" : ObjectId("60a63d4343da550015ac12a8"),
        "title" : "The Hobbit: An Unexpected Journey",
        "writer" : "J.R.R Tolkein",
        "year" : "2012",
        "franchise" : "The Hobbit",
        "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home-and the gold within it- form the dragon Smaug."
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

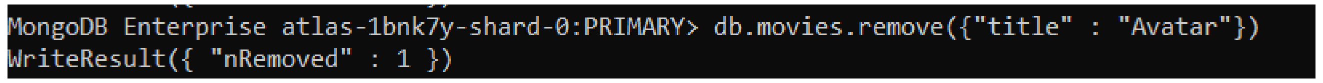
C) Delete Documents

15. Delete the movie "Pee Wee Herman's Big Adventure"



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.find({"title": "Pee wee Hurman's big Adventure"})
{ "_id" : ObjectId("60a659f6dc49dadd5f6498ed"), "title" : "Pee wee Hurman's big Adventure", "year" : "2013" }
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.remove({"title" : "Pee wee Hurman's big Adventure"})
WriteResult({ "nRemoved" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

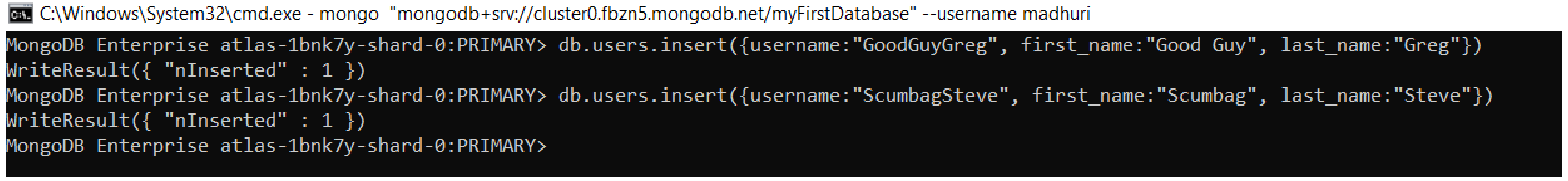
16. Delete the movie “Avatar"



```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.movies.remove({"title" : "Avatar"})
WriteResult({ "nRemoved" : 1 })
```

D) Relationships

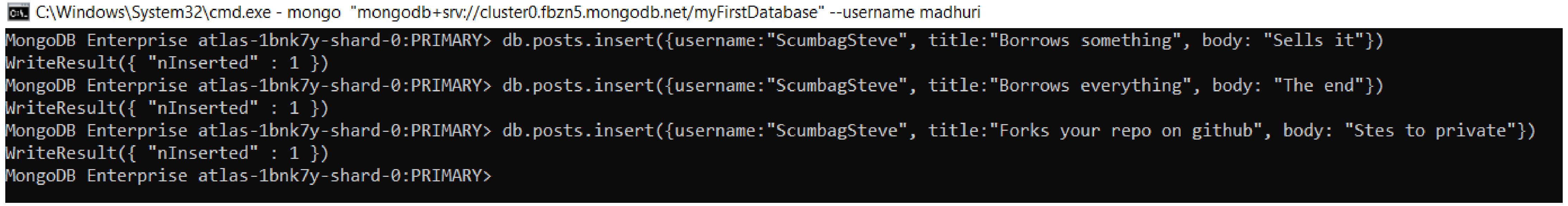
17. Insert the following documents into a **users** collection.



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

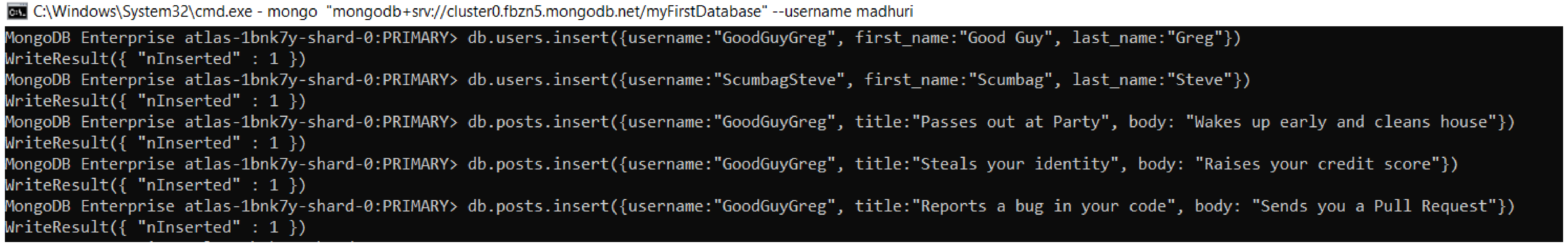
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.users.insert({username:"GoodGuyGreg", first_name:"Good Guy", last_name:"Greg"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.users.insert({username:"ScumbagSteve", first_name:"Scumbag", last_name:"Steve"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

18. Insert the following documents into a **posts** collection.



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

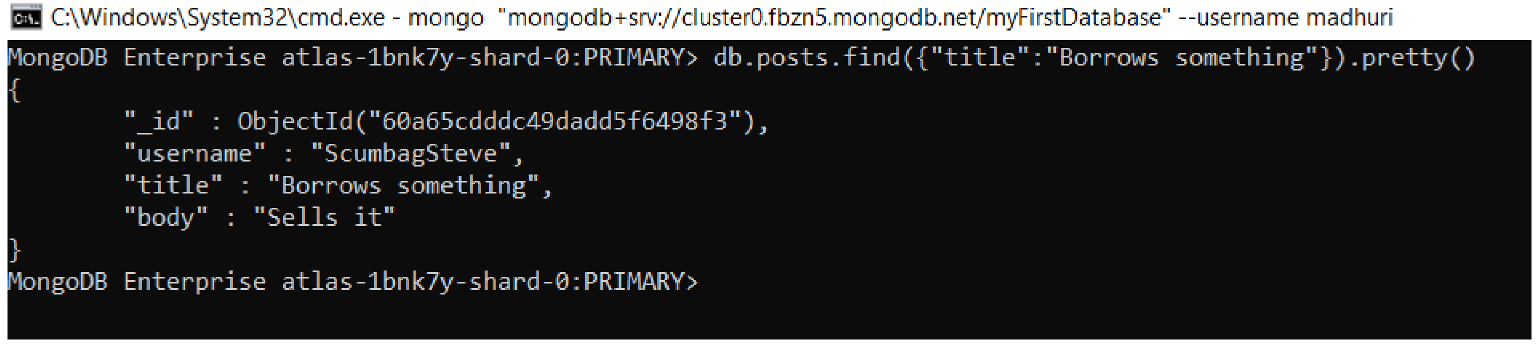
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"ScumbagSteve", title:"Borrows something", body: "Sells it"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"ScumbagSteve", title:"Borrows everything", body: "The end"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body: "Stes to private"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

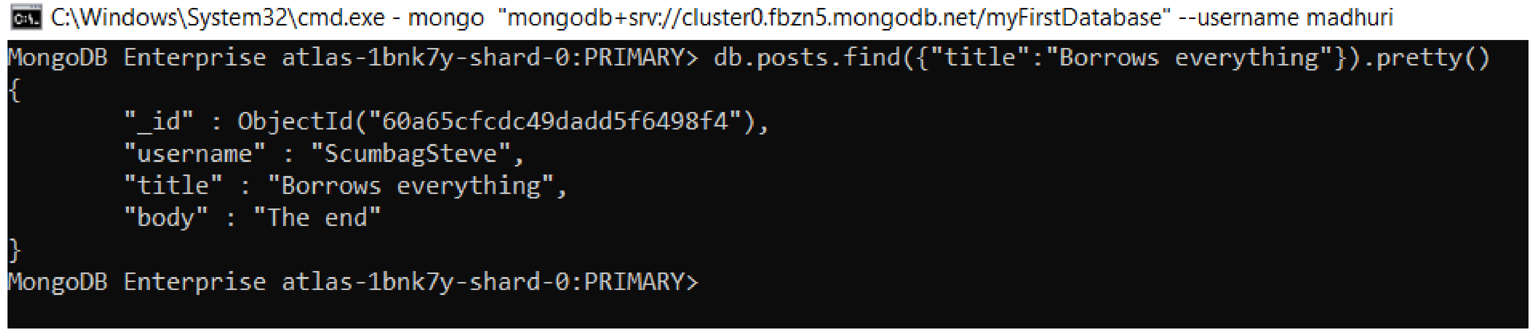
```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.users.insert({username:"GoodGuyGreg", first_name:"Good Guy", last_name:"Greg"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.users.insert({username:"ScumbagSteve", first_name:"Scumbag", last_name:"Steve"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body: "Wakes up early and cleans house"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Steals your identity", body: "Raises your credit score"})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body: "Sends you a Pull Request"})
WriteResult({ "nInserted" : 1 })
```

19. Insert the following documents into a **comments** collection.



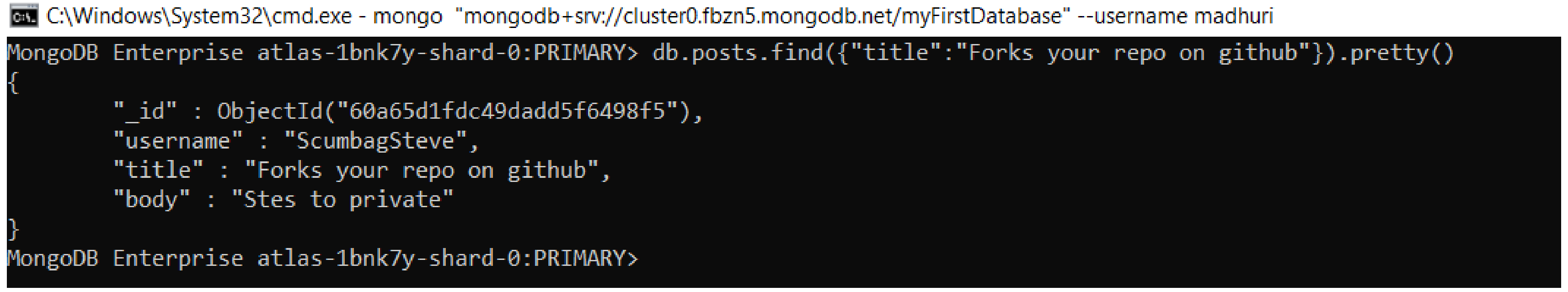
C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"title":"Borrows something"}).pretty()
{
        "_id" : ObjectId("60a65cdddc49dadd5f6498f3"),
        "username" : "ScumbagSteve",
        "title" : "Borrows something",
        "body" : "Sells it"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```



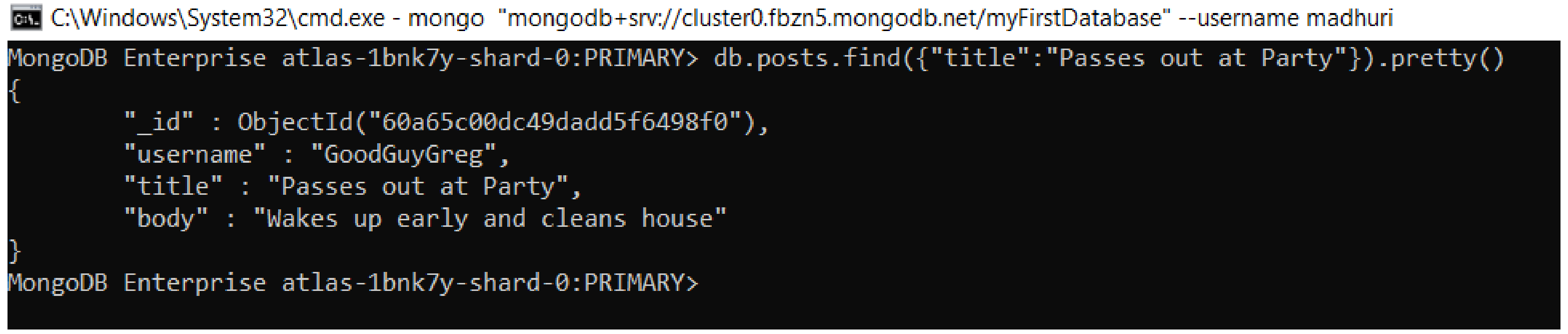
C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"title":"Borrows everything"}).pretty()
{
        "_id" : ObjectId("60a65cfcdc49dadd5f6498f4"),
        "username" : "ScumbagSteve",
        "title" : "Borrows everything",
        "body" : "The end"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```



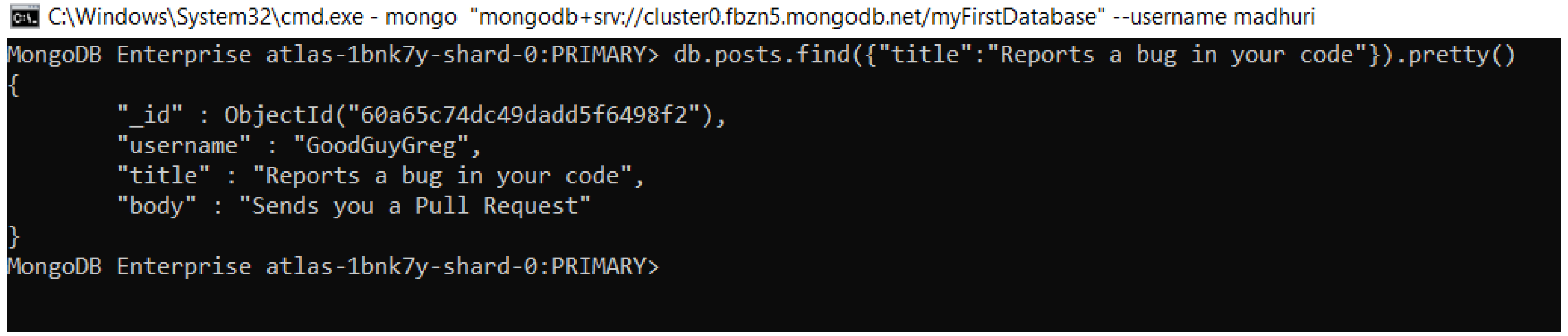
C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"title":"Forks your repo on github"}).pretty()
{
        "_id" : ObjectId("60a65d1fdc49dadd5f6498f5"),
        "username" : "ScumbagSteve",
        "title" : "Forks your repo on github",
        "body" : "Stes to private"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"title":"Passes out at Party"}).pretty()
{
        "_id" : ObjectId("60a65c00dc49dadd5f6498f0"),
        "username" : "GoodGuyGreg",
        "title" : "Passes out at Party",
        "body" : "Wakes up early and cleans house"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

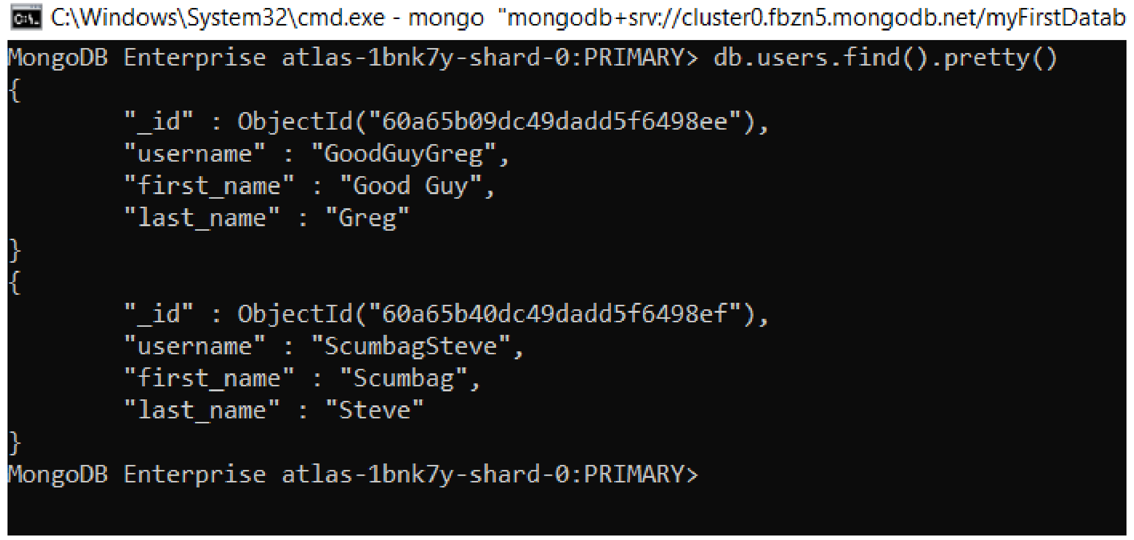


C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"title":"Reports a bug in your code"}).pretty()
{
        "_id" : ObjectId("60a65c74dc49dadd5f6498f2"),
        "username" : "GoodGuyGreg",
        "title" : "Reports a bug in your code",
        "body" : "Sends you a Pull Request"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

E) Querying related collection

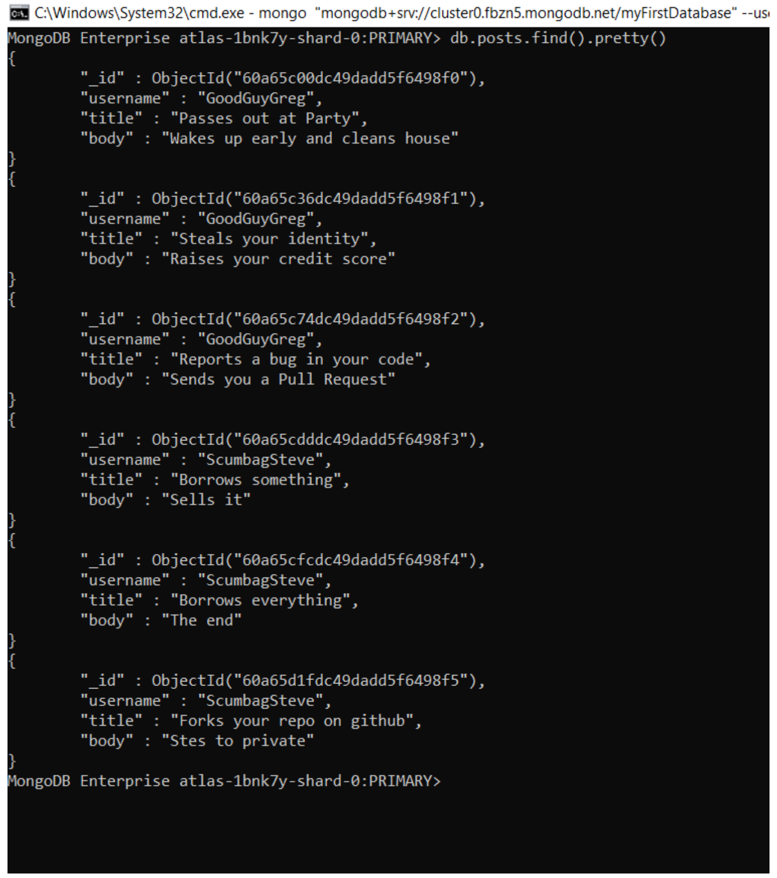
20. Find all users



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatab

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.users.find().pretty()
{
        "_id" : ObjectId("60a65b09dc49dadd5f6498ee"),
        "username" : "GoodGuyGreg",
        "first_name" : "Good Guy",
        "last_name" : "Greg"
}
{
        "_id" : ObjectId("60a65b40dc49dadd5f6498ef"),
        "username" : "ScumbagSteve",
        "first_name" : "Scumbag",
        "last_name" : "Steve"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

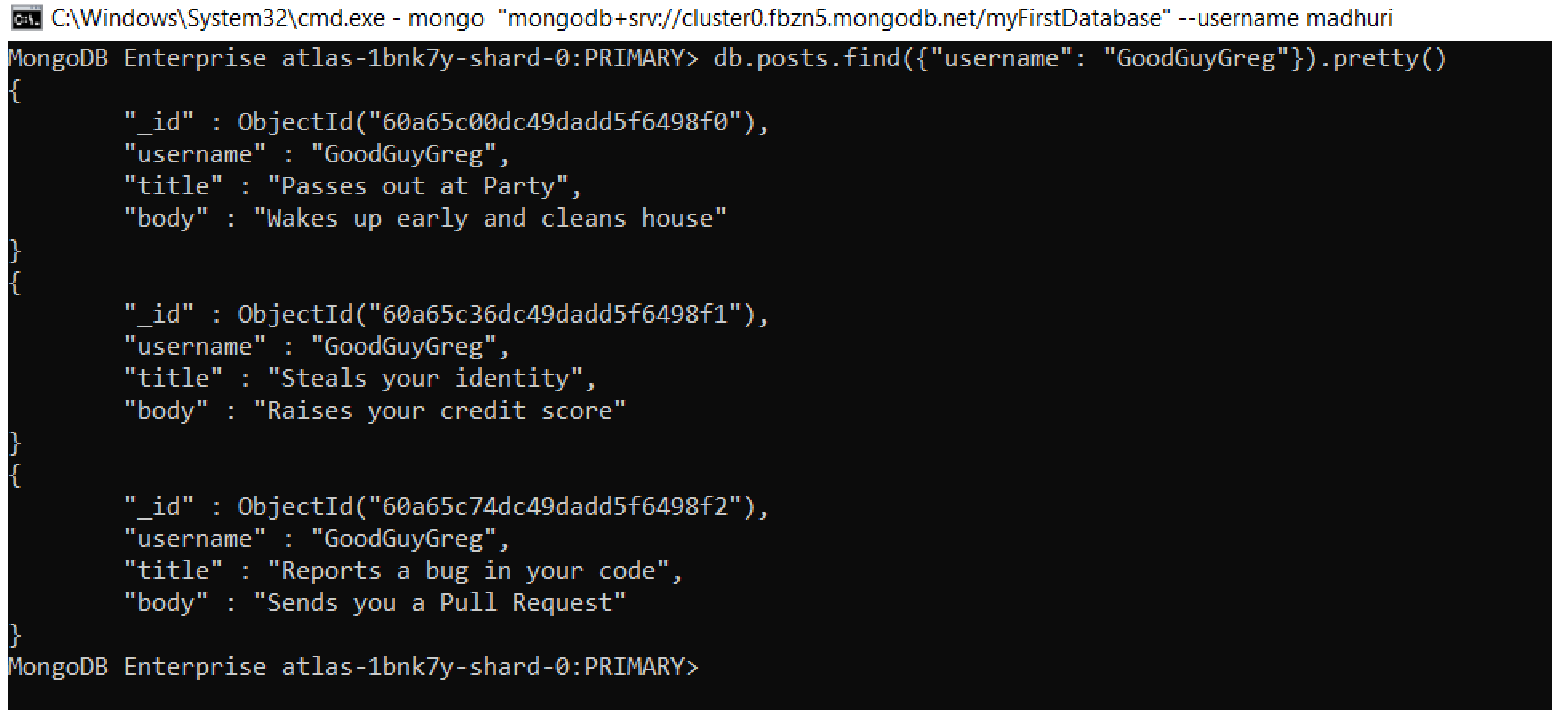
21. Find all posts



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --us

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find().pretty()
{
        "_id" : ObjectId("60a65c00dc49dadd5f6498f0"),
        "username" : "GoodGuyGreg",
        "title" : "Passes out at Party",
        "body" : "Wakes up early and cleans house"
}
{
        "_id" : ObjectId("60a65c36dc49dadd5f6498f1"),
        "username" : "GoodGuyGreg",
        "title" : "Steals your identity",
        "body" : "Raises your credit score"
}
{
        "_id" : ObjectId("60a65c74dc49dadd5f6498f2"),
        "username" : "GoodGuyGreg",
        "title" : "Reports a bug in your code",
        "body" : "Sends you a Pull Request"
}
{
        "_id" : ObjectId("60a65cdddc49dadd5f6498f3"),
        "username" : "ScumbagSteve",
        "title" : "Borrows something",
        "body" : "Sells it"
}
{
        "_id" : ObjectId("60a65cfcdc49dadd5f6498f4"),
        "username" : "ScumbagSteve",
        "title" : "Borrows everything",
        "body" : "The end"
}
{
        "_id" : ObjectId("60a65d1fdc49dadd5f6498f5"),
        "username" : "ScumbagSteve",
        "title" : "Forks your repo on github",
        "body" : "Stes to private"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

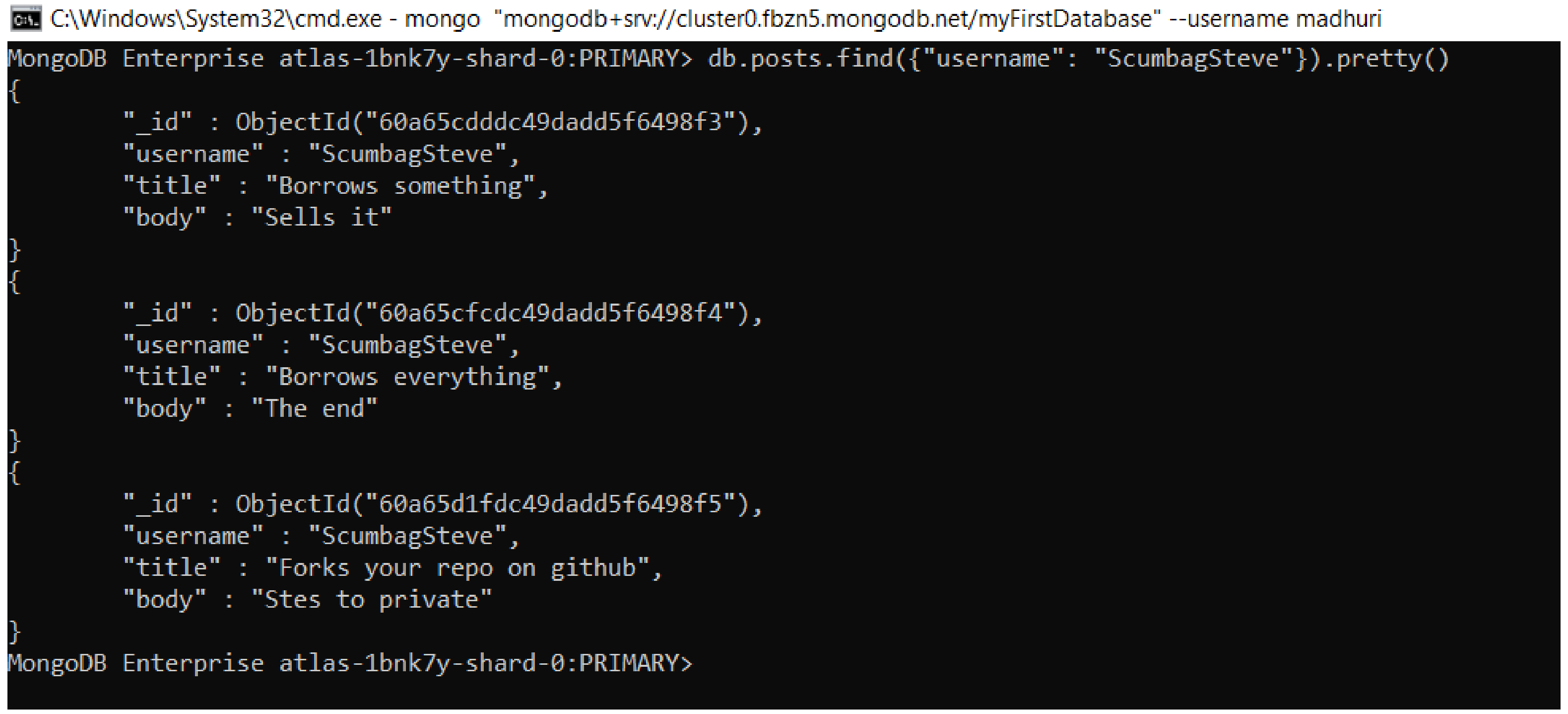
22. Find all posts that was authored by "GoodGuyGreg".



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"username": "GoodGuyGreg"}).pretty()
{
        "_id" : ObjectId("60a65c00dc49dadd5f6498f0"),
        "username" : "GoodGuyGreg",
        "title" : "Passes out at Party",
        "body" : "Wakes up early and cleans house"
}
{
        "_id" : ObjectId("60a65c36dc49dadd5f6498f1"),
        "username" : "GoodGuyGreg",
        "title" : "Steals your identity",
        "body" : "Raises your credit score"
}
{
        "_id" : ObjectId("60a65c74dc49dadd5f6498f2"),
        "username" : "GoodGuyGreg",
        "title" : "Reports a bug in your code",
        "body" : "Sends you a Pull Request"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

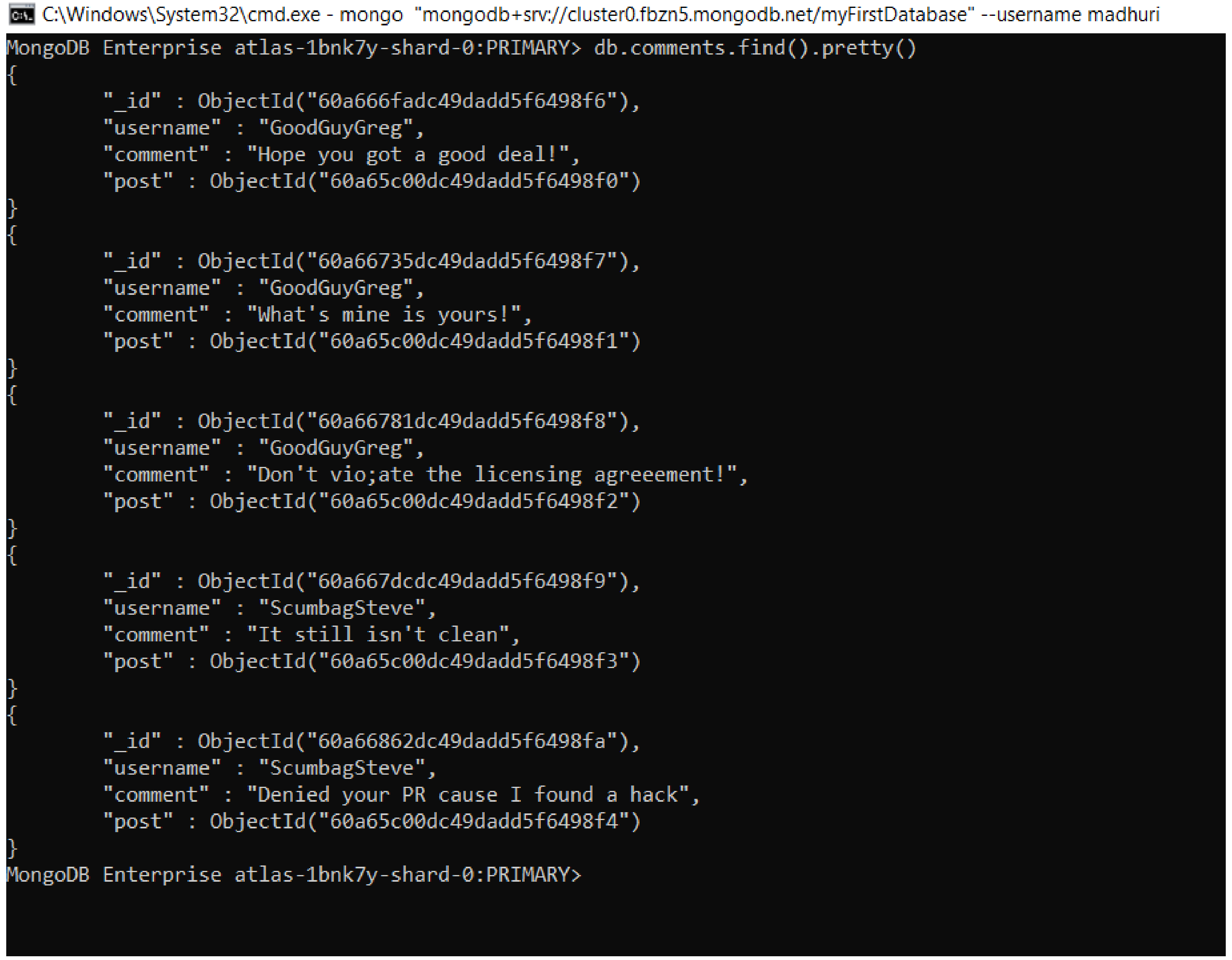
23. Find all posts that was authored by "ScumbagSteve”.



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.posts.find({"username": "ScumbagSteve"}).pretty()
{
        "_id" : ObjectId("60a65cdddc49dadd5f6498f3"),
        "username" : "ScumbagSteve",
        "title" : "Borrows something",
        "body" : "Sells it"
}
{
        "_id" : ObjectId("60a65cfcdc49dadd5f6498f4"),
        "username" : "ScumbagSteve",
        "title" : "Borrows everything",
        "body" : "The end"
}
{
        "_id" : ObjectId("60a65d1fdc49dadd5f6498f5"),
        "username" : "ScumbagSteve",
        "title" : "Forks your repo on github",
        "body" : "Stes to private"
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

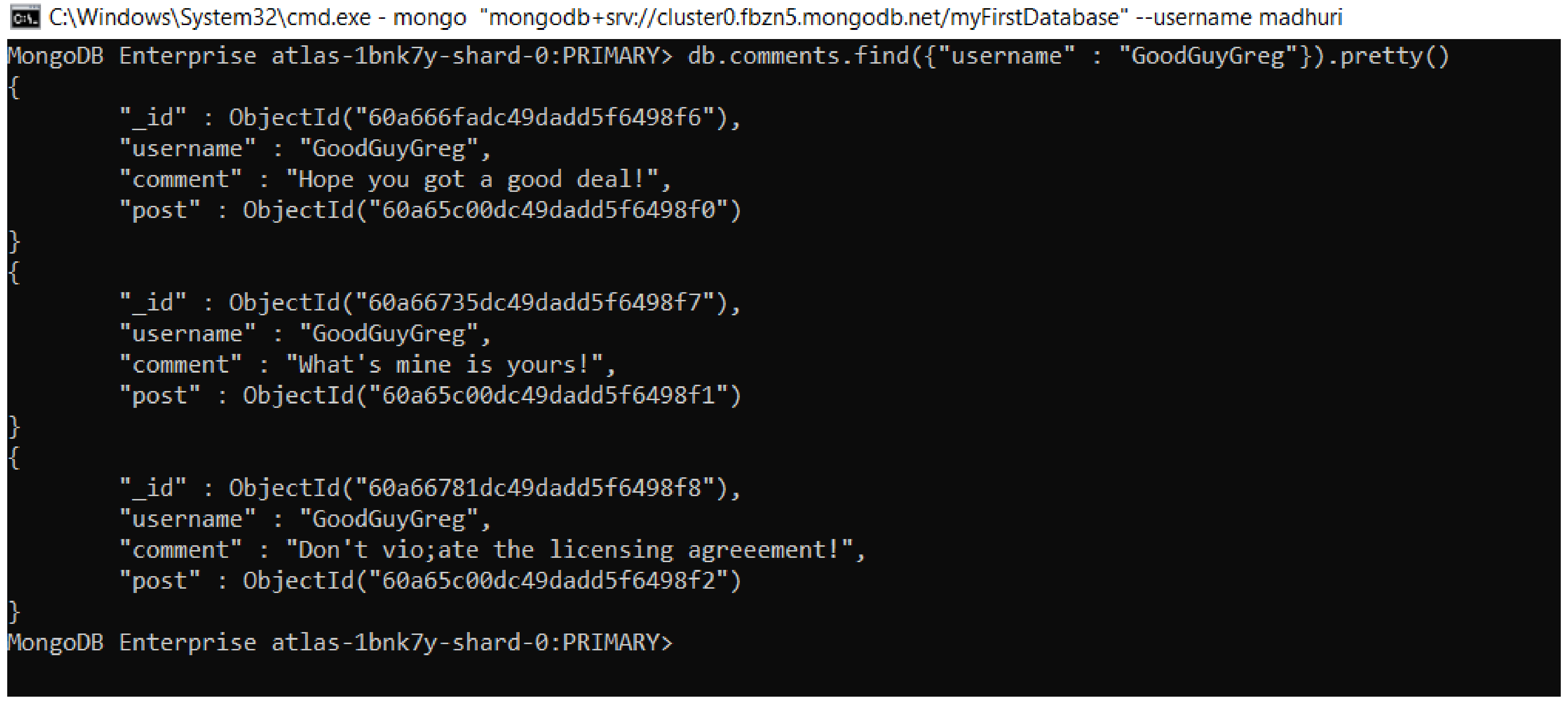
24. Find all comments.



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.comments.find().pretty()
{
        "_id" : ObjectId("60a666fadc49dadd5f6498f6"),
        "username" : "GoodGuyGreg",
        "comment" : "Hope you got a good deal!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f0")
}
{
        "_id" : ObjectId("60a66735dc49dadd5f6498f7"),
        "username" : "GoodGuyGreg",
        "comment" : "What's mine is yours!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f1")
}
{
        "_id" : ObjectId("60a66781dc49dadd5f6498f8"),
        "username" : "GoodGuyGreg",
        "comment" : "Don't vio;ate the licensing agreeement!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f2")
}
{
        "_id" : ObjectId("60a667dcdc49dadd5f6498f9"),
        "username" : "ScumbagSteve",
        "comment" : "It still isn't clean",
        "post" : ObjectId("60a65c00dc49dadd5f6498f3")
}
{
        "_id" : ObjectId("60a66862dc49dadd5f6498fa"),
        "username" : "ScumbagSteve",
        "comment" : "Denied your PR cause I found a hack",
        "post" : ObjectId("60a65c00dc49dadd5f6498f4")
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

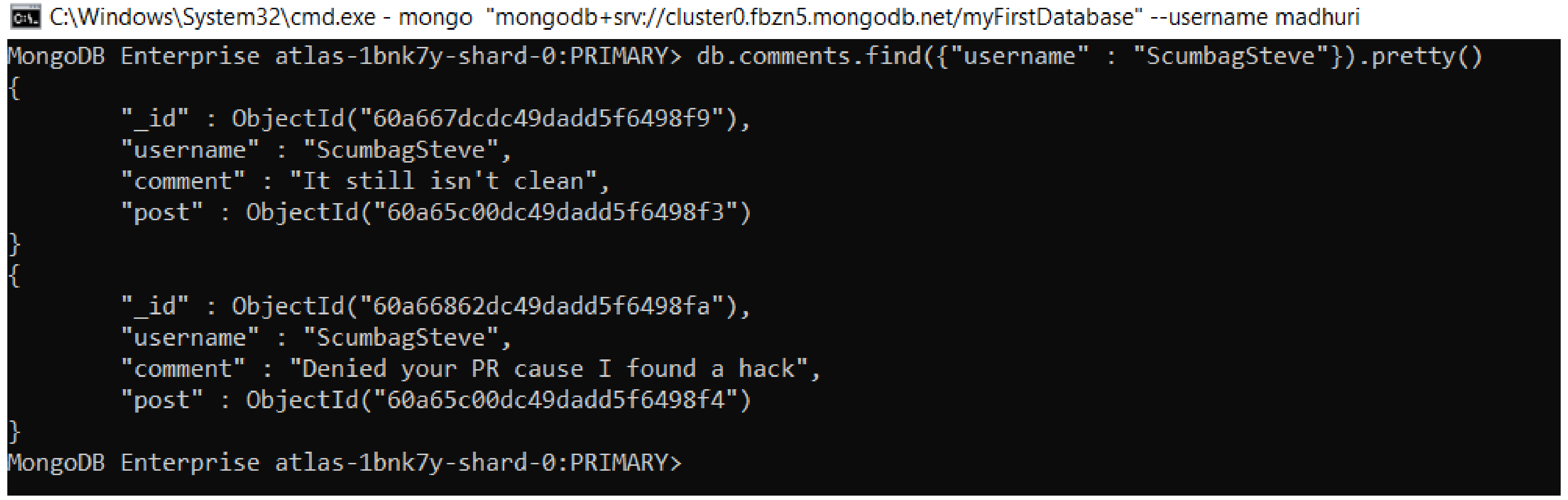
25. Find all comments that was authored by "GoodGuyGreg"



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.comments.find({"username" : "GoodGuyGreg"}).pretty()
{
        "_id" : ObjectId("60a666fadc49dadd5f6498f6"),
        "username" : "GoodGuyGreg",
        "comment" : "Hope you got a good deal!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f0")
}
{
        "_id" : ObjectId("60a66735dc49dadd5f6498f7"),
        "username" : "GoodGuyGreg",
        "comment" : "What's mine is yours!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f1")
}
{
        "_id" : ObjectId("60a66781dc49dadd5f6498f8"),
        "username" : "GoodGuyGreg",
        "comment" : "Don't vio;ate the licensing agreeement!",
        "post" : ObjectId("60a65c00dc49dadd5f6498f2")
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

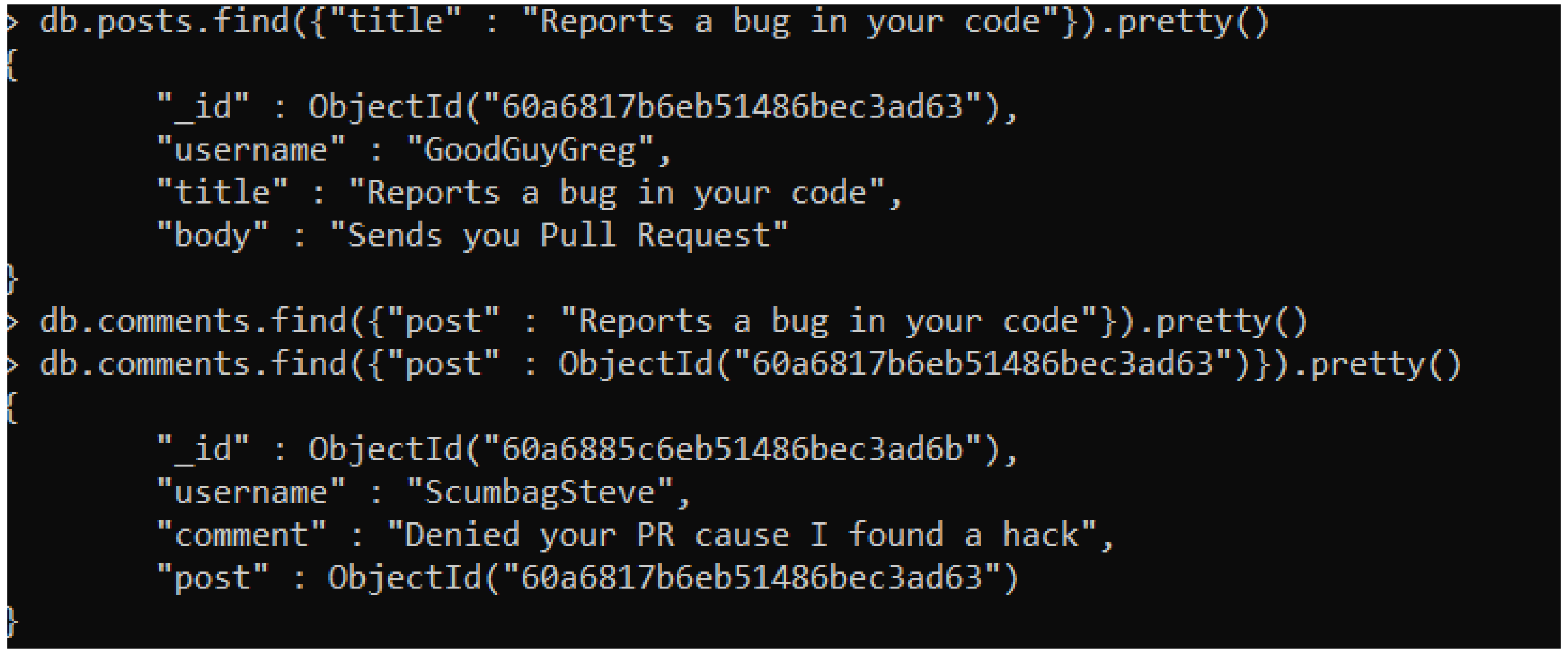
26. find all comments that was authored by "ScumbagSteve".



C:\Windows\System32\cmd.exe - mongo "mongodb+srv://cluster0.fbzn5.mongodb.net/myFirstDatabase" --username madhuri

```
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY> db.comments.find({"username" : "ScumbagSteve"}).pretty()
{
        "_id" : ObjectId("60a667dcdc49dadd5f6498f9"),
        "username" : "ScumbagSteve",
        "comment" : "It still isn't clean",
        "post" : ObjectId("60a65c00dc49dadd5f6498f3")
}
{
        "_id" : ObjectId("60a66862dc49dadd5f6498fa"),
        "username" : "ScumbagSteve",
        "comment" : "Denied your PR cause I found a hack",
        "post" : ObjectId("60a65c00dc49dadd5f6498f4")
}
MongoDB Enterprise atlas-1bnk7y-shard-0:PRIMARY>
```

27. Find all comments belonging to the post "Reports a bug in your code"



```
> db.posts.find({"title" : "Reports a bug in your code"}).pretty()
{
        "_id" : ObjectId("60a6817b6eb51486bec3ad63"),
        "username" : "GoodGuyGreg",
        "title" : "Reports a bug in your code",
        "body" : "Sends you Pull Request"
}
> db.comments.find({"post" : "Reports a bug in your code"}).pretty()
> db.comments.find({"post" : ObjectId("60a6817b6eb51486bec3ad63")}).pretty()
{
        "_id" : ObjectId("60a6885c6eb51486bec3ad6b"),
        "username" : "ScumbagSteve",
        "comment" : "Denied your PR cause I found a hack",
        "post" : ObjectId("60a6817b6eb51486bec3ad63")
}
```