MongoDB Lab Assignment-1

MongoDB Exercise in mongo shell

Connect to a running mongo instance, use a database named mongo_practice. Document all your queries in a javascript file to use as a reference.

Insert Documents

Insert the following documents into a movies collection.

Query / Find Documents query the movies collection to

1. get all documents

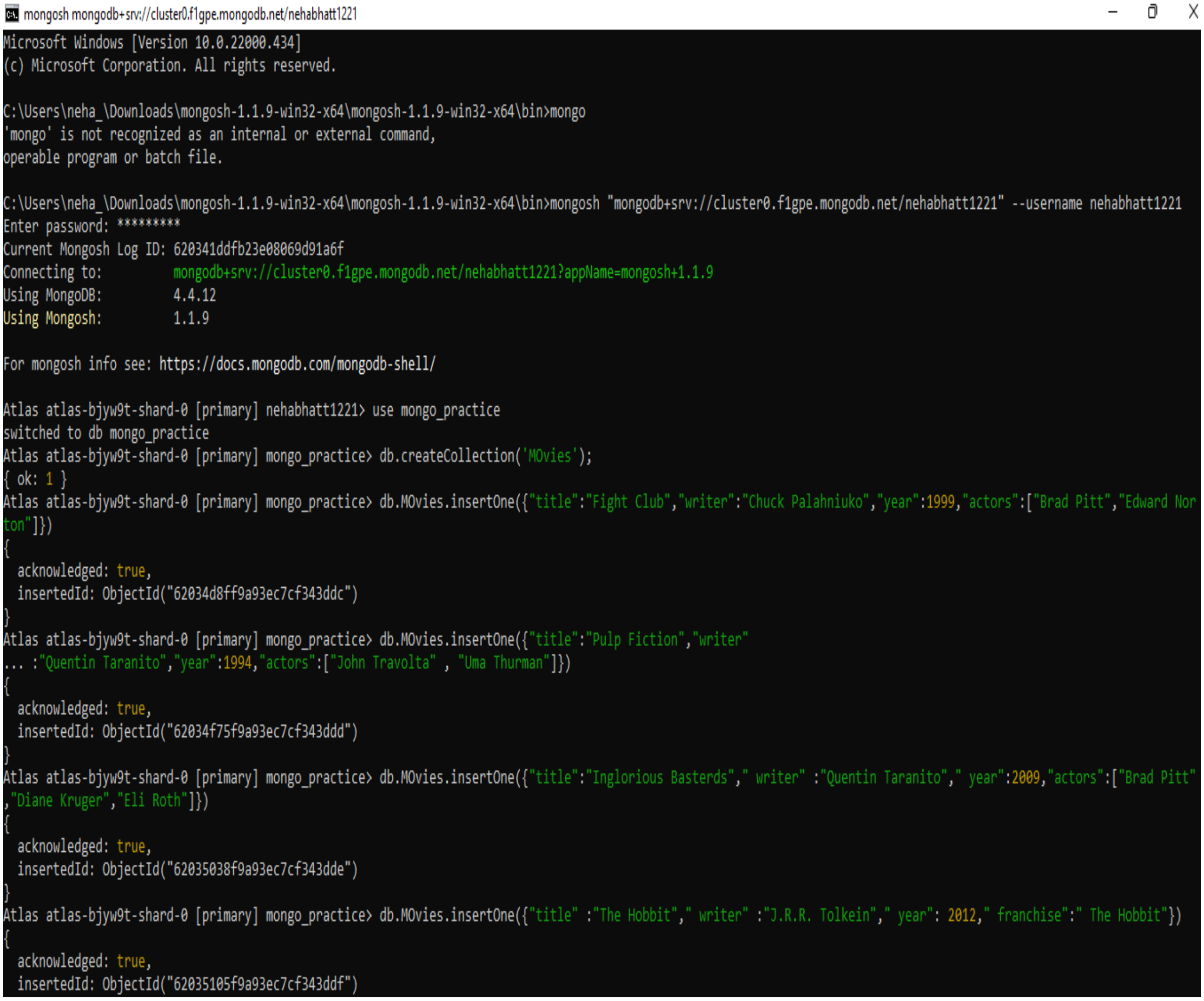
2. get all documents with writer set to "Quentin Tarantino"

3. get all documents where actors include "Brad Pitt"

4. get all documents with franchise set to "The Hobbit"

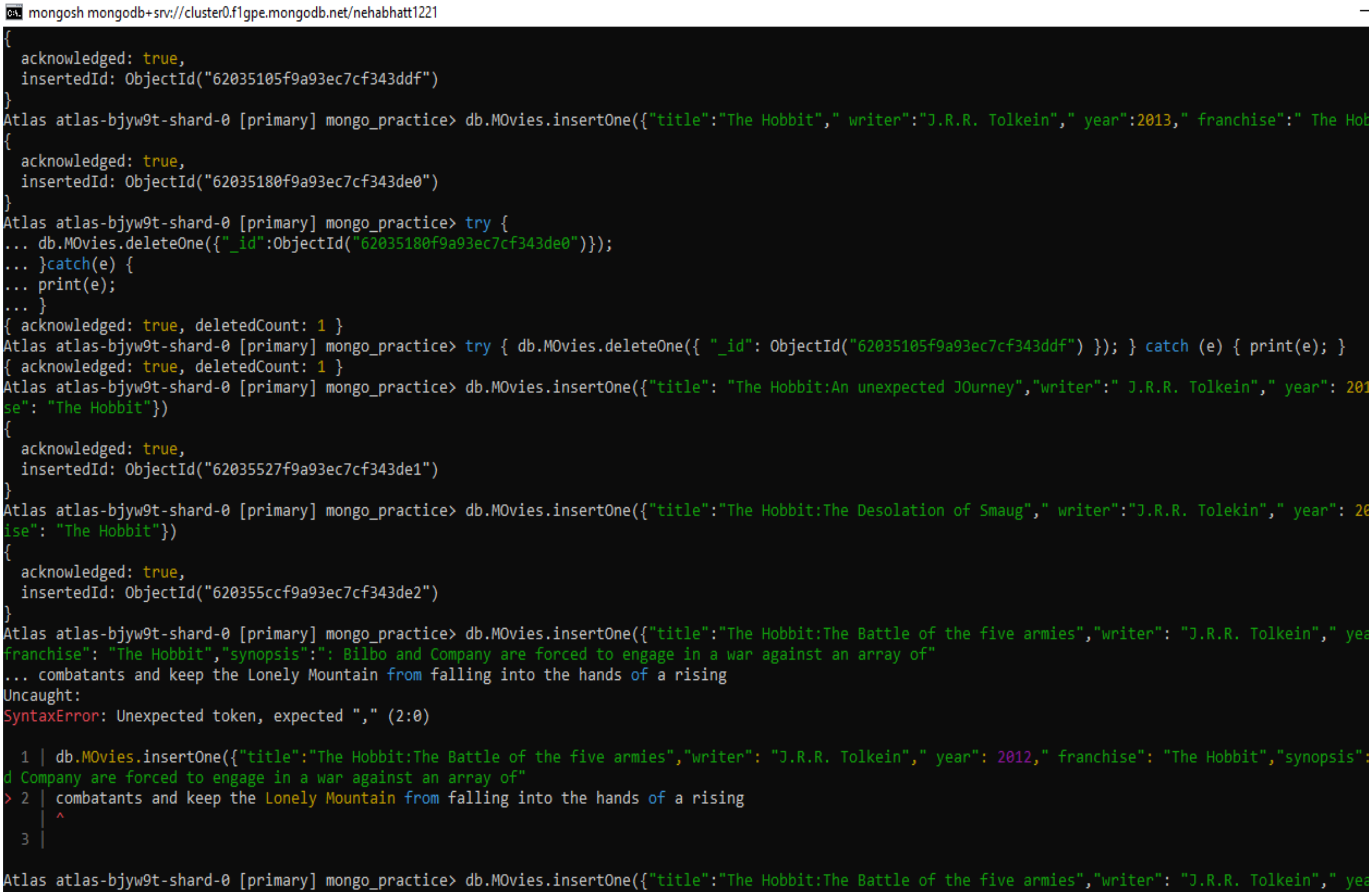
5. get all movies released in the 90s

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

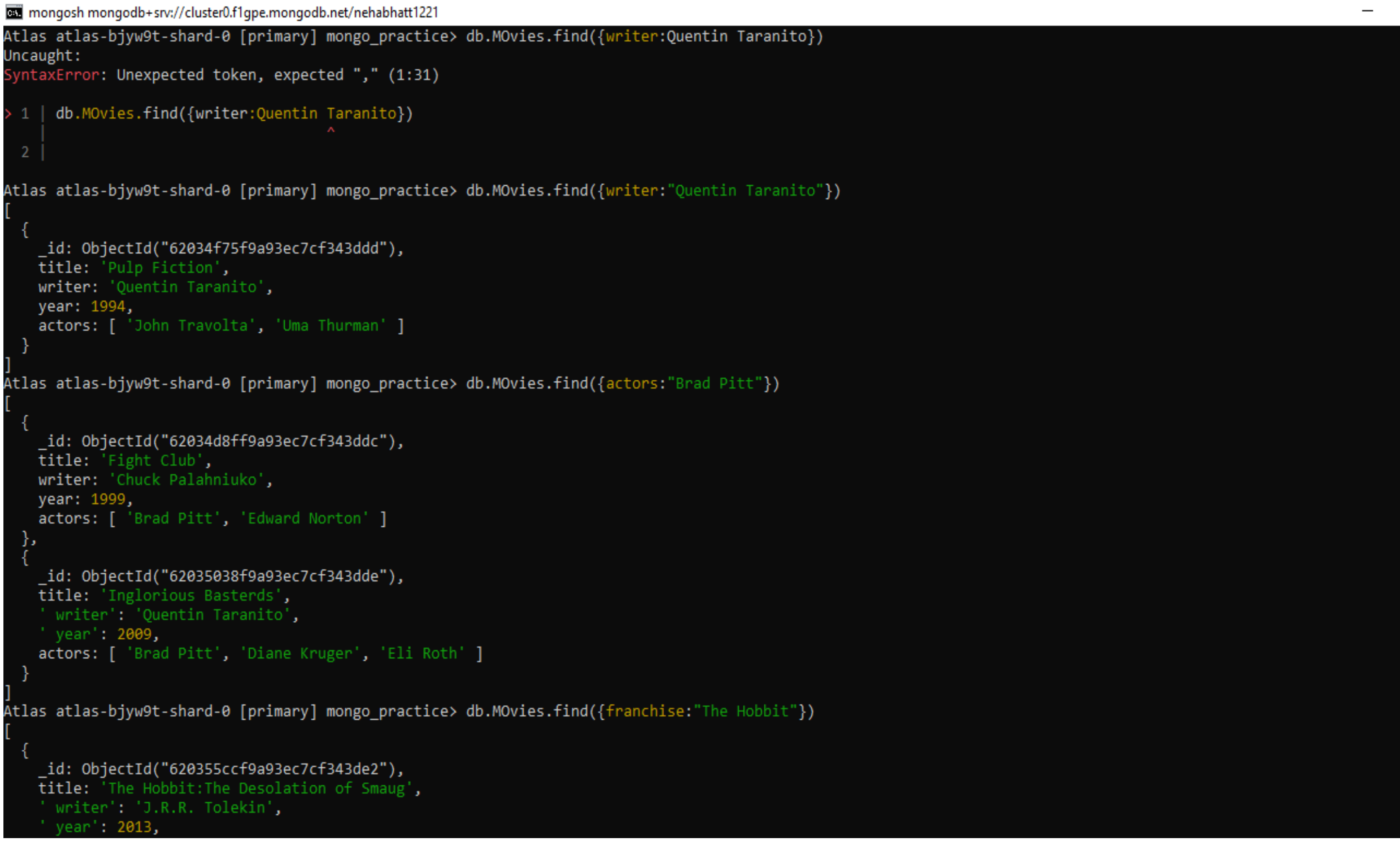


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\neha_\Downloads\mongosh-1.1.9-win32-x64\mongosh-1.1.9-win32-x64\bin>mongo
'mongo' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\neha_\Downloads\mongosh-1.1.9-win32-x64\mongosh-1.1.9-win32-x64\bin>mongosh "mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221" --username nehabhatt1221
Enter password: *********
Current Mongosh Log ID: 620341ddfb23e08069d91a6f
Connecting to:          mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221?appName=mongosh+1.1.9
Using MongoDB:          4.4.12
Using Mongosh:          1.1.9

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

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> use mongo_practice
switched to db mongo_practice
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.createCollection('MOvies');
{ ok: 1 }
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"Fight Club","writer":"Chuck Palahniuko","year":1999,"actors":["Brad Pitt","Edward Nor
ton"]})
{
  acknowledged: true,
  insertedId: ObjectId("62034d8ff9a93ec7cf343ddc")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"Pulp Fiction","writer"
... :"Quentin Taranito","year":1994,"actors":["John Travolta" , "Uma Thurman"]})
{
  acknowledged: true,
  insertedId: ObjectId("62034f75f9a93ec7cf343ddd")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"Inglorious Basterds"," writer" :"Quentin Taranito"," year":2009,"actors":["Brad Pitt"
,"Diane Kruger","Eli Roth"]})
{
  acknowledged: true,
  insertedId: ObjectId("62035038f9a93ec7cf343dde")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title" :"The Hobbit"," writer" :"J.R.R. Tolkein"," year": 2012," franchise":" The Hobbit"})
{
  acknowledged: true,
  insertedId: ObjectId("62035105f9a93ec7cf343ddf")
```

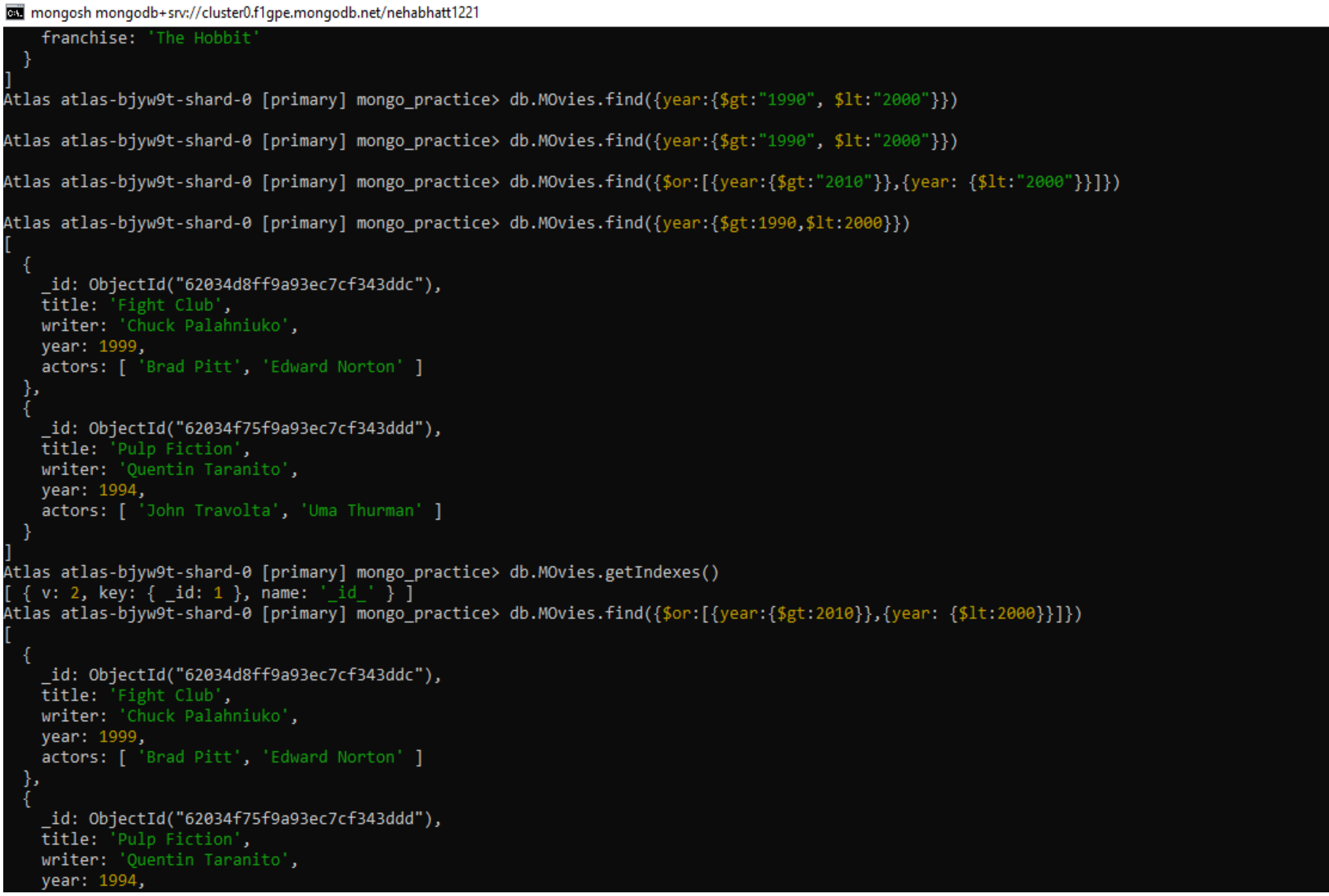

mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
{
  acknowledged: true,
  insertedId: ObjectId("62035105f9a93ec7cf343ddf")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"The Hobbit"," writer":"J.R.R. Tolkein"," year":2013," franchise":" The Hob
{
  acknowledged: true,
  insertedId: ObjectId("62035180f9a93ec7cf343de0")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> try {
... db.MOvies.deleteOne({"_id":ObjectId("62035180f9a93ec7cf343de0")});
... }catch(e) {
... print(e);
... }
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> try { db.MOvies.deleteOne({ "_id": ObjectId("62035105f9a93ec7cf343ddf") }); } catch (e) { print(e); }
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title": "The Hobbit:An unexpected JOurney","writer":" J.R.R. Tolkein"," year": 201
se": "The Hobbit"})
{
  acknowledged: true,
  insertedId: ObjectId("62035527f9a93ec7cf343de1")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"The Hobbit:The Desolation of Smaug"," writer":"J.R.R. Tolekin"," year": 20
ise": "The Hobbit"})
{
  acknowledged: true,
  insertedId: ObjectId("620355ccf9a93ec7cf343de2")
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"The Hobbit:The Battle of the five armies","writer": "J.R.R. Tolkein"," yea
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 rising
Uncaught:
SyntaxError: Unexpected token, expected "," (2:0)

  1 | db.MOvies.insertOne({"title":"The Hobbit:The Battle of the five armies","writer": "J.R.R. Tolkein"," year": 2012," franchise": "The Hobbit","synopsis":
d Company are forced to engage in a war against an array of"
> 2 | combatants and keep the Lonely Mountain from falling into the hands of a rising
    | ^
  3 |

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.insertOne({"title":"The Hobbit:The Battle of the five armies","writer": "J.R.R. Tolkein"," yea
```



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({writer:Quentin Taranito})
Uncaught:
SyntaxError: Unexpected token, expected "," (1:31)

> 1 | db.MOvies.find({writer:Quentin Taranito})
    |                                ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({writer:"Quentin Taranito"})
[
  {
    _id: ObjectId("62034f75f9a93ec7cf343ddd"),
    title: 'Pulp Fiction',
    writer: 'Quentin Taranito',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({actors:"Brad Pitt"})
[
  {
    _id: ObjectId("62034d8ff9a93ec7cf343ddc"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62035038f9a93ec7cf343dde"),
    title: 'Inglorious Basterds',
    ' writer': 'Quentin Taranito',
    ' year': 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({franchise:"The Hobbit"})
[
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

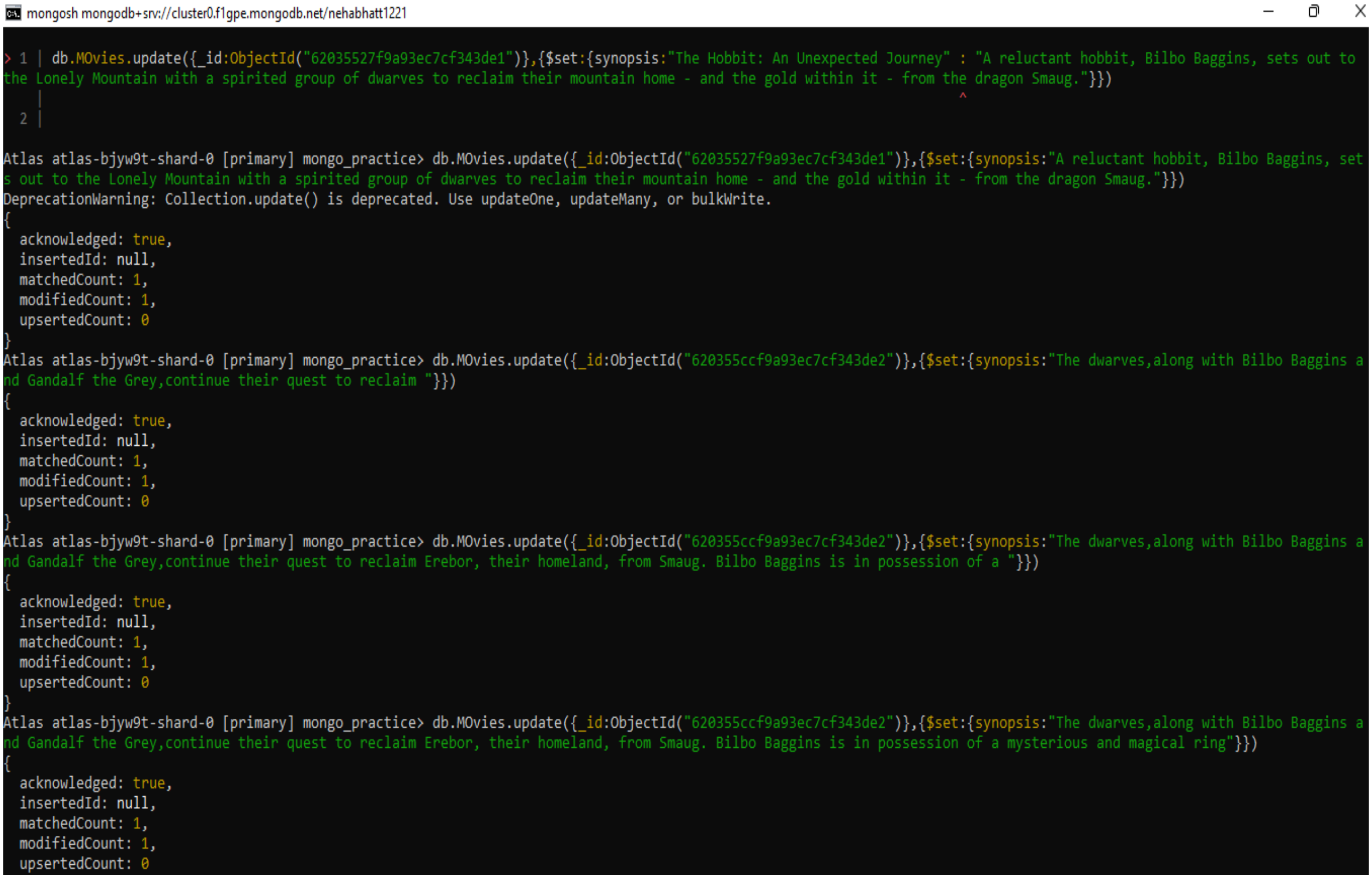
```
    franchise: 'The Hobbit'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({year:{$gt:"1990", $lt:"2000"}})

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({year:{$gt:"1990", $lt:"2000"}})

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({year:{$gt:1990,$lt:2000}})
[
  {
    _id: ObjectId("62034d8ff9a93ec7cf343ddc"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62034f75f9a93ec7cf343ddd"),
    title: 'Pulp Fiction',
    writer: 'Quentin Taranito',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$or:[{year:{$gt:2010}},{year: {$lt:2000}}]})
[
  {
    _id: ObjectId("62034d8ff9a93ec7cf343ddc"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62034f75f9a93ec7cf343ddd"),
    title: 'Pulp Fiction',
    writer: 'Quentin Taranito',
    year: 1994,
```

Update Documents

1. 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."

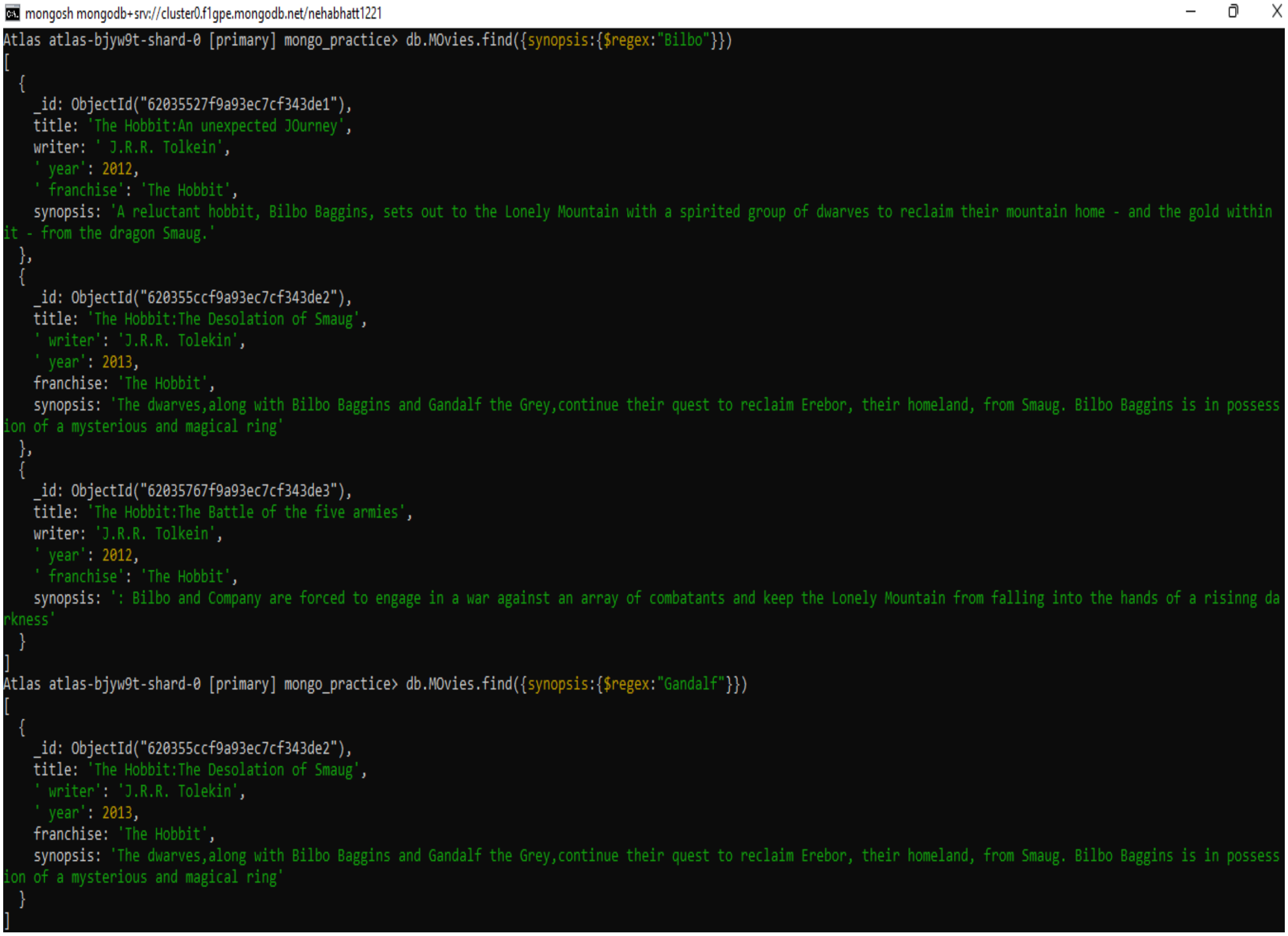
2. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
> 1 | db.MOvies.update({_id:ObjectId("62035527f9a93ec7cf343de1")},{$set:{synopsis:"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."}})
    |                                                                                                                     ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("62035527f9a93ec7cf343de1")},{$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, set
s out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Baggins a
nd Gandalf the Grey,continue their quest to reclaim "}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Baggins a
nd Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a "}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Baggins a
nd Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({synopsis:{$regex:"Bilbo"}})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    title: 'The Hobbit:An unexpected JOurney',
    writer: ' J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within
it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  },
  {
    _id: ObjectId("62035767f9a93ec7cf343de3"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: ': Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a risinng da
rkness'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({synopsis:{$regex:"Gandalf"}})
[
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  }
]
```

Text Search

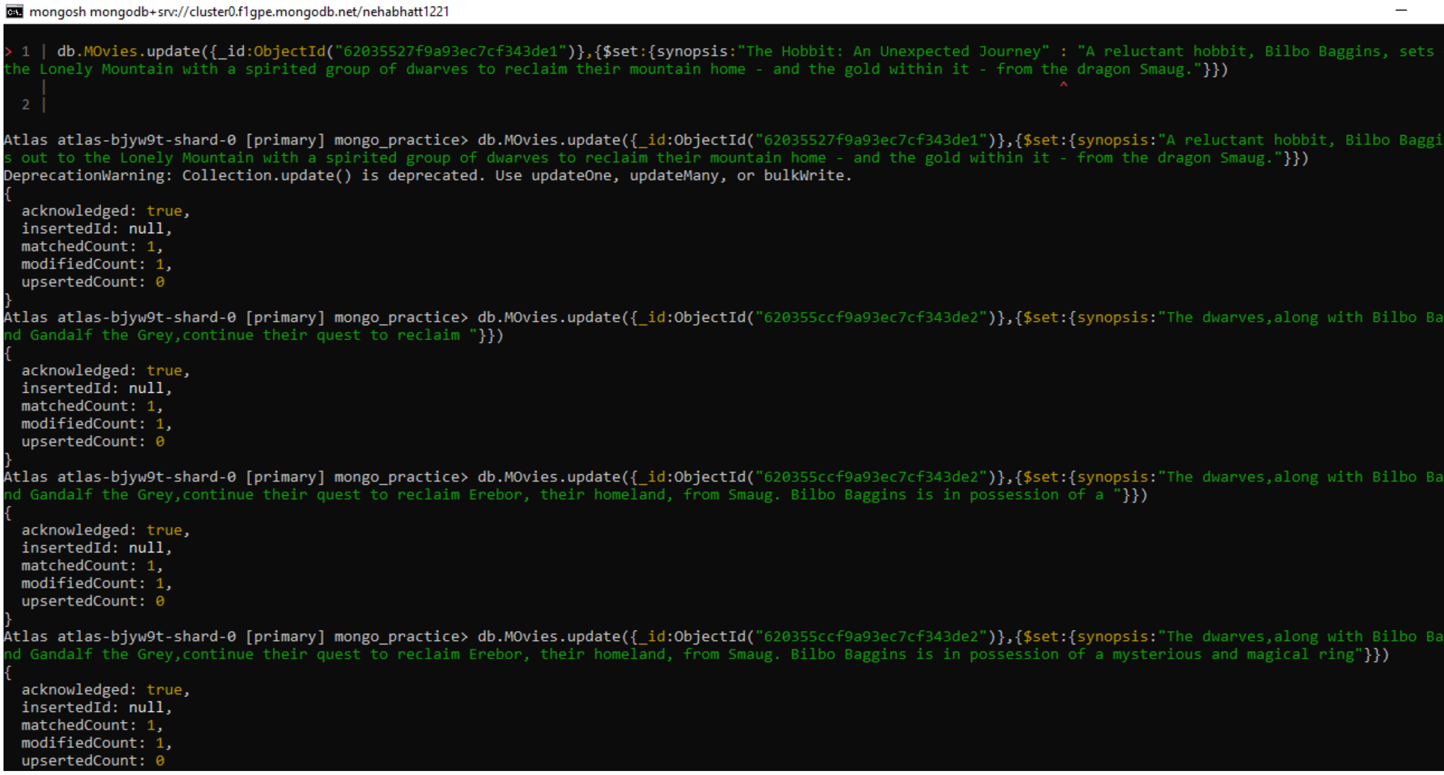
1. find all movies that have a synopsis that contains the word "Bilbo"

2. find all movies that have a synopsis that contains the word "Gandalf"

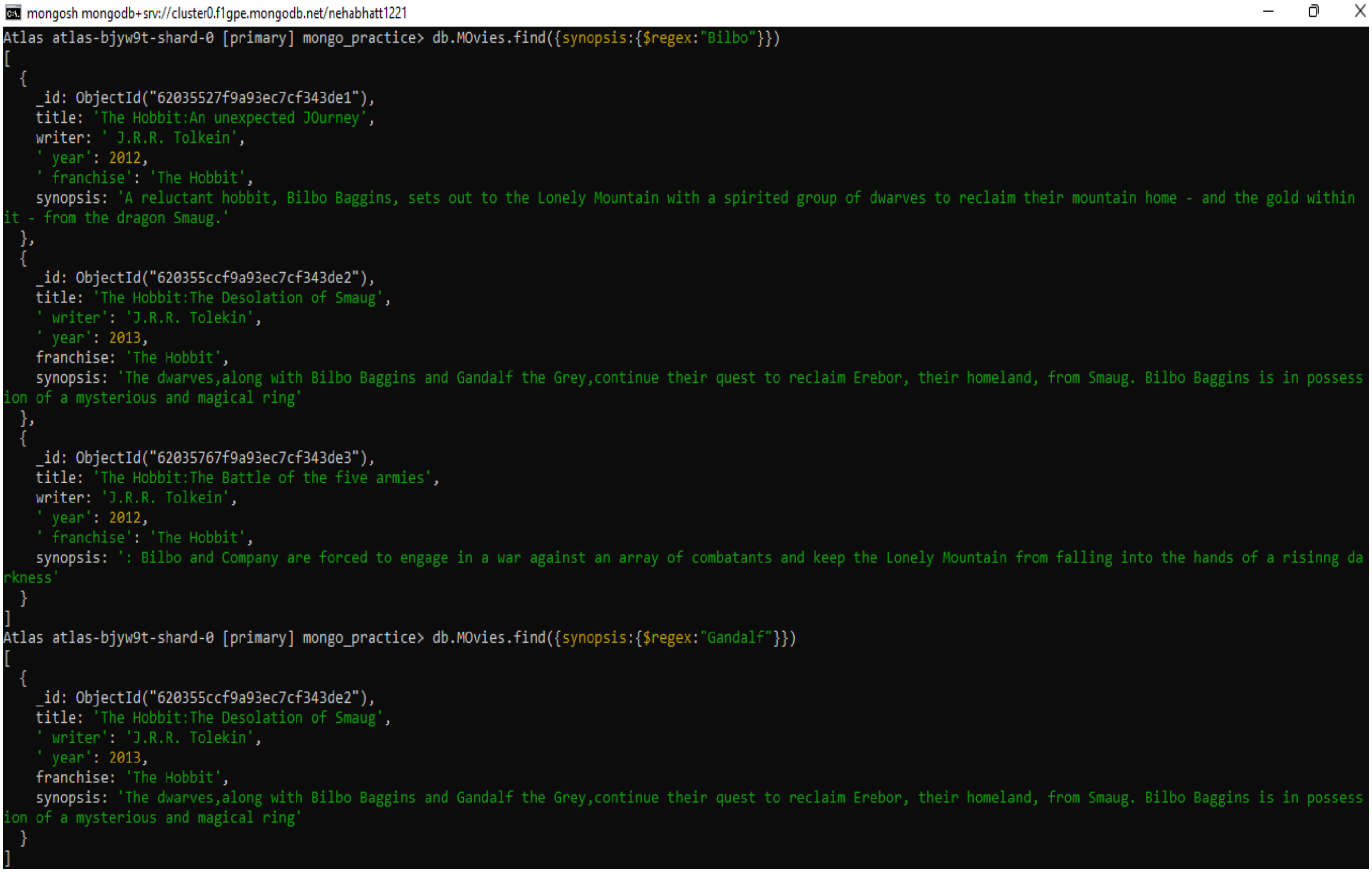
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

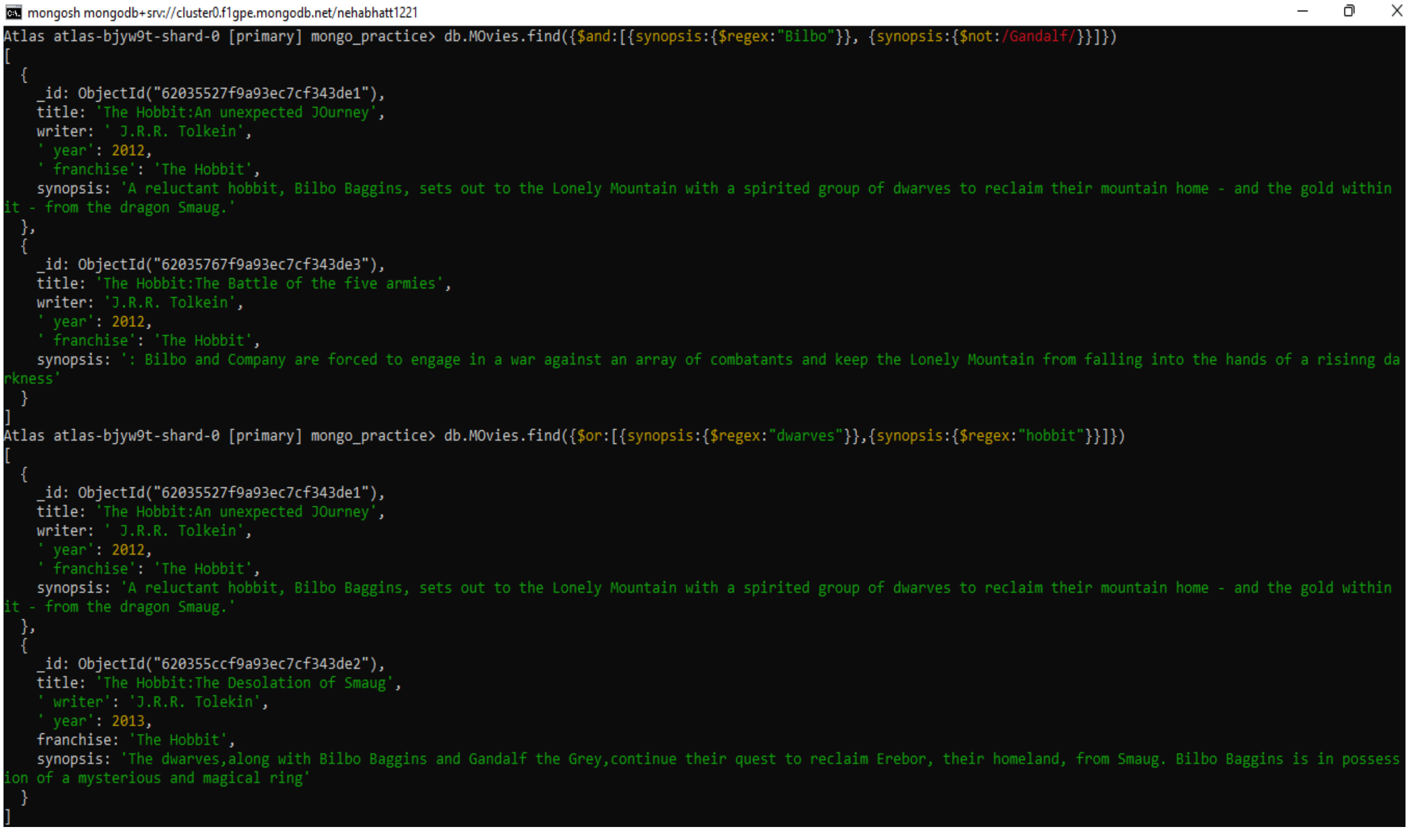
5. find all movies that have a synopsis that contains the word "gold" and "dragon"


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

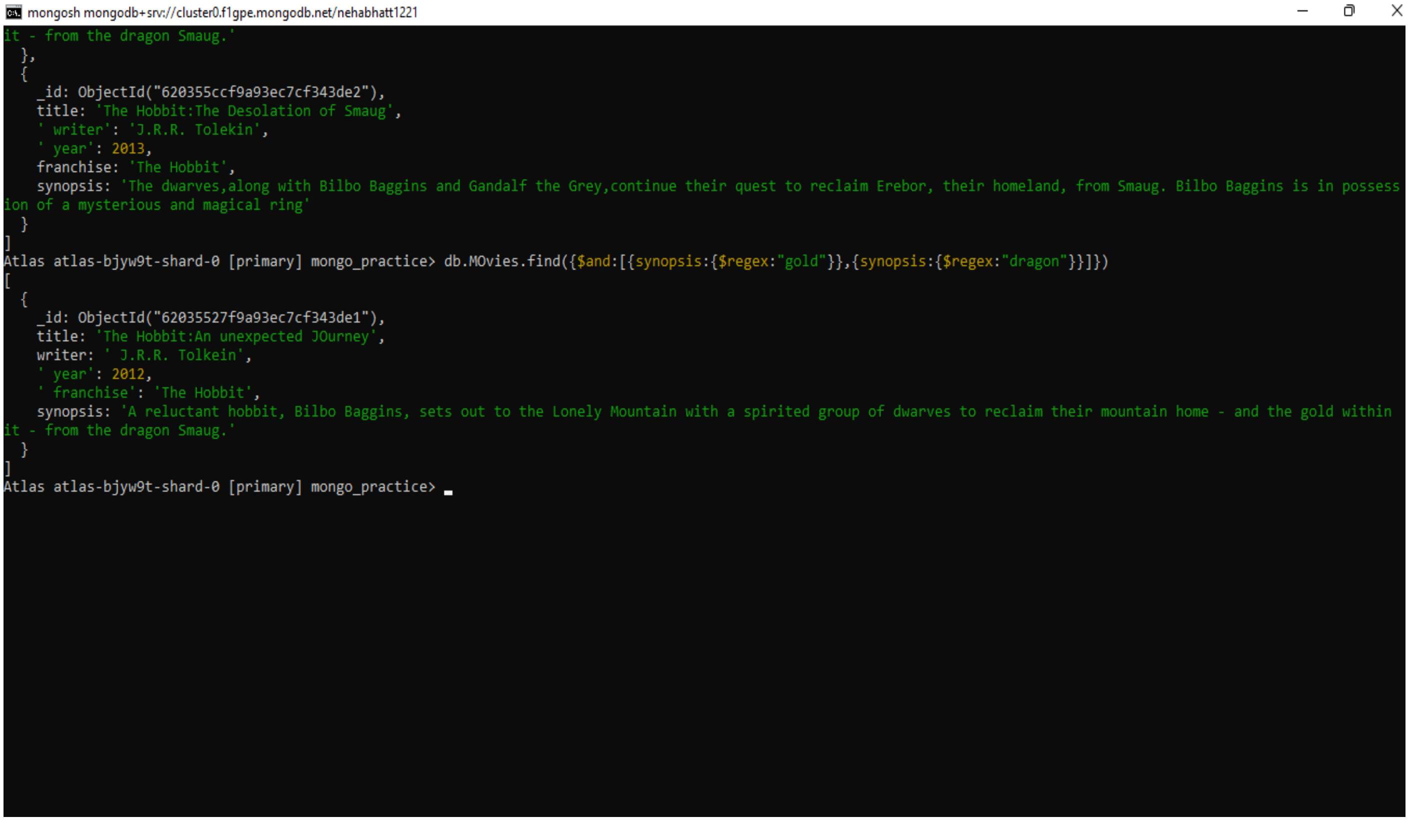
```
> 1 | db.MOvies.update({_id:ObjectId("62035527f9a93ec7cf343de1")},{$set:{synopsis:"The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets
the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
    |                                                                                                     ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("62035527f9a93ec7cf343de1")},{$set:{synopsis:"A reluctant hobbit, Bilbo Baggi
s out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Ba
nd Gandalf the Grey,continue their quest to reclaim "}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Ba
nd Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a "}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.update({_id:ObjectId("620355ccf9a93ec7cf343de2")},{$set:{synopsis:"The dwarves,along with Bilbo Ba
nd Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({synopsis:{$regex:"Bilbo"}})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    title: 'The Hobbit:An unexpected JOurney',
    writer: ' J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within
it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  },
  {
    _id: ObjectId("62035767f9a93ec7cf343de3"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: ': Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a risinng da
rkness'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({synopsis:{$regex:"Gandalf"}})
[
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  }
]
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    title: 'The Hobbit:An unexpected JOurney',
    writer: ' J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within
it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("62035767f9a93ec7cf343de3"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R. Tolkein',
    ' year': 2012,
    ' franchise': 'The Hobbit',
    synopsis: ': Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a risinng da
rkness'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$or:[{synopsis:{$regex:"dwarves"}},{synopsis:{$regex:"hobbit"}}]})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    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 - from the dragon Smaug.'
  },
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  }
]
```

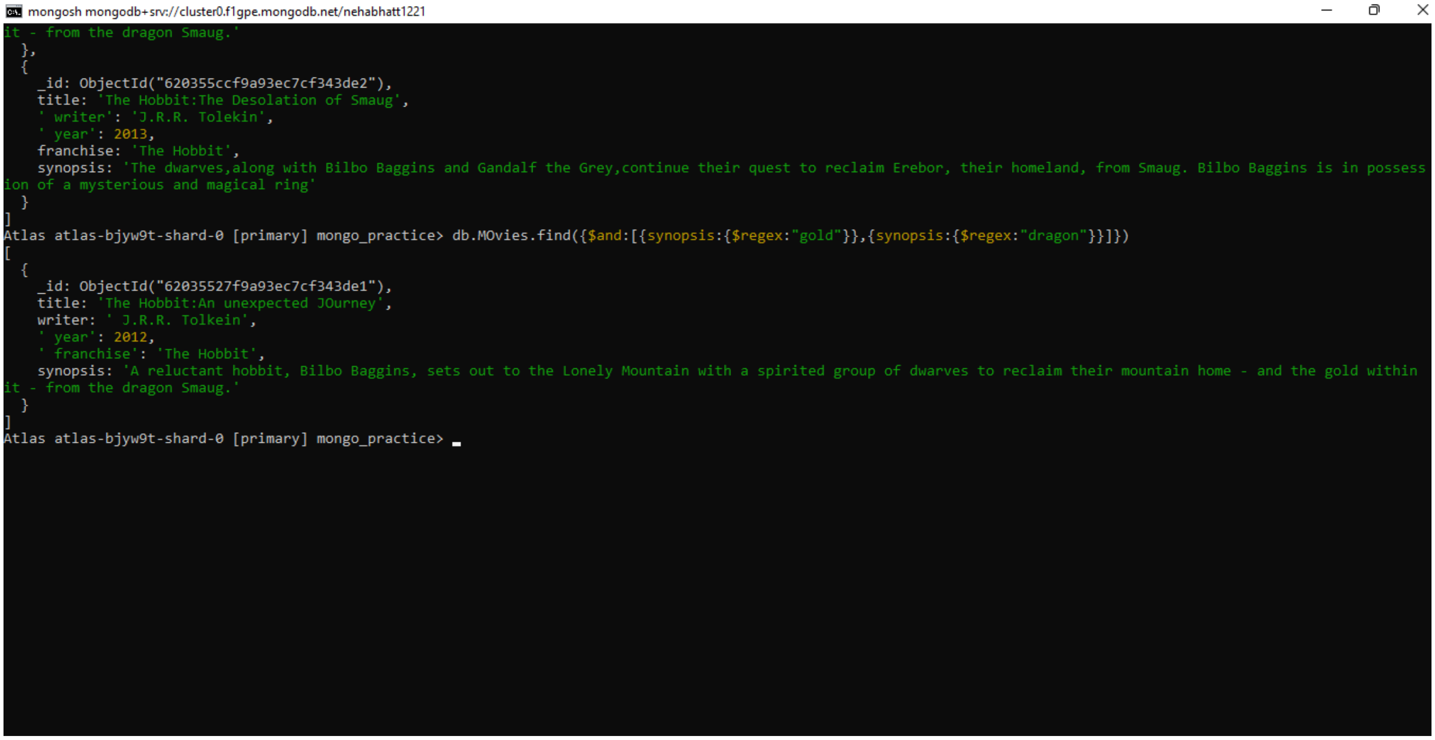

mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$and:[{synopsis:{$regex:"gold"}},{synopsis:{$regex:"dragon"}}]})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    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 - from the dragon Smaug.'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> _
```

Delete Documents

1. delete the movie "Pee Wee Herman's Big Adventure"

2. delete the movie "Avatar



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

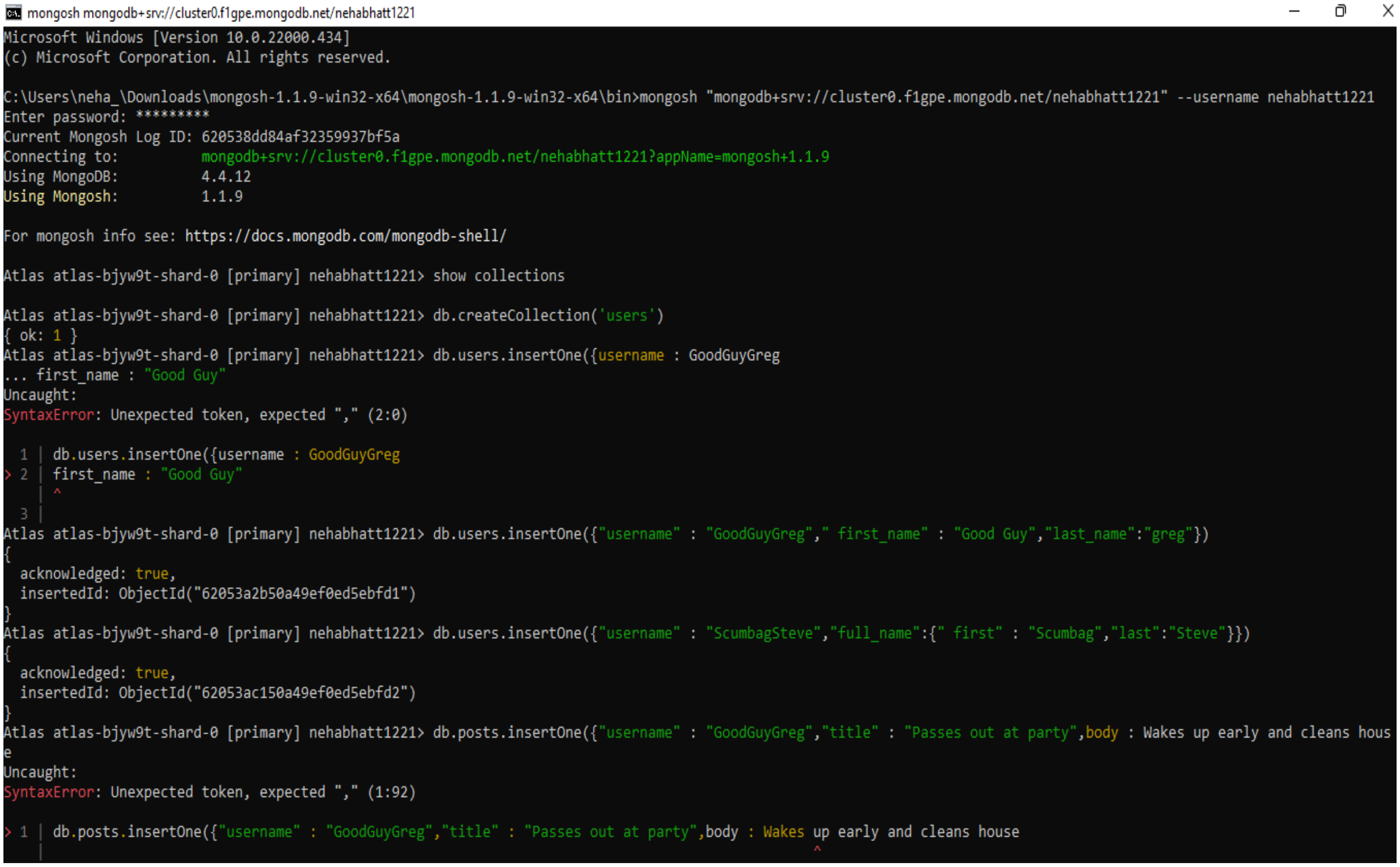
```
it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("620355ccf9a93ec7cf343de2"),
    title: 'The Hobbit:The Desolation of Smaug',
    ' writer': 'J.R.R. Tolekin',
    ' year': 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possess
ion of a mysterious and magical ring'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> db.MOvies.find({$and:[{synopsis:{$regex:"gold"}},{synopsis:{$regex:"dragon"}}]})
[
  {
    _id: ObjectId("62035527f9a93ec7cf343de1"),
    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 - from the dragon Smaug.'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] mongo_practice> _
```

Relationships

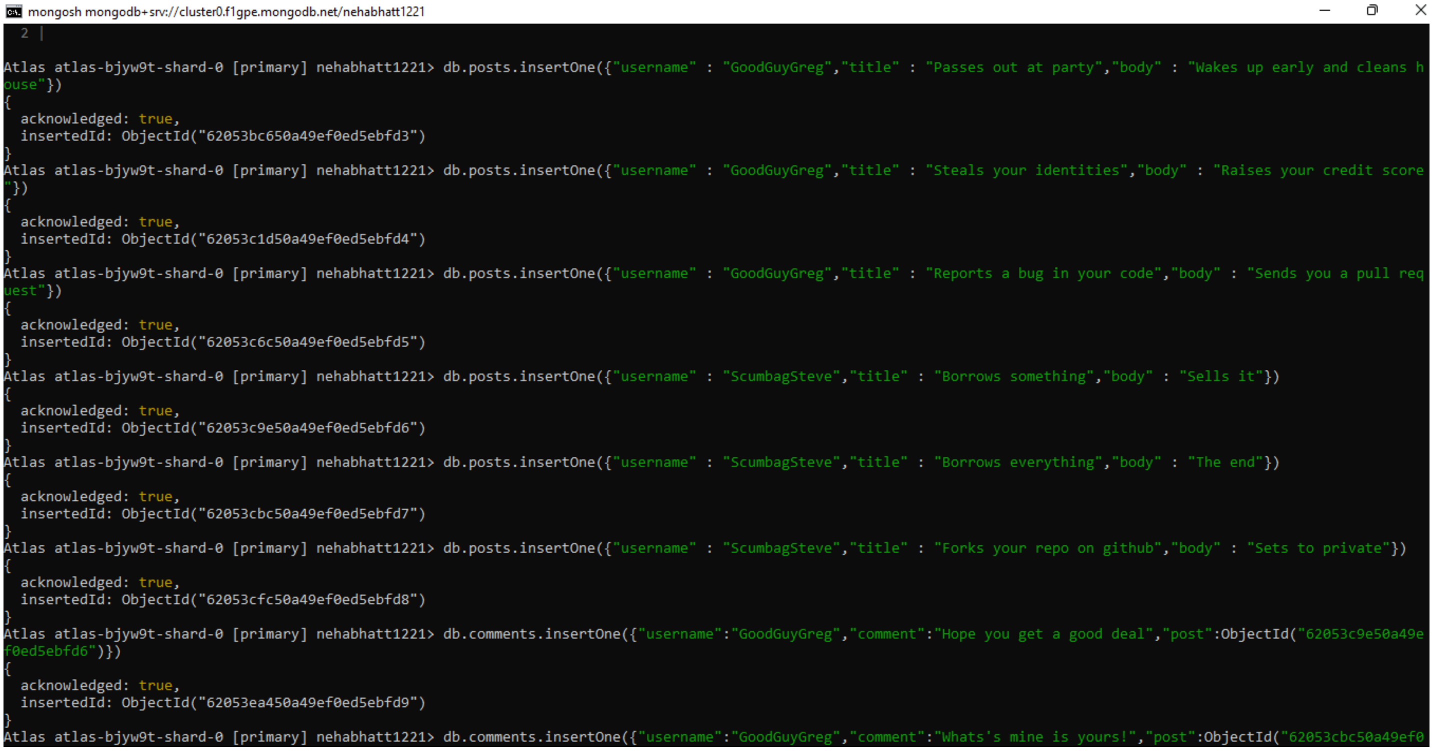
Insert the following documents into a users collection

Insert the following documents into a posts collection

Insert the following documents into a comments collection

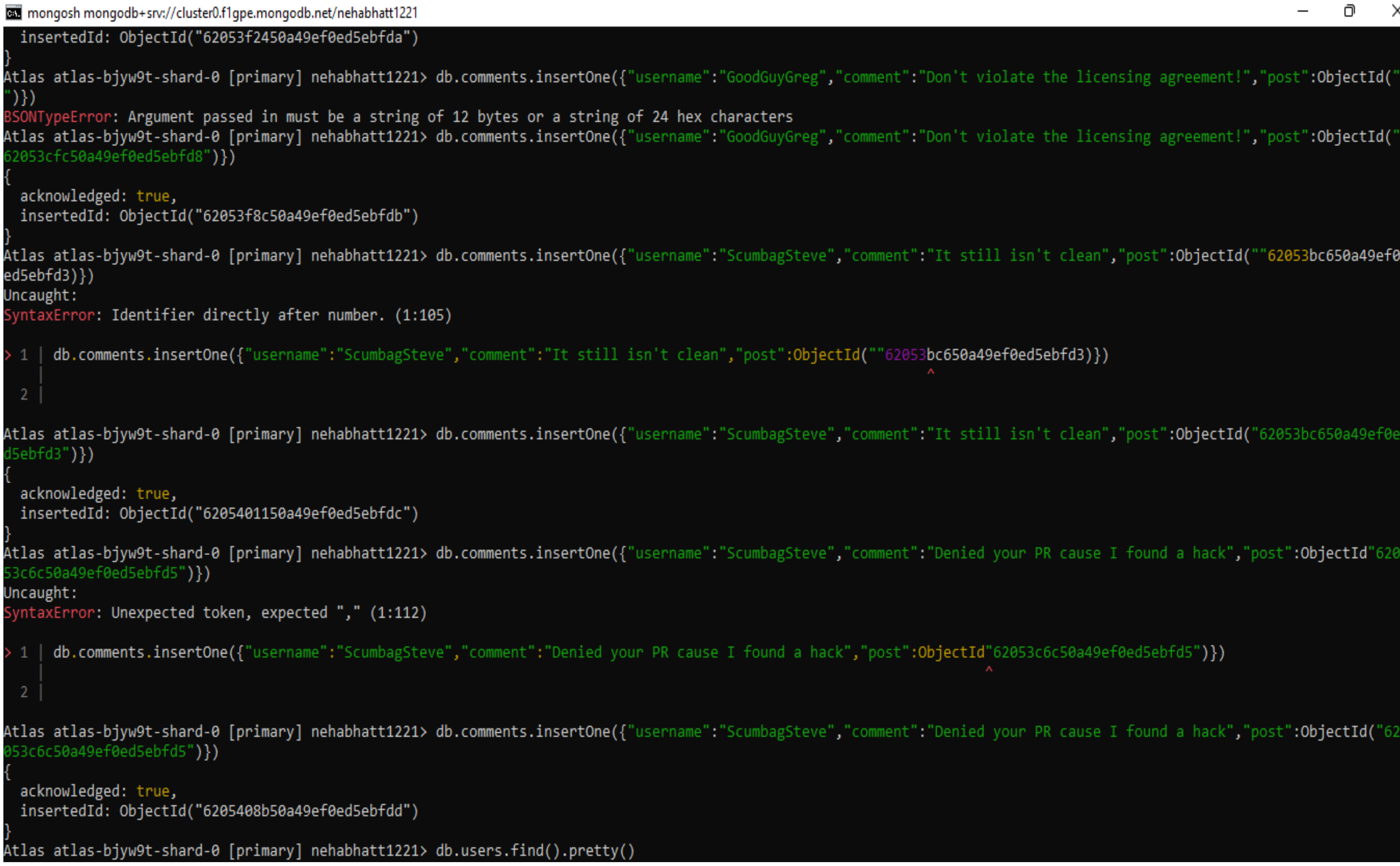

mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\neha_\Downloads\mongosh-1.1.9-win32-x64\mongosh-1.1.9-win32-x64\bin>mongosh "mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221" --username nehabhatt1221
Enter password: *********
Current Mongosh Log ID: 620538dd84af32359937bf5a
Connecting to:          mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221?appName=mongosh+1.1.9
Using MongoDB:          4.4.12
Using Mongosh:          1.1.9

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

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> show collections

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.createCollection('users')
{ ok: 1 }
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.users.insertOne({username : GoodGuyGreg
... first_name : "Good Guy"
Uncaught:
SyntaxError: Unexpected token, expected "," (2:0)

  1 | db.users.insertOne({username : GoodGuyGreg
> 2 | first_name : "Good Guy"
    | ^
  3 |
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.users.insertOne({"username" : "GoodGuyGreg"," first_name" : "Good Guy","last_name":"greg"})
{
  acknowledged: true,
  insertedId: ObjectId("62053a2b50a49ef0ed5ebfd1")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.users.insertOne({"username" : "ScumbagSteve","full_name":{" first" : "Scumbag","last":"Steve"}})
{
  acknowledged: true,
  insertedId: ObjectId("62053ac150a49ef0ed5ebfd2")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "GoodGuyGreg","title" : "Passes out at party",body : Wakes up early and cleans hous
e
Uncaught:
SyntaxError: Unexpected token, expected "," (1:92)

> 1 | db.posts.insertOne({"username" : "GoodGuyGreg","title" : "Passes out at party",body : Wakes up early and cleans house
    |                                                                                             ^
```



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

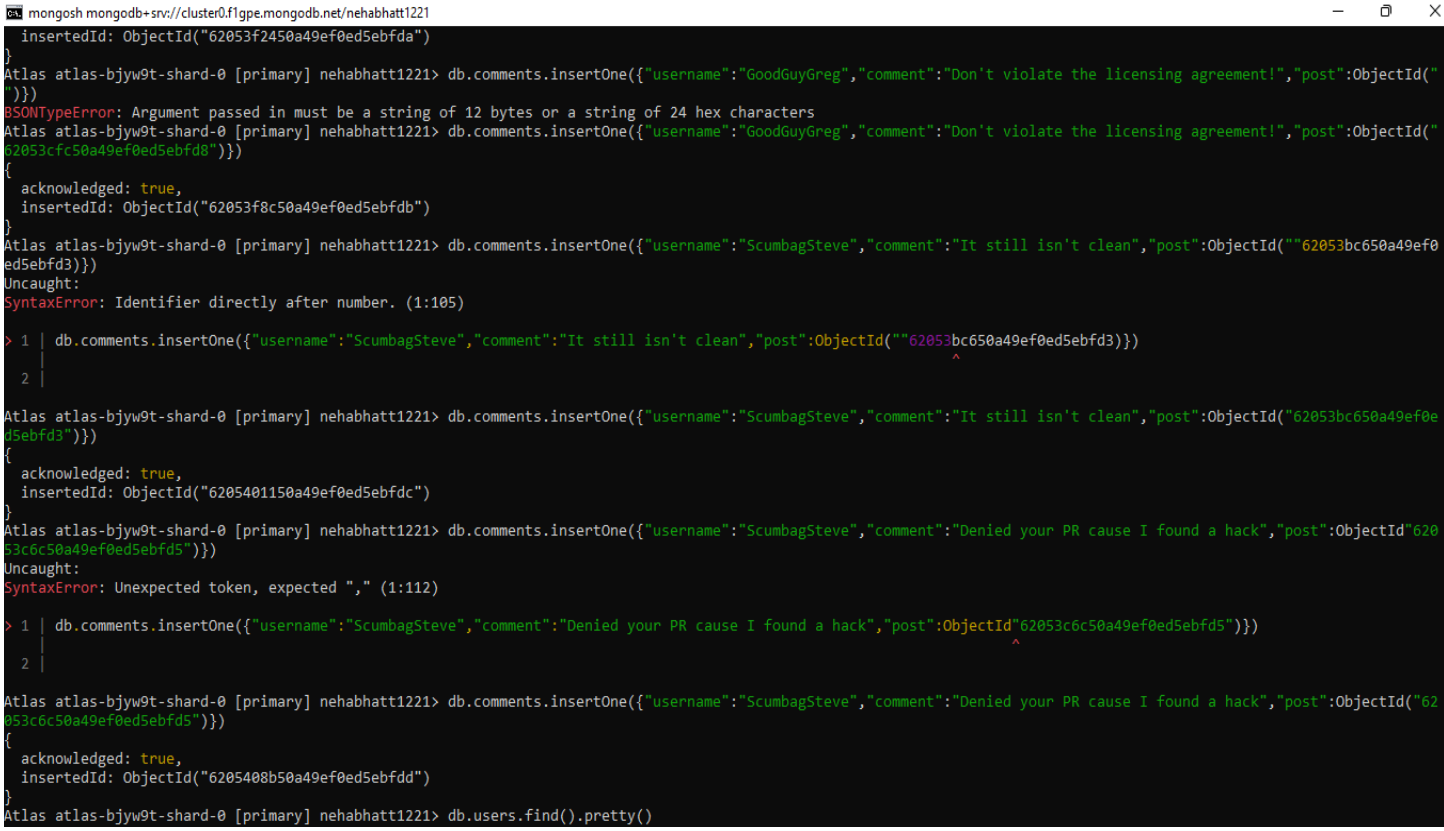
```
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "GoodGuyGreg","title" : "Passes out at party","body" : "Wakes up early and cleans h
ouse"})
{
  acknowledged: true,
  insertedId: ObjectId("62053bc650a49ef0ed5ebfd3")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "GoodGuyGreg","title" : "Steals your identities","body" : "Raises your credit score
"})
{
  acknowledged: true,
  insertedId: ObjectId("62053c1d50a49ef0ed5ebfd4")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "GoodGuyGreg","title" : "Reports a bug in your code","body" : "Sends you a pull req
uest"})
{
  acknowledged: true,
  insertedId: ObjectId("62053c6c50a49ef0ed5ebfd5")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "ScumbagSteve","title" : "Borrows something","body" : "Sells it"})
{
  acknowledged: true,
  insertedId: ObjectId("62053c9e50a49ef0ed5ebfd6")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "ScumbagSteve","title" : "Borrows everything","body" : "The end"})
{
  acknowledged: true,
  insertedId: ObjectId("62053cbc50a49ef0ed5ebfd7")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.insertOne({"username" : "ScumbagSteve","title" : "Forks your repo on github","body" : "Sets to private"})
{
  acknowledged: true,
  insertedId: ObjectId("62053cfc50a49ef0ed5ebfd8")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"Hope you get a good deal","post":ObjectId("62053c9e50a49e
f0ed5ebfd6")})
{
  acknowledged: true,
  insertedId: ObjectId("62053ea450a49ef0ed5ebfd9")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"What's mine is yours!","post":ObjectId("62053cbc50a49ef0
```



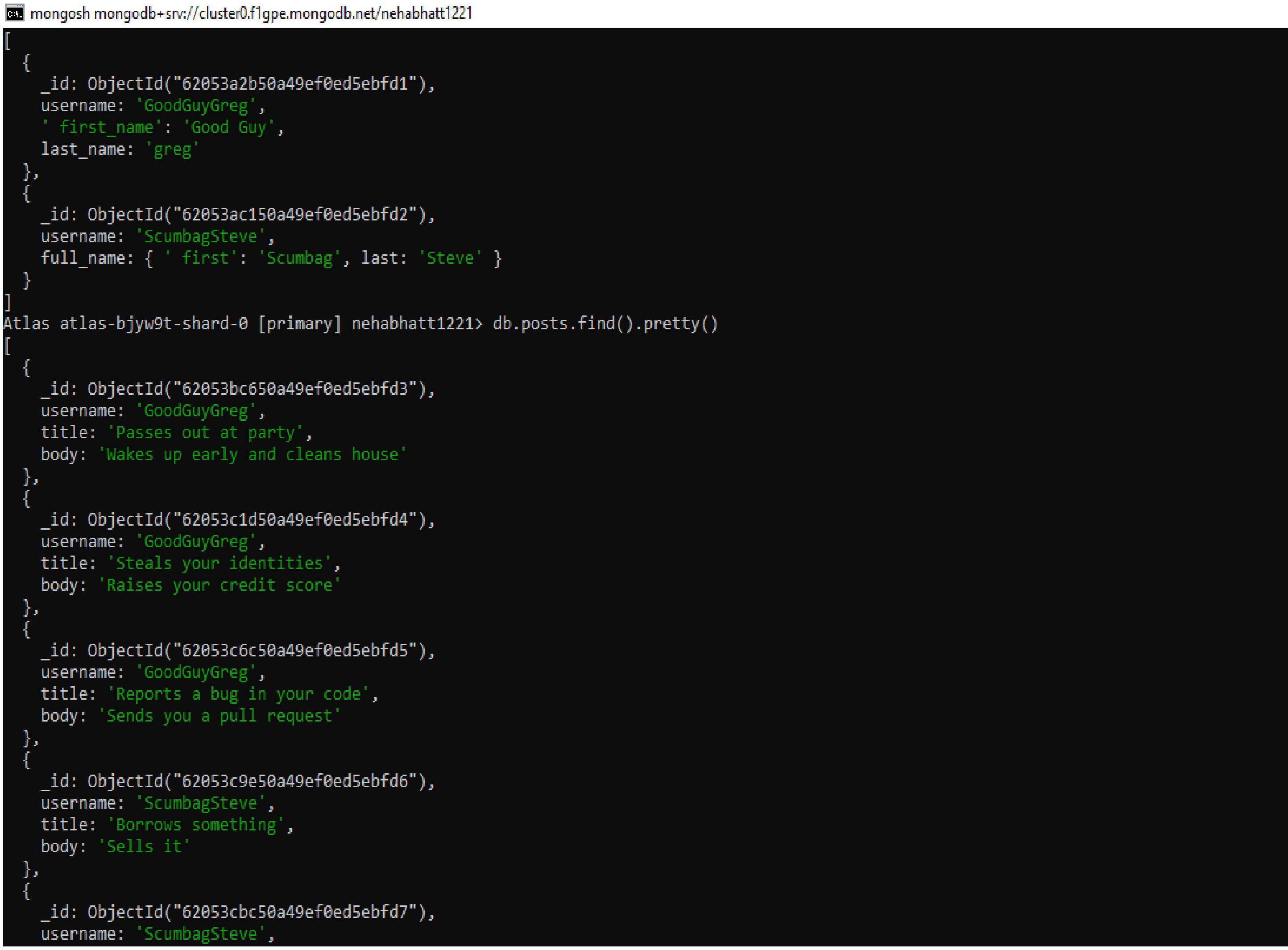
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  insertedId: ObjectId("62053f2450a49ef0ed5ebfda")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"Don't violate the licensing agreement!","post":ObjectId("
")})
BSONTypeError: Argument passed in must be a string of 12 bytes or a string of 24 hex characters
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"Don't violate the licensing agreement!","post":ObjectId("
62053cfc50a49ef0ed5ebfd8")})
{
  acknowledged: true,
  insertedId: ObjectId("62053f8c50a49ef0ed5ebfdb")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId(""62053bc650a49ef0
ed5ebfd3)})
Uncaught:
SyntaxError: Identifier directly after number. (1:105)

> 1 | db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId(""62053bc650a49ef0ed5ebfd3)})
    |                                                                                                          ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId("62053bc650a49ef0e
d5ebfd3")})
{
  acknowledged: true,
  insertedId: ObjectId("6205401150a49ef0ed5ebfdc")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId"620
53c6c50a49ef0ed5ebfd5")})
Uncaught:
SyntaxError: Unexpected token, expected "," (1:112)

> 1 | db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId"62053c6c50a49ef0ed5ebfd5")})
    |                                                                                                                 ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId("62
053c6c50a49ef0ed5ebfd5")})
{
  acknowledged: true,
  insertedId: ObjectId("6205408b50a49ef0ed5ebfdd")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.users.find().pretty()
```

Querying related collections

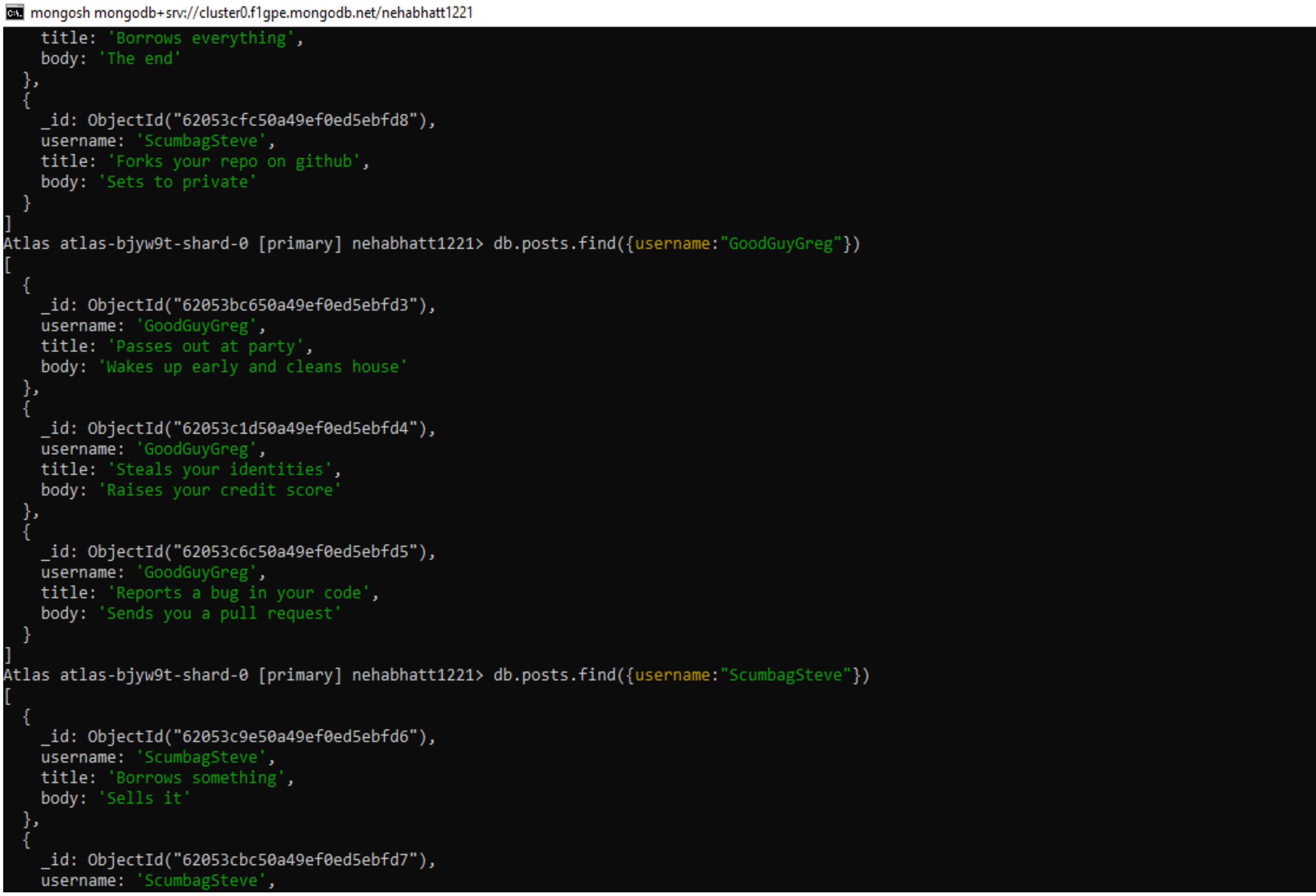

mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  insertedId: ObjectId("62053f2450a49ef0ed5ebfda")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"Don't violate the licensing agreement!","post":ObjectId("
")})
BSONTypeError: Argument passed in must be a string of 12 bytes or a string of 24 hex characters
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"GoodGuyGreg","comment":"Don't violate the licensing agreement!","post":ObjectId("
62053cfc50a49ef0ed5ebfd8")})
{
  acknowledged: true,
  insertedId: ObjectId("62053f8c50a49ef0ed5ebfdb")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId(""62053bc650a49ef0
ed5ebfd3)})
Uncaught:
SyntaxError: Identifier directly after number. (1:105)

> 1 | db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId(""62053bc650a49ef0ed5ebfd3)})
    |                                                                                                          ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId("62053bc650a49ef0e
d5ebfd3")})
{
  acknowledged: true,
  insertedId: ObjectId("6205401150a49ef0ed5ebfdc")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId"620
53c6c50a49ef0ed5ebfd5")})
Uncaught:
SyntaxError: Unexpected token, expected "," (1:112)

> 1 | db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId"62053c6c50a49ef0ed5ebfd5")})
    |                                                                                                                 ^
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.insertOne({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId("62
053c6c50a49ef0ed5ebfd5")})
{
  acknowledged: true,
  insertedId: ObjectId("6205408b50a49ef0ed5ebfdd")
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.users.find().pretty()
```



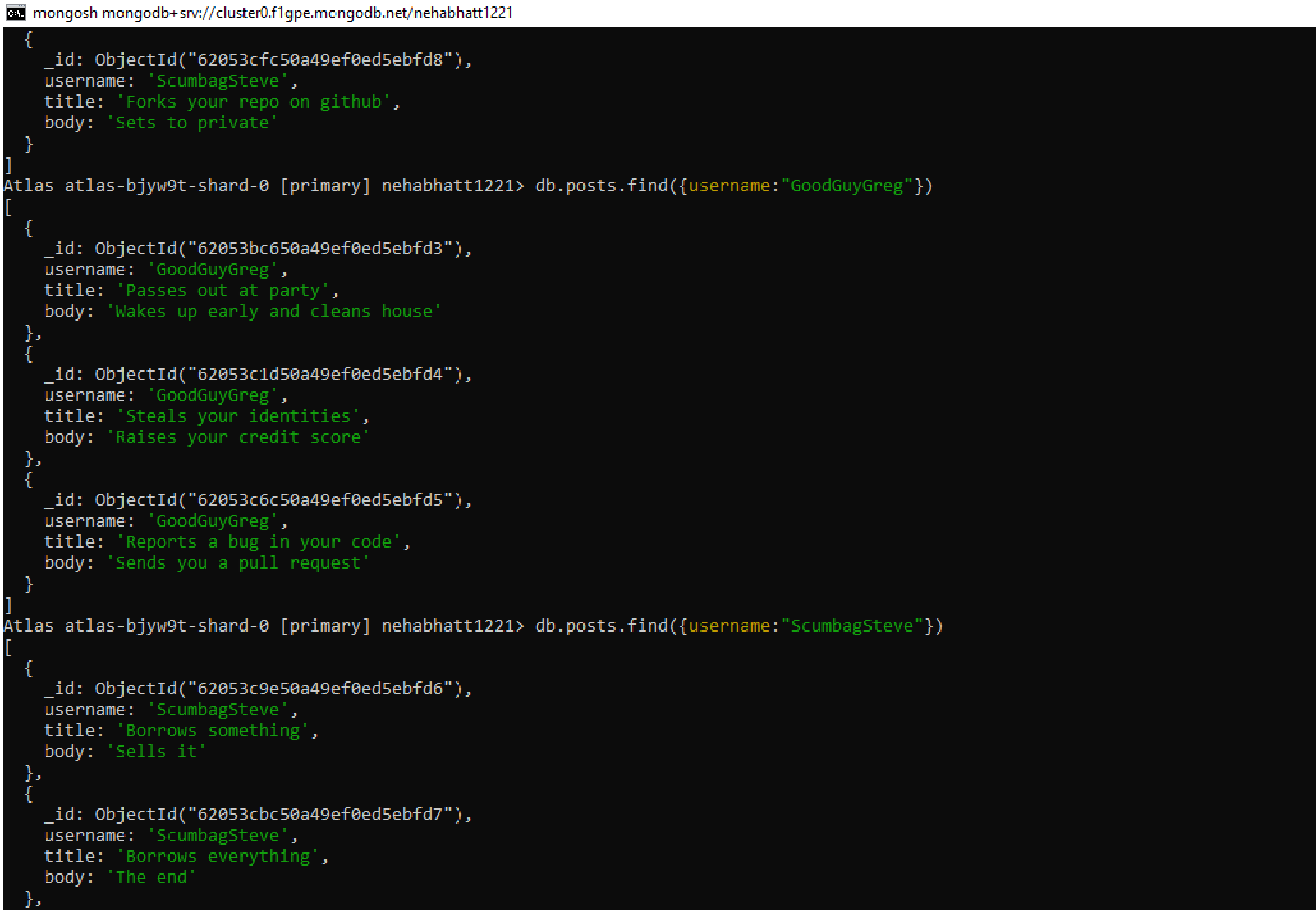
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
[
  {
    _id: ObjectId("62053a2b50a49ef0ed5ebfd1"),
    username: 'GoodGuyGreg',
    ' first_name': 'Good Guy',
    last_name: 'greg'
  },
  {
    _id: ObjectId("62053ac150a49ef0ed5ebfd2"),
    username: 'ScumbagSteve',
    full_name: { ' first': 'Scumbag', last: 'Steve' }
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.find().pretty()
[
  {
    _id: ObjectId("62053bc650a49ef0ed5ebfd3"),
    username: 'GoodGuyGreg',
    title: 'Passes out at party',
    body: 'Wakes up early and cleans house'
  },
  {
    _id: ObjectId("62053c1d50a49ef0ed5ebfd4"),
    username: 'GoodGuyGreg',
    title: 'Steals your identities',
    body: 'Raises your credit score'
  },
  {
    _id: ObjectId("62053c6c50a49ef0ed5ebfd5"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request'
  },
  {
    _id: ObjectId("62053c9e50a49ef0ed5ebfd6"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it'
  },
  {
    _id: ObjectId("62053cbc50a49ef0ed5ebfd7"),
    username: 'ScumbagSteve',
```



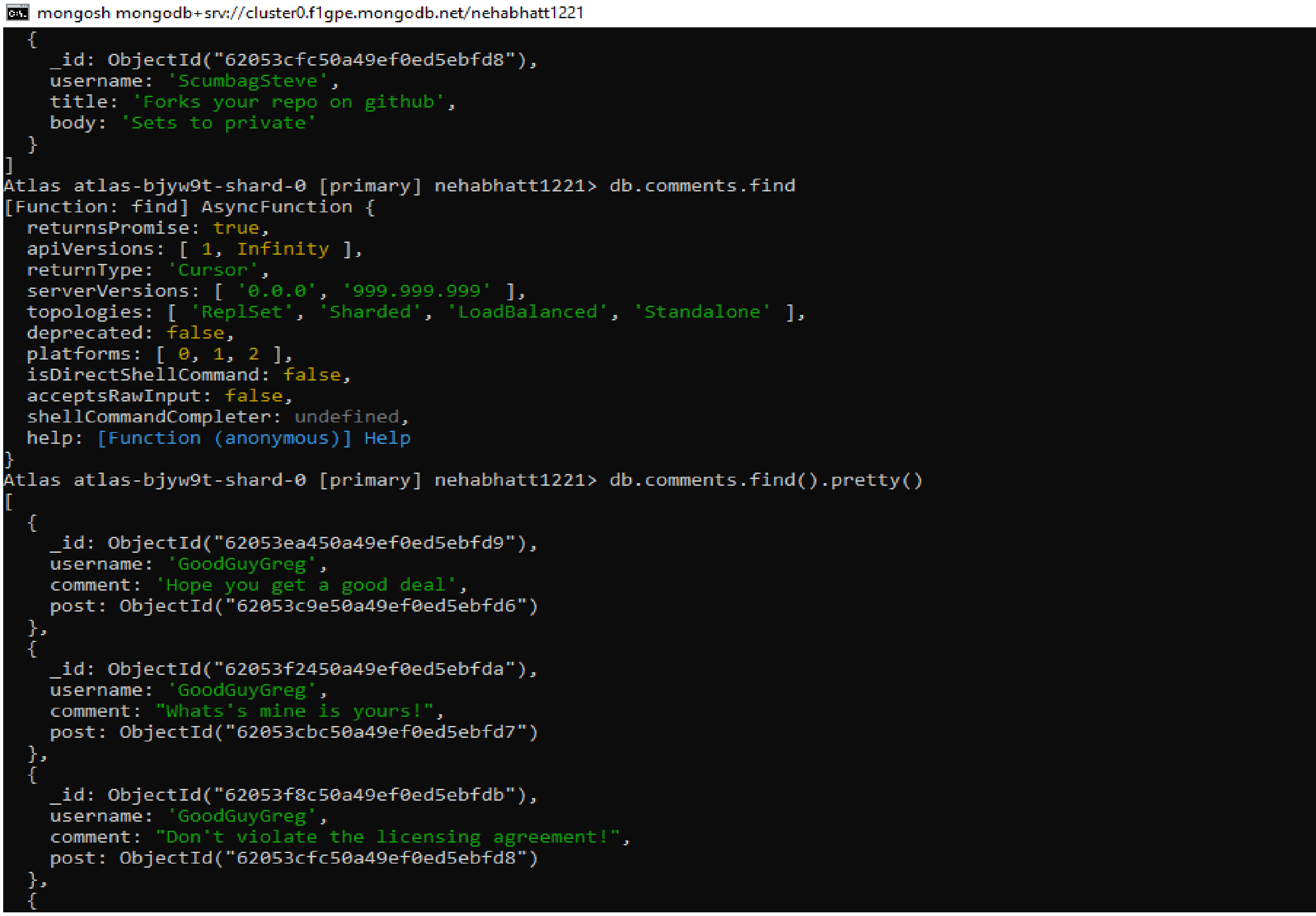
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
    title: 'Borrows everything',
    body: 'The end'
  },
  {
    _id: ObjectId("62053cfc50a49ef0ed5ebfd8"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.find({username:"GoodGuyGreg"})
[
  {
    _id: ObjectId("62053bc650a49ef0ed5ebfd3"),
    username: 'GoodGuyGreg',
    title: 'Passes out at party',
    body: 'Wakes up early and cleans house'
  },
  {
    _id: ObjectId("62053c1d50a49ef0ed5ebfd4"),
    username: 'GoodGuyGreg',
    title: 'Steals your identities',
    body: 'Raises your credit score'
  },
  {
    _id: ObjectId("62053c6c50a49ef0ed5ebfd5"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.find({username:"ScumbagSteve"})
[
  {
    _id: ObjectId("62053c9e50a49ef0ed5ebfd6"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it'
  },
  {
    _id: ObjectId("62053cbc50a49ef0ed5ebfd7"),
    username: 'ScumbagSteve',
```



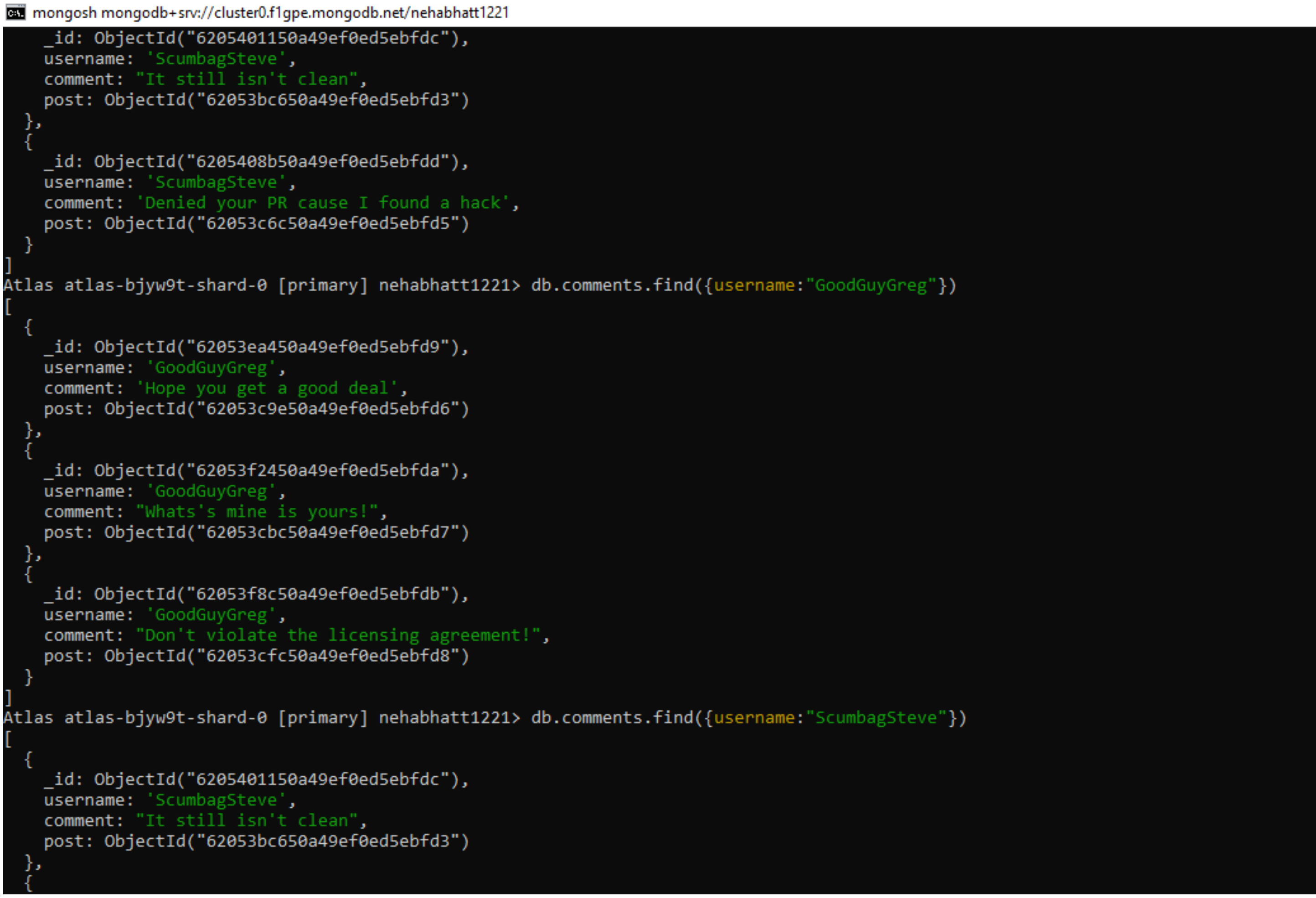
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  {
    _id: ObjectId("62053cfc50a49ef0ed5ebfd8"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.find({username:"GoodGuyGreg"})
[
  {
    _id: ObjectId("62053bc650a49ef0ed5ebfd3"),
    username: 'GoodGuyGreg',
    title: 'Passes out at party',
    body: 'Wakes up early and cleans house'
  },
  {
    _id: ObjectId("62053c1d50a49ef0ed5ebfd4"),
    username: 'GoodGuyGreg',
    title: 'Steals your identities',
    body: 'Raises your credit score'
  },
  {
    _id: ObjectId("62053c6c50a49ef0ed5ebfd5"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.posts.find({username:"ScumbagSteve"})
[
  {
    _id: ObjectId("62053c9e50a49ef0ed5ebfd6"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it'
  },
  {
    _id: ObjectId("62053cbc50a49ef0ed5ebfd7"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end'
  },
```

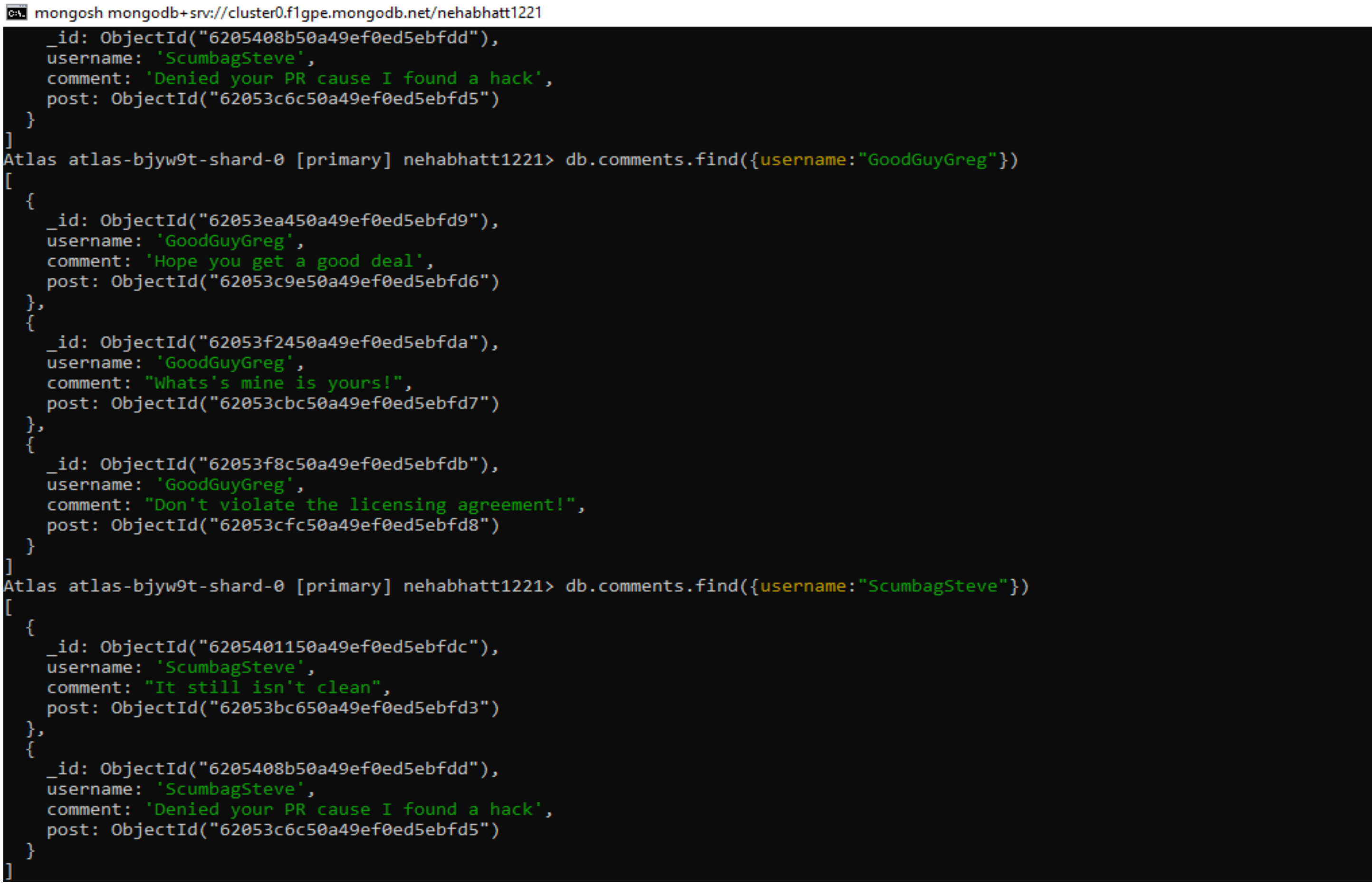


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  {
    _id: ObjectId("62053cfc50a49ef0ed5ebfd8"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find
[Function: find] AsyncFunction {
  returnsPromise: true,
  apiVersions: [ 1, Infinity ],
  returnType: 'Cursor',
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find().pretty()
[
  {
    _id: ObjectId("62053ea450a49ef0ed5ebfd9"),
    username: 'GoodGuyGreg',
    comment: 'Hope you get a good deal',
    post: ObjectId("62053c9e50a49ef0ed5ebfd6")
  },
  {
    _id: ObjectId("62053f2450a49ef0ed5ebfda"),
    username: 'GoodGuyGreg',
    comment: "Whats's mine is yours!",
    post: ObjectId("62053cbc50a49ef0ed5ebfd7")
  },
  {
    _id: ObjectId("62053f8c50a49ef0ed5ebfdb"),
    username: 'GoodGuyGreg',
    comment: "Don't violate the licensing agreement!",
    post: ObjectId("62053cfc50a49ef0ed5ebfd8")
  },
  {
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
    _id: ObjectId("6205401150a49ef0ed5ebfdc"),
    username: 'ScumbagSteve',
    comment: "It still isn't clean",
    post: ObjectId("62053bc650a49ef0ed5ebfd3")
  },
  {
    _id: ObjectId("6205408b50a49ef0ed5ebfdd"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("62053c6c50a49ef0ed5ebfd5")
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find({username:"GoodGuyGreg"})
[
  {
    _id: ObjectId("62053ea450a49ef0ed5ebfd9"),
    username: 'GoodGuyGreg',
    comment: 'Hope you get a good deal',
    post: ObjectId("62053c9e50a49ef0ed5ebfd6")
  },
  {
    _id: ObjectId("62053f2450a49ef0ed5ebfda"),
    username: 'GoodGuyGreg',
    comment: "Whats's mine is yours!",
    post: ObjectId("62053cbc50a49ef0ed5ebfd7")
  },
  {
    _id: ObjectId("62053f8c50a49ef0ed5ebfdb"),
    username: 'GoodGuyGreg',
    comment: "Don't violate the licensing agreement!",
    post: ObjectId("62053cfc50a49ef0ed5ebfd8")
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find({username:"ScumbagSteve"})
[
  {
    _id: ObjectId("6205401150a49ef0ed5ebfdc"),
    username: 'ScumbagSteve',
    comment: "It still isn't clean",
    post: ObjectId("62053bc650a49ef0ed5ebfd3")
  },
  {
```


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
    _id: ObjectId("6205408b50a49ef0ed5ebfdd"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("62053c6c50a49ef0ed5ebfd5")
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find({username:"GoodGuyGreg"})
[
  {
    _id: ObjectId("62053ea450a49ef0ed5ebfd9"),
    username: 'GoodGuyGreg',
    comment: 'Hope you get a good deal',
    post: ObjectId("62053c9e50a49ef0ed5ebfd6")
  },
  {
    _id: ObjectId("62053f2450a49ef0ed5ebfda"),
    username: 'GoodGuyGreg',
    comment: "Whats's mine is yours!",
    post: ObjectId("62053cbc50a49ef0ed5ebfd7")
  },
  {
    _id: ObjectId("62053f8c50a49ef0ed5ebfdb"),
    username: 'GoodGuyGreg',
    comment: "Don't violate the licensing agreement!",
    post: ObjectId("62053cfc50a49ef0ed5ebfd8")
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> db.comments.find({username:"ScumbagSteve"})
[
  {
    _id: ObjectId("6205401150a49ef0ed5ebfdc"),
    username: 'ScumbagSteve',
    comment: "It still isn't clean",
    post: ObjectId("62053bc650a49ef0ed5ebfd3")
  },
  {
    _id: ObjectId("6205408b50a49ef0ed5ebfdd"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("62053c6c50a49ef0ed5ebfd5")
  }
]
```