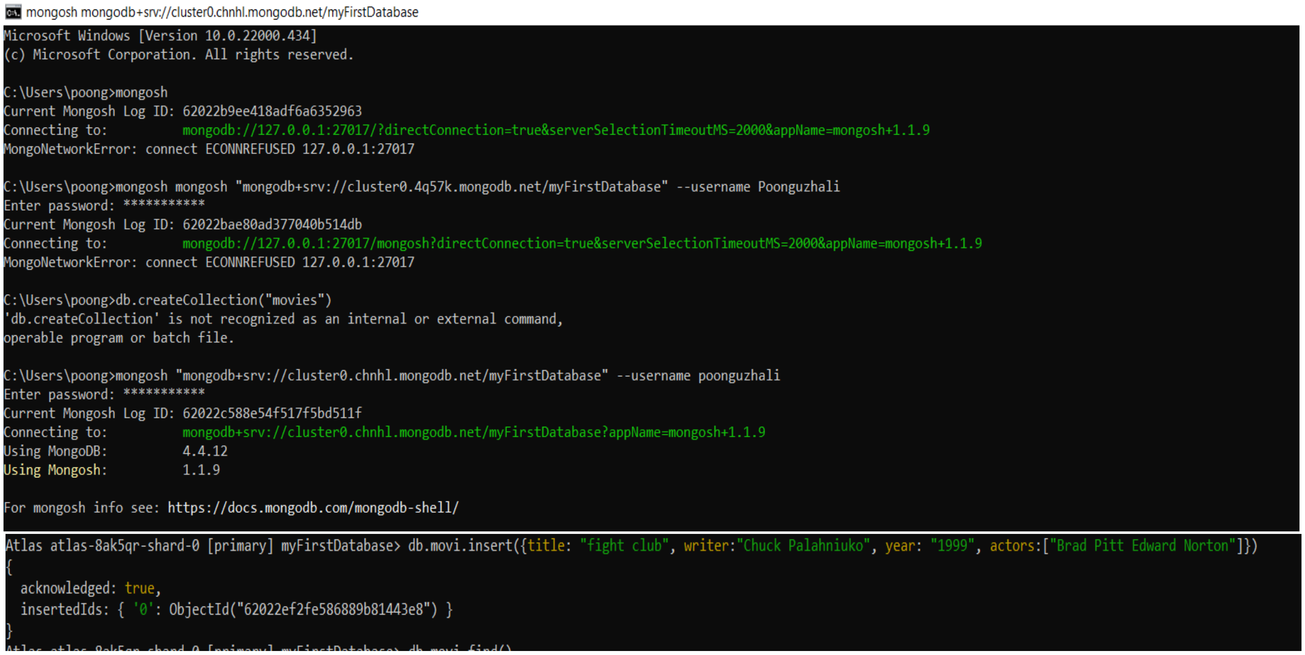
Poonguzhali A

MongoDB-Assignments1

MongoDB Exercise in mongo shell

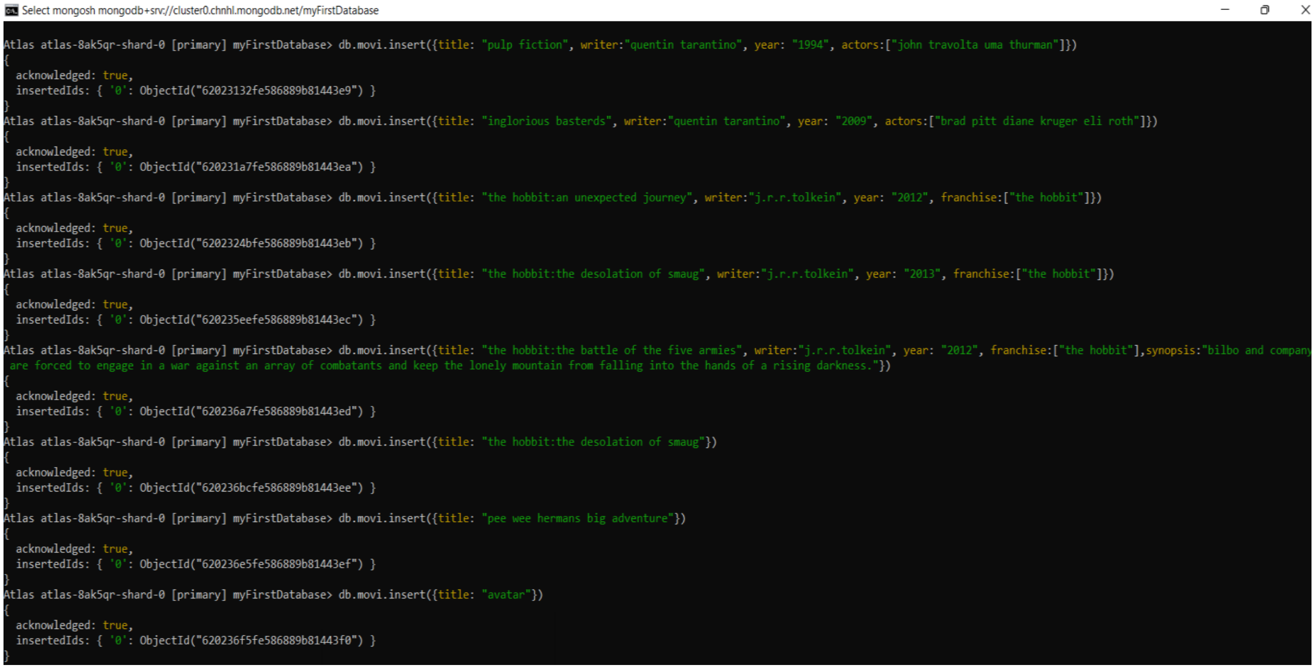
Insert Documents

Insert the following documents into a **movies** collection.


mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

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

C:\Users\poong>mongosh
Current Mongosh Log ID: 62022b9ee418adf6a6352963
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.1.9
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017

C:\Users\poong>mongosh mongosh "mongodb+srv://cluster0.4q57k.mongodb.net/myFirstDatabase" --username Poonguzhali
Enter password: ***********
Current Mongosh Log ID: 62022bae80ad377040b514db
Connecting to:          mongodb://127.0.0.1:27017/mongosh?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.1.9
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017

C:\Users\poong>db.createCollection("movies")
'db.createCollection' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\poong>mongosh "mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase" --username poonguzhali
Enter password: ***********
Current Mongosh Log ID: 62022c588e54f517f5bd511f
Connecting to:          mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase?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-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "fight club", writer:"Chuck Palahniuko", year: "1999", actors:["Brad Pitt Edward Norton"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62022ef2fe586889b81443e8") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find()
```

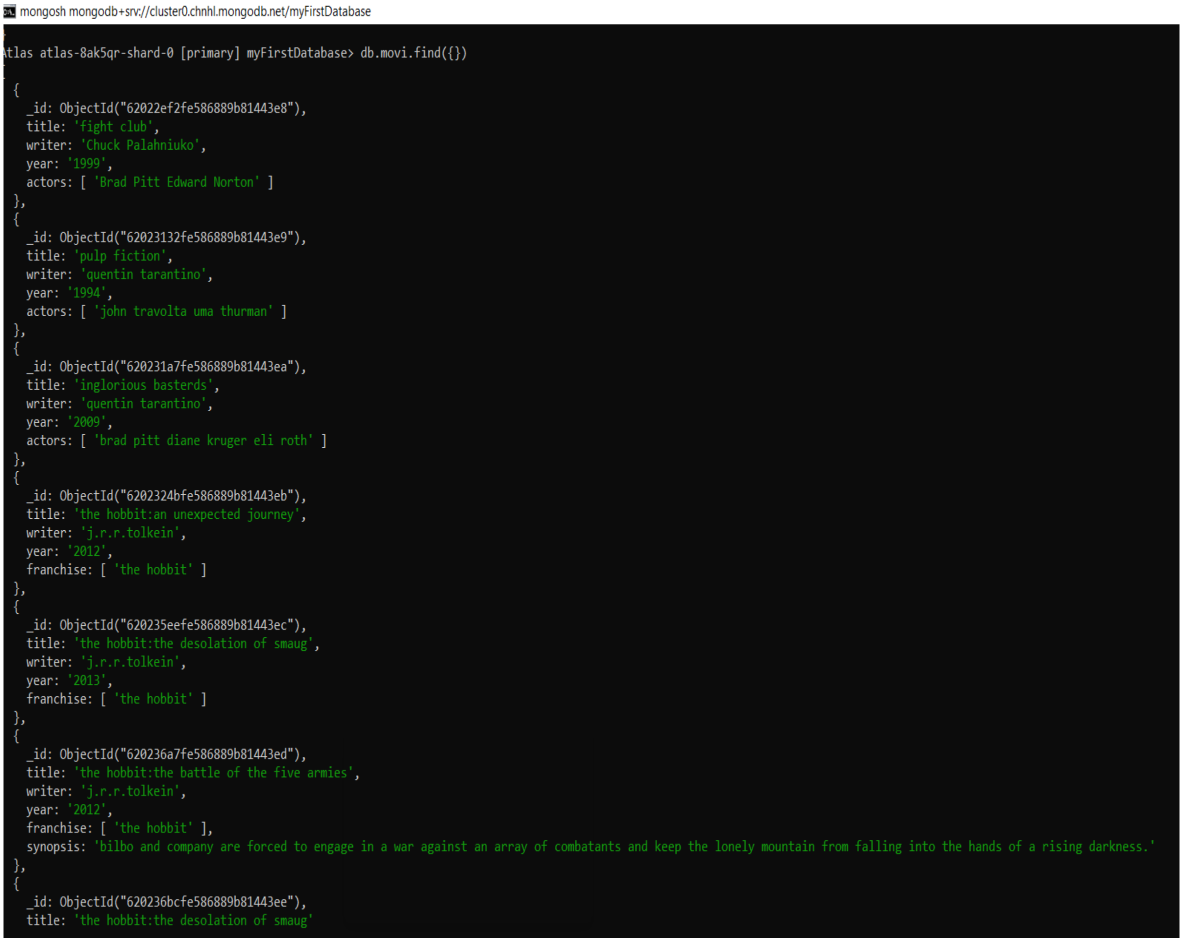

Select mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "pulp fiction", writer:"quentin tarantino", year: "1994", actors:["john travolta uma thurman"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62023132fe586889b81443e9") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "inglorious basterds", writer:"quentin tarantino", year: "2009", actors:["brad pitt diane kruger eli roth"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620231a7fe586889b81443ea") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "the hobbit:an unexpected journey", writer:"j.r.r.tolkein", year: "2012", franchise:["the hobbit"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6202324bfe586889b81443eb") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "the hobbit:the desolation of smaug", writer:"j.r.r.tolkein", year: "2013", franchise:["the hobbit"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620235eefe586889b81443ec") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({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 rising darkness."})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620236a7fe586889b81443ed") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "the hobbit:the desolation of smaug"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620236bcfe586889b81443ee") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "pee wee hermans big adventure"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620236e5fe586889b81443ef") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.insert({title: "avatar"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620236f5fe586889b81443f0") }
}
```

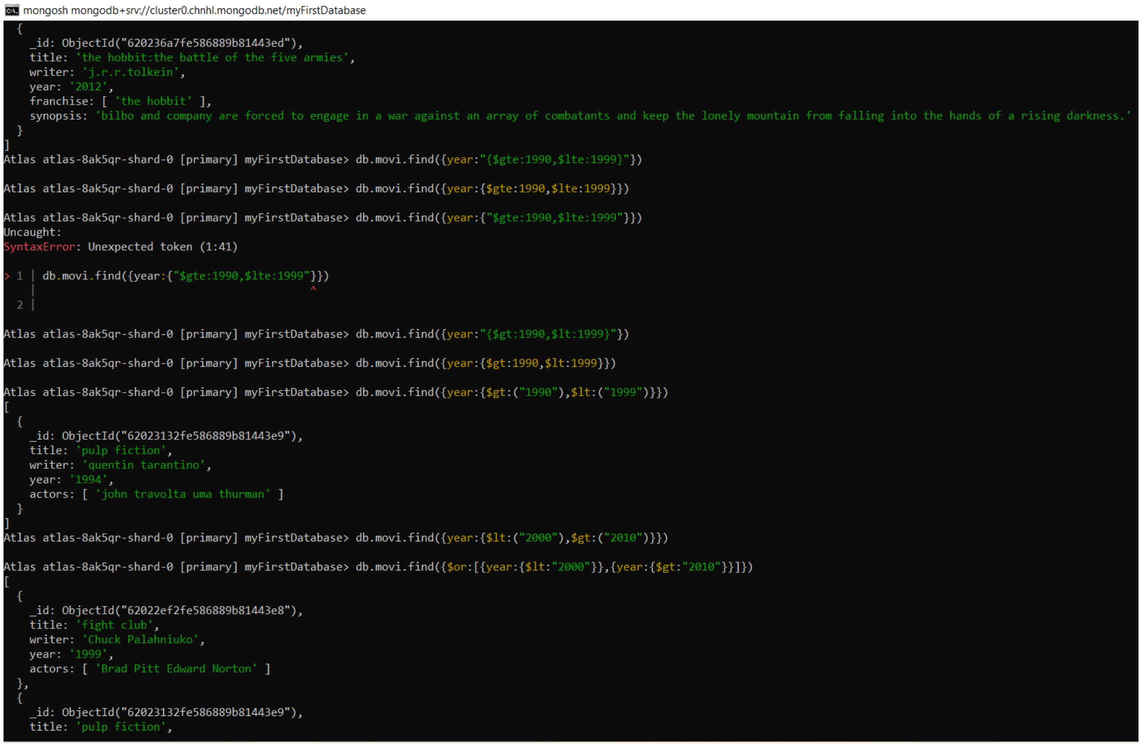
Query / Find Documents

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

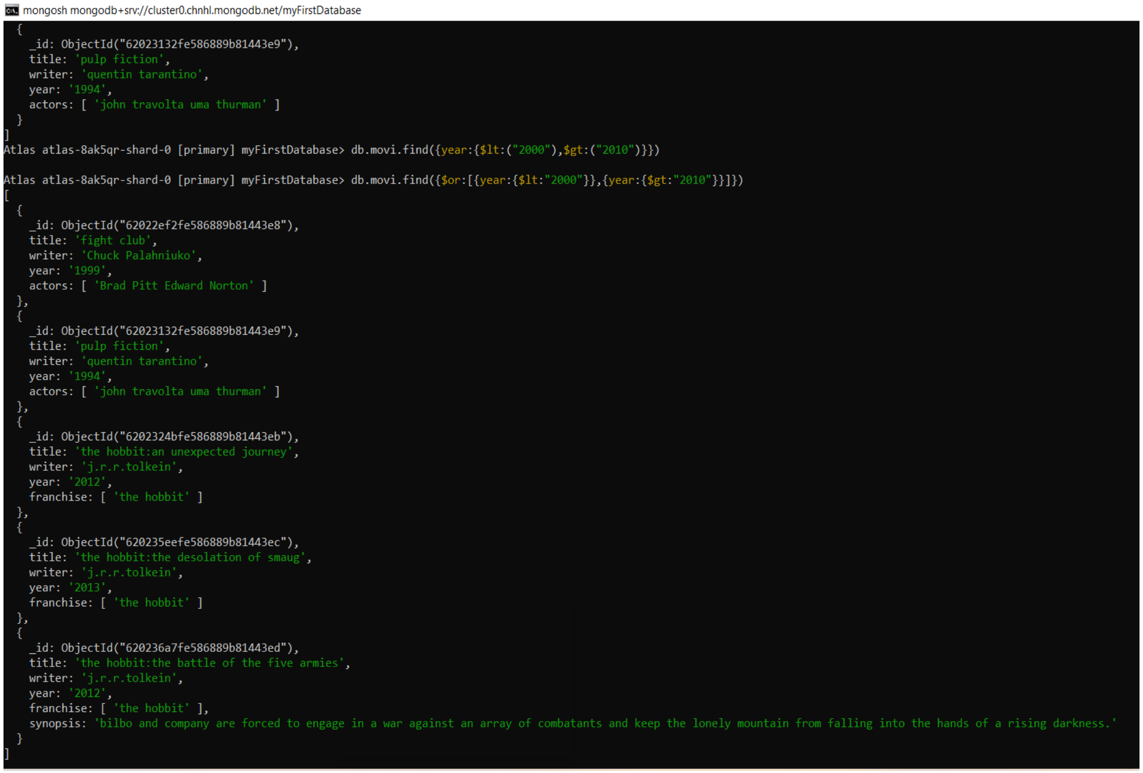

mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({})

  {
    _id: ObjectId("62022ef2fe586889b81443e8"),
    title: 'fight club',
    writer: 'Chuck Palahniuko',
    year: '1999',
    actors: [ 'Brad Pitt Edward Norton' ]
  },
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
    writer: 'quentin tarantino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  },
  {
    _id: ObjectId("620231a7fe586889b81443ea"),
    title: 'inglorious basterds',
    writer: 'quentin tarantino',
    year: '2009',
    actors: [ 'brad pitt diane kruger eli roth' ]
  },
  {
    _id: ObjectId("6202324bfe586889b81443eb"),
    title: 'the hobbit:an unexpected journey',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("620235eefe586889b81443ec"),
    title: 'the hobbit:the desolation of smaug',
    writer: 'j.r.r.tolkein',
    year: '2013',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("620236a7fe586889b81443ed"),
    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 rising darkness.'
  },
  {
    _id: ObjectId("620236bcfe586889b81443ee"),
    title: 'the hobbit:the desolation of smaug'
```



mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
  {
    _id: ObjectId("620236a7fe586889b81443ed"),
    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 rising darkness.'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:"{$gte:1990,$lte:1999}"})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{$gte:1990,$lte:1999}})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{"$gte:1990,$lte:1999"}})
Uncaught:
SyntaxError: Unexpected token (1:41)

> 1 | db.movi.find({year:{"$gte:1990,$lte:1999"}})
    |                                          ^
  2 |

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:"{$gt:1990,$lt:1999}"})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{$gt:1990,$lt:1999}})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{$gt:("1990"),$lt:("1999")}})
[
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
    writer: 'quentin tarantino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{$lt:("2000"),$gt:("2010")}})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$or:[{year:{$lt:"2000"}},{year:{$gt:"2010"}}]})
[
  {
    _id: ObjectId("62022ef2fe586889b81443e8"),
    title: 'fight club',
    writer: 'Chuck Palahniuko',
    year: '1999',
    actors: [ 'Brad Pitt Edward Norton' ]
  },
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
```

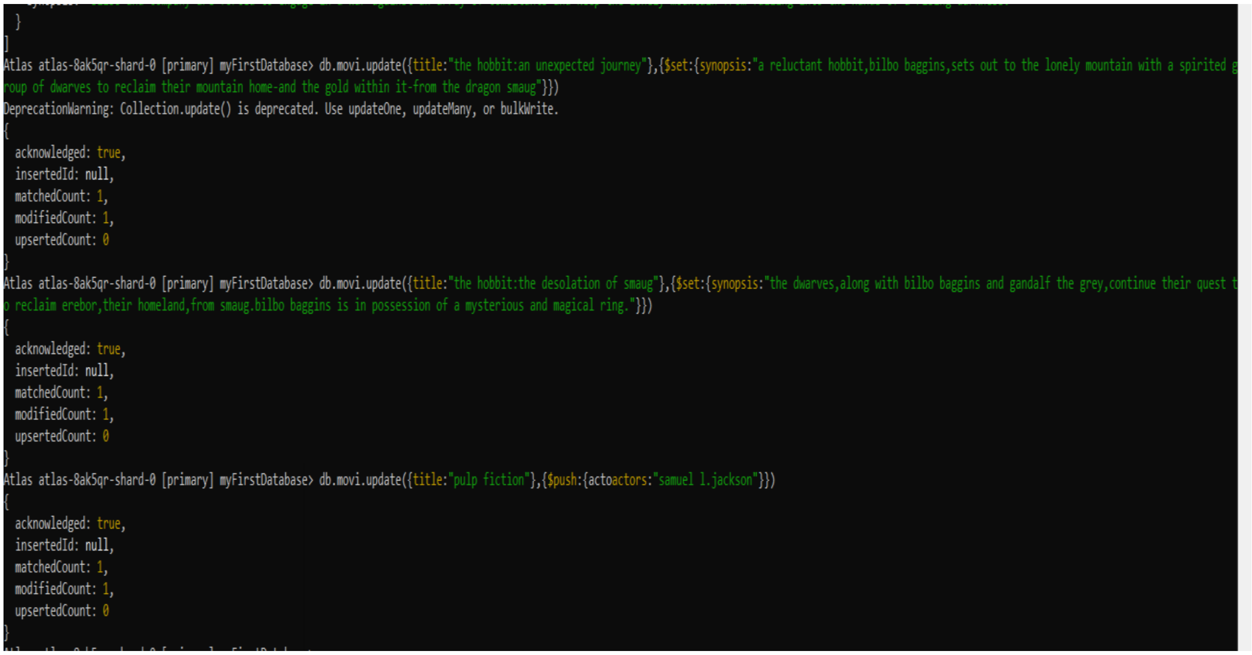


mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
    writer: 'quentin tarantino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({year:{$lt:("2000"),$gt:("2010")}})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$or:[{year:{$lt:"2000"}},{year:{$gt:"2010"}}]})
[
  {
    _id: ObjectId("62022ef2fe586889b81443e8"),
    title: 'fight club',
    writer: 'Chuck Palahniuko',
    year: '1999',
    actors: [ 'Brad Pitt Edward Norton' ]
  },
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
    writer: 'quentin tarantino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  },
  {
    _id: ObjectId("6202324bfe586889b81443eb"),
    title: 'the hobbit:an unexpected journey',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("620235eefe586889b81443ec"),
    title: 'the hobbit:the desolation of smaug',
    writer: 'j.r.r.tolkein',
    year: '2013',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("620236a7fe586889b81443ed"),
    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 rising darkness.'
  }
]
```

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"

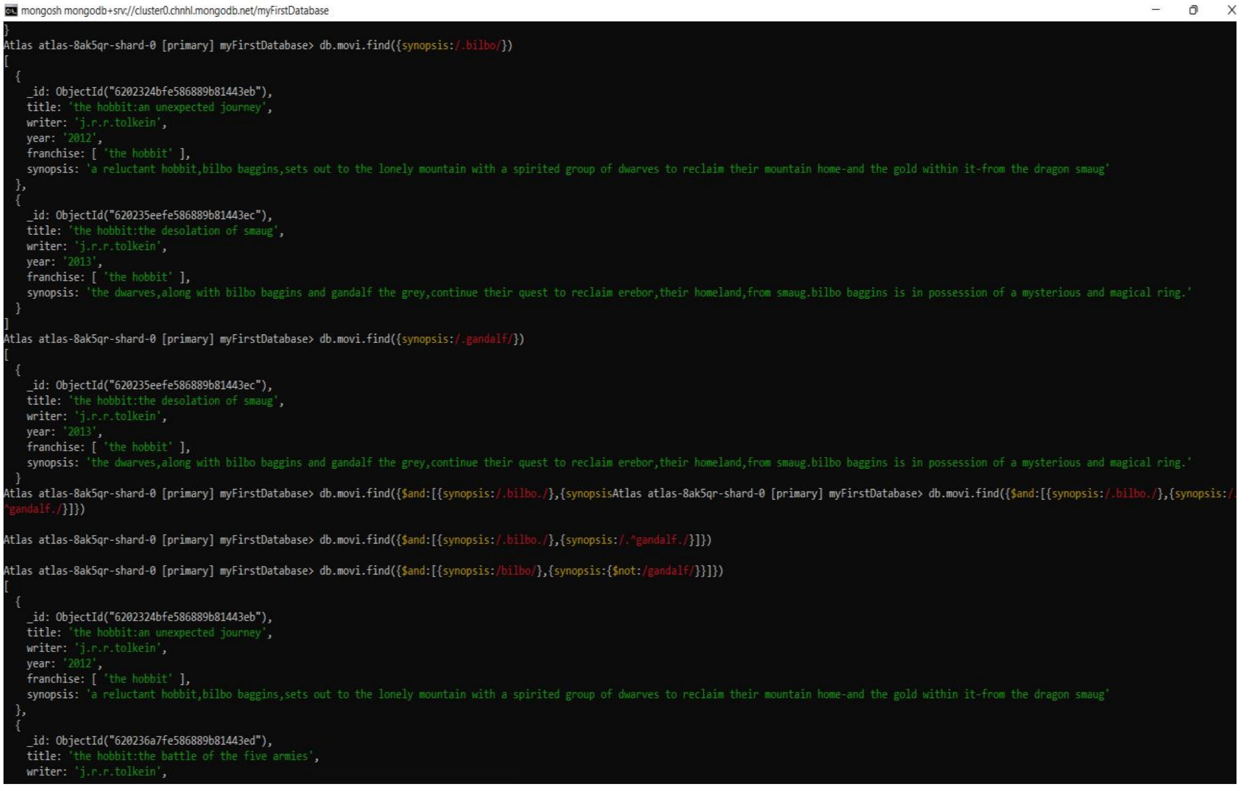


```
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.update({title:"the hobbit:an unexpected journey"},{$set:{synopsis:"a reluctant hobbit,bilbo baggins,sets out to the lonely mountain with a spirited g
roup of dwarves to reclaim their mountain home-and the gold within it-from the dragon smaug"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.update({title:"the hobbit:the desolation of smaug"},{$set:{synopsis:"the dwarves,along with bilbo baggins and gandalf the grey,continue their quest t
o 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
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.update({title:"pulp fiction"},{$push:{actoactors:"samuel l.jackson"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

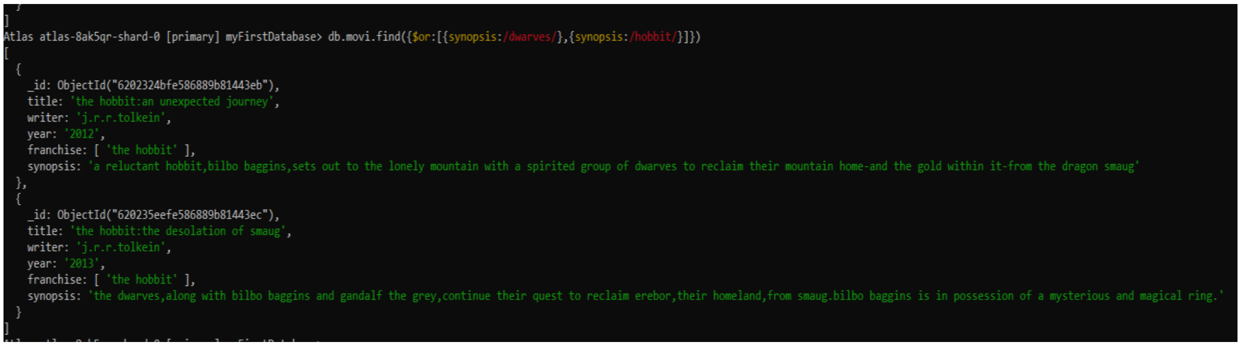
Text Search

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"

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


mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

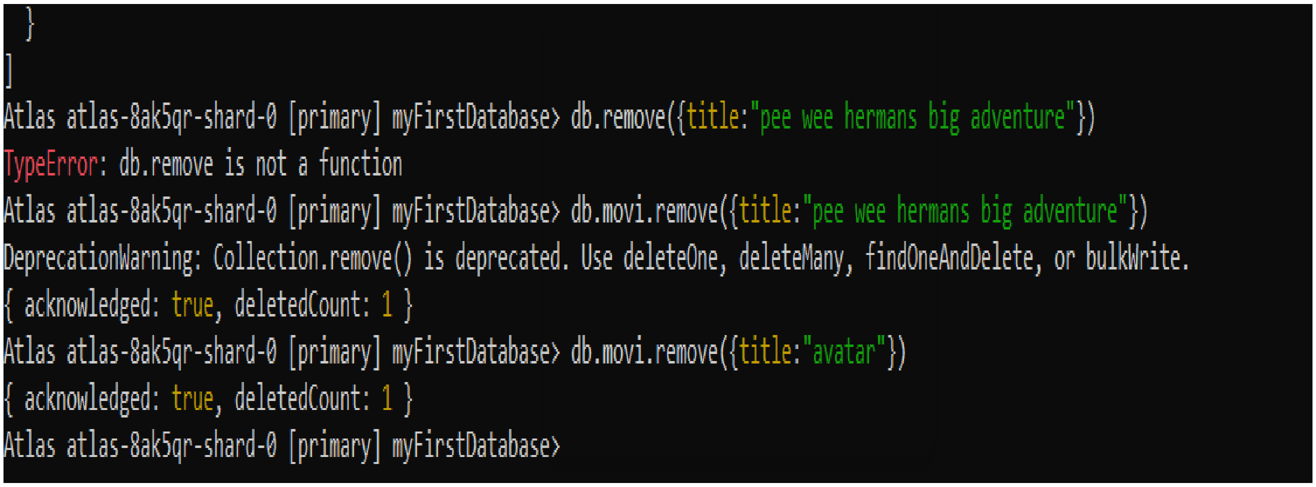
```
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({synopsis:/.bilbo/})
[
  {
    _id: ObjectId("6202324bfe586889b81443eb"),
    title: 'the hobbit:an unexpected journey',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ 'the hobbit' ],
    synopsis: 'a reluctant 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("620235eefe586889b81443ec"),
    title: 'the hobbit:the desolation of smaug',
    writer: 'j.r.r.tolkein',
    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 possession of a mysterious and magical ring.'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({synopsis:/.gandalf/})
[
  {
    _id: ObjectId("620235eefe586889b81443ec"),
    title: 'the hobbit:the desolation of smaug',
    writer: 'j.r.r.tolkein',
    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 possession of a mysterious and magical ring.'
  }
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$and:[{synopsis:/.bilbo./},{synopsisAtlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$and:[{synopsis:/.bilbo./},{synopsis:/.
^gandalf./}]})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$and:[{synopsis:/.bilbo./},{synopsis:/.^gandalf./}]})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$and:[{synopsis:/bilbo/},{synopsis:{$not:/gandalf/}}]})
[
  {
    _id: ObjectId("6202324bfe586889b81443eb"),
    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("620236a7fe586889b81443ed"),
    title: 'the hobbit:the battle of the five armies',
    writer: 'j.r.r.tolkein',
```



```
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.find({$or:[{synopsis:/dwarves/},{synopsis:/hobbit/}]})
[
  {
    _id: ObjectId("6202324bfe586889b81443eb"),
    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("620235eefe586889b81443ec"),
    title: 'the hobbit:the desolation of smaug',
    writer: 'j.r.r.tolkein',
    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 possession of a mysterious and magical ring.'
  }
]
```

Delete Documents

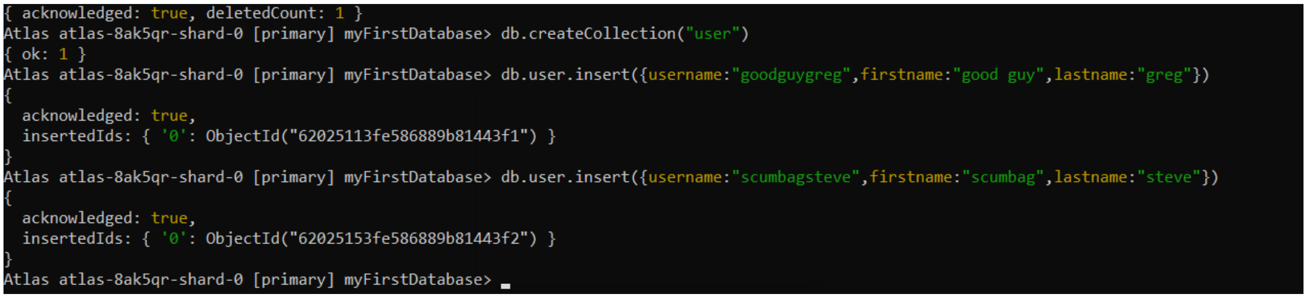
1. delete the movie "Pee Wee Herman's Big Adventure"
2. delete the movie "Avatar"



```
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.remove({title:"pee wee hermans big adventure"})
TypeError: db.remove is not a function
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.remove({title:"pee wee hermans big adventure"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.movi.remove({title:"avatar"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase>
```

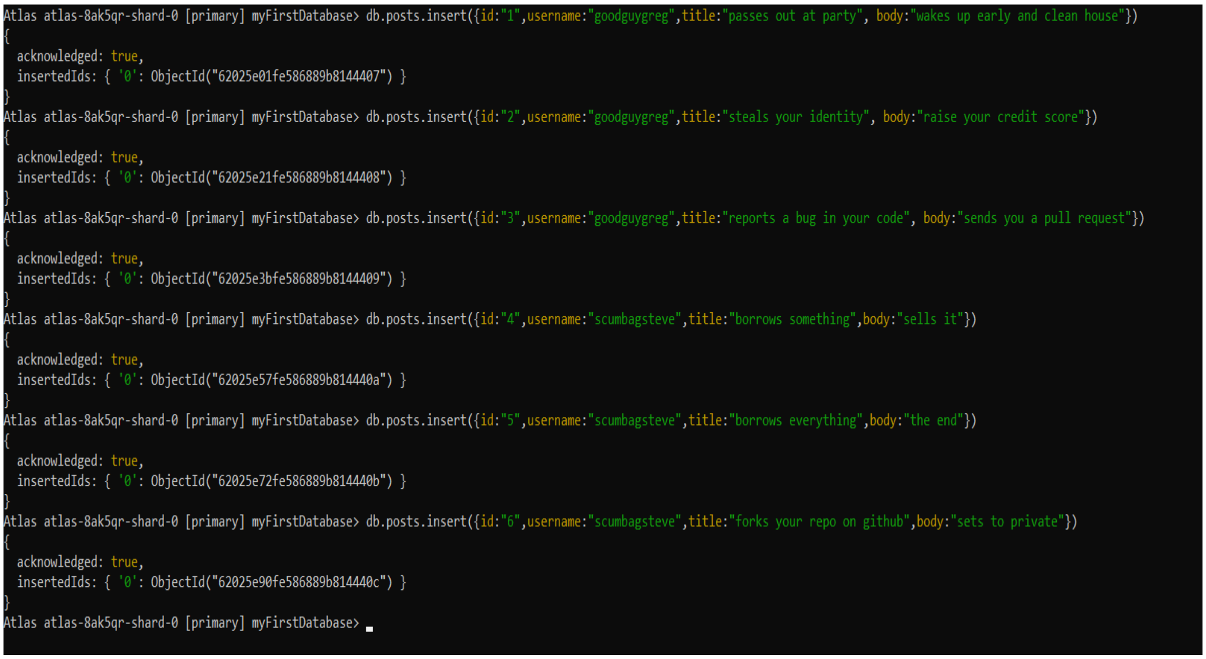
Relationships

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



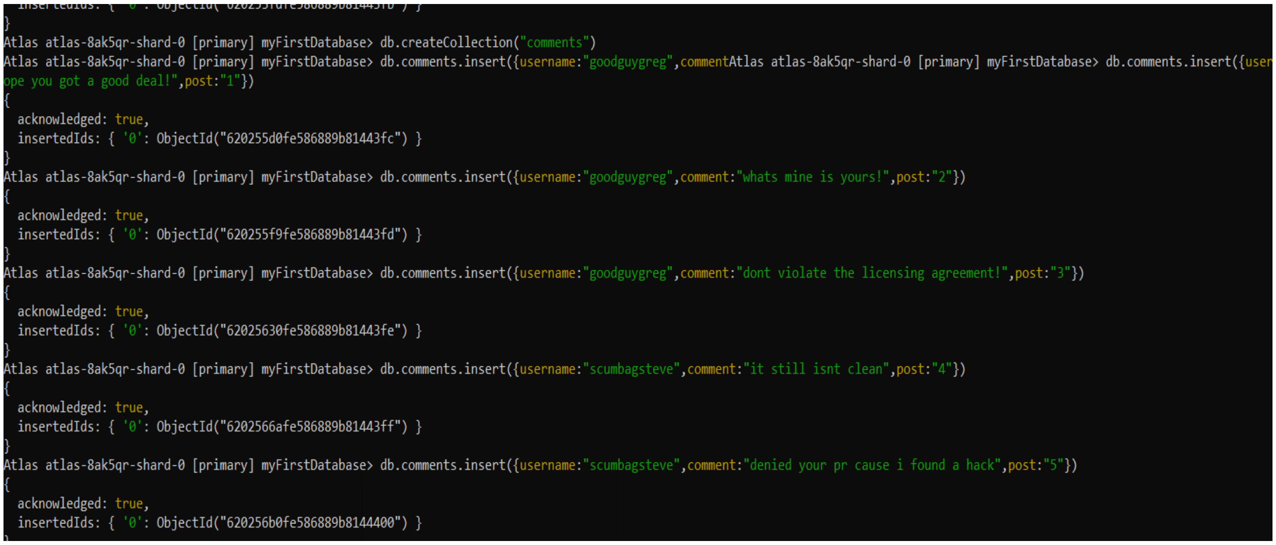
```
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.createCollection("user")
{ ok: 1 }
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.user.insert({username:"goodguygreg",firstname:"good guy",lastname:"greg"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025113fe586889b81443f1") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.user.insert({username:"scumbagsteve",firstname:"scumbag",lastname:"steve"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025153fe586889b81443f2") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> 
```

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



```
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"1",username:"goodguygreg",title:"passes out at party", body:"wakes up early and clean house"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e01fe586889b8144407") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"2",username:"goodguygreg",title:"steals your identity", body:"raise your credit score"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e21fe586889b8144408") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"3",username:"goodguygreg",title:"reports a bug in your code", body:"sends you a pull request"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e3bfe586889b8144409") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"4",username:"scumbagsteve",title:"borrows something",body:"sells it"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e57fe586889b814440a") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"5",username:"scumbagsteve",title:"borrows everything",body:"the end"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e72fe586889b814440b") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"6",username:"scumbagsteve",title:"forks your repo on github",body:"sets to private"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025e90fe586889b814440c") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> 
```

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



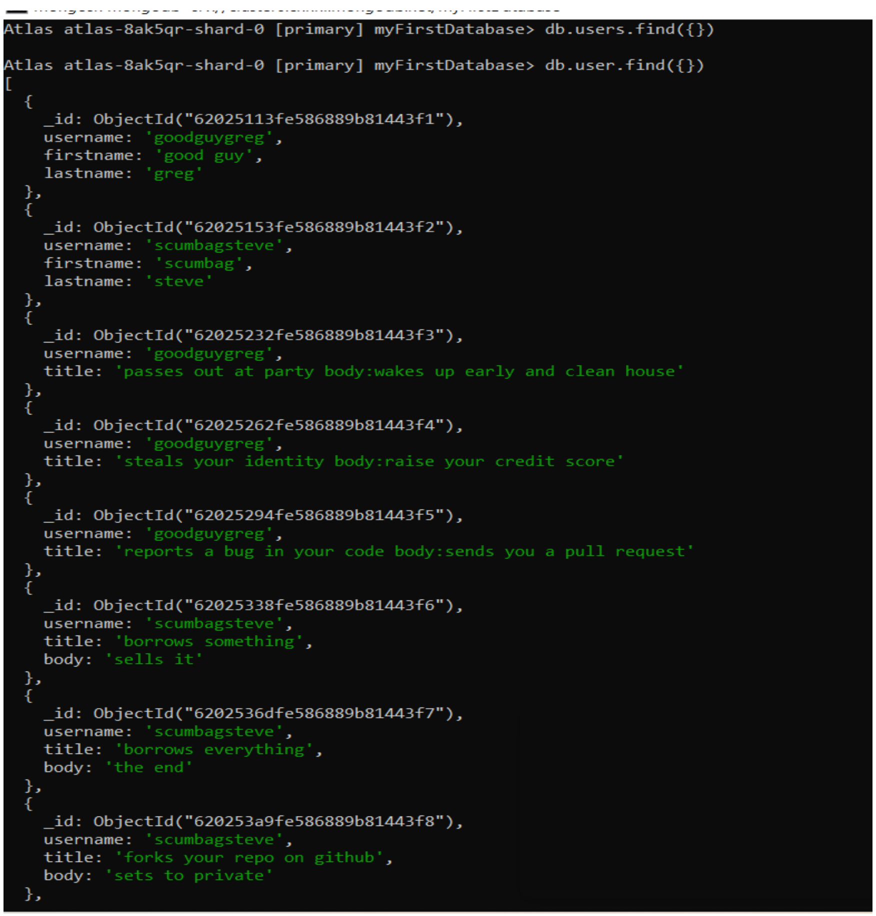
```
  insertedIds: { '0': ObjectId("620255fdfe586889b81443fb") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.createCollection("comments")
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"goodguygreg",commentAtlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({user
ope you got a good deal!",post:"1"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620255d0fe586889b81443fc") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"goodguygreg",comment:"whats mine is yours!",post:"2"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620255f9fe586889b81443fd") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"goodguygreg",comment:"dont violate the licensing agreement!",post:"3"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62025630fe586889b81443fe") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"scumbagsteve",comment:"it still isnt clean",post:"4"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6202566afe586889b81443ff") }
}
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"scumbagsteve",comment:"denied your pr cause i found a hack",post:"5"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620256b0fe586889b8144400") }
```

Querying related collections

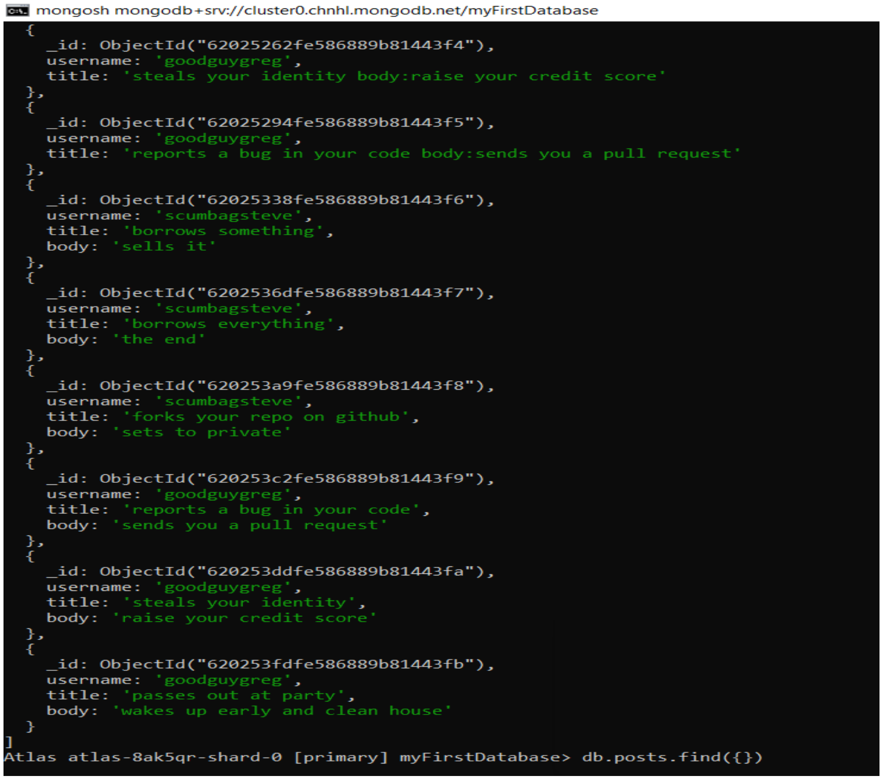
1. find all users
2. find all posts
3. find all posts that was authored by "GoodGuyGreg"
4. find all posts that was authored by "ScumbagSteve"
5. find all comments
6. find all comments that was authored by "GoodGuyGreg"

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

8. find all comments belonging to the post "Reports a bug in your code"

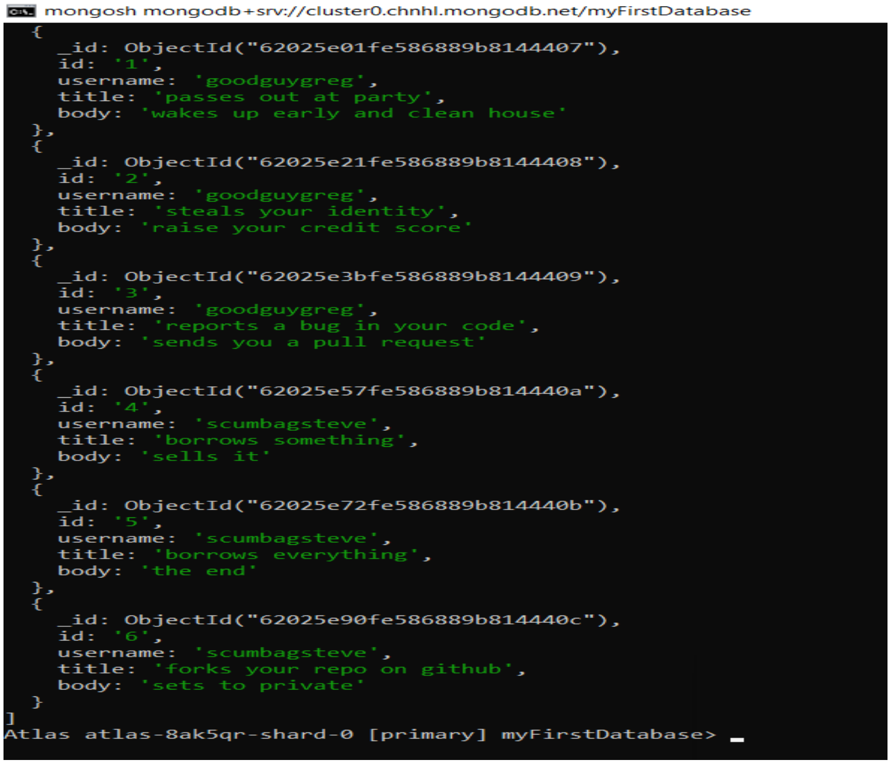


```
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.users.find({})

Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.user.find({})
[
  {
    _id: ObjectId("62025113fe586889b81443f1"),
    username: 'goodguygreg',
    firstname: 'good guy',
    lastname: 'greg'
  },
  {
    _id: ObjectId("62025153fe586889b81443f2"),
    username: 'scumbagsteve',
    firstname: 'scumbag',
    lastname: 'steve'
  },
  {
    _id: ObjectId("62025232fe586889b81443f3"),
    username: 'goodguygreg',
    title: 'passes out at party body:wakes up early and clean house'
  },
  {
    _id: ObjectId("62025262fe586889b81443f4"),
    username: 'goodguygreg',
    title: 'steals your identity body:raise your credit score'
  },
  {
    _id: ObjectId("62025294fe586889b81443f5"),
    username: 'goodguygreg',
    title: 'reports a bug in your code body:sends you a pull request'
  },
  {
    _id: ObjectId("62025338fe586889b81443f6"),
    username: 'scumbagsteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("6202536dfe586889b81443f7"),
    username: 'scumbagsteve',
    title: 'borrows everything',
    body: 'the end'
  },
  {
    _id: ObjectId("620253a9fe586889b81443f8"),
    username: 'scumbagsteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  },
```



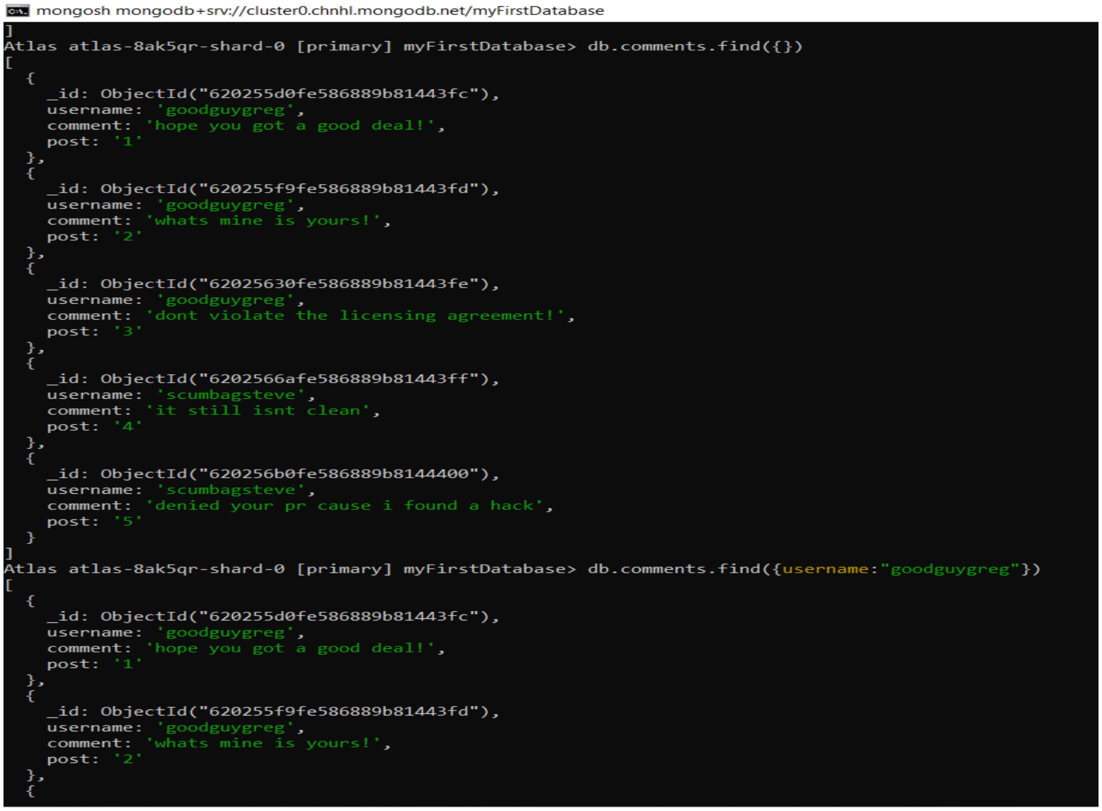
mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
  {
    _id: ObjectId("62025262fe586889b81443f4"),
    username: 'goodguygreg',
    title: 'steals your identity body:raise your credit score'
  },
  {
    _id: ObjectId("62025294fe586889b81443f5"),
    username: 'goodguygreg',
    title: 'reports a bug in your code body:sends you a pull request'
  },
  {
    _id: ObjectId("62025338fe586889b81443f6"),
    username: 'scumbagsteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("6202536dfe586889b81443f7"),
    username: 'scumbagsteve',
    title: 'borrows everything',
    body: 'the end'
  },
  {
    _id: ObjectId("620253a9fe586889b81443f8"),
    username: 'scumbagsteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  },
  {
    _id: ObjectId("620253c2fe586889b81443f9"),
    username: 'goodguygreg',
    title: 'reports a bug in your code',
    body: 'sends you a pull request'
  },
  {
    _id: ObjectId("620253ddfe586889b81443fa"),
    username: 'goodguygreg',
    title: 'steals your identity',
    body: 'raise your credit score'
  },
  {
    _id: ObjectId("620253fdfe586889b81443fb"),
    username: 'goodguygreg',
    title: 'passes out at party',
    body: 'wakes up early and clean house'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.posts.find({})
```



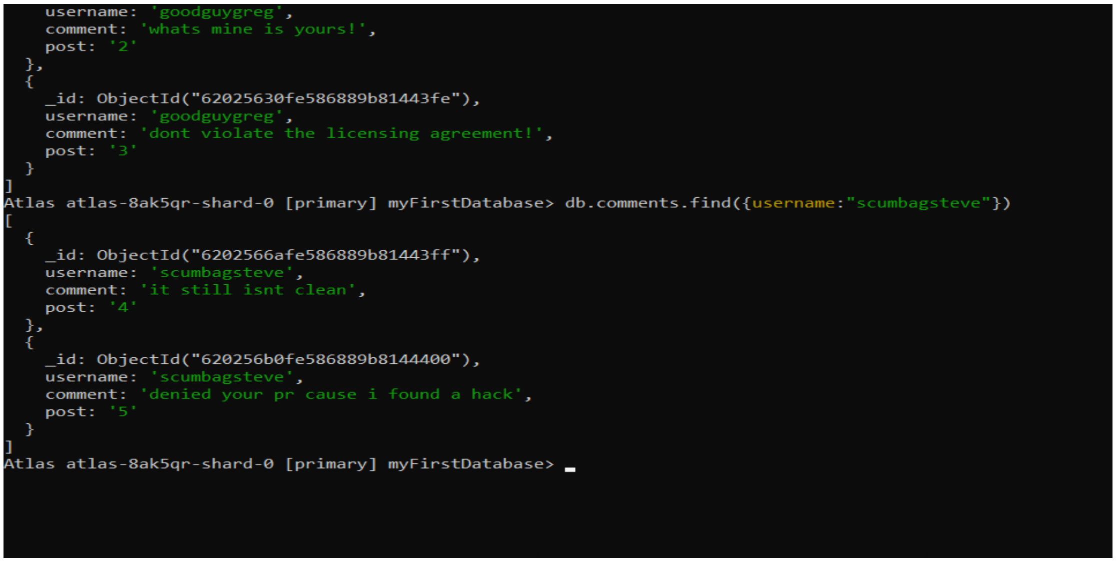
mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
  {
    _id: ObjectId("62025e01fe586889b8144407"),
    id: '1',
    username: 'goodguygreg',
    title: 'passes out at party',
    body: 'wakes up early and clean house'
  },
  {
    _id: ObjectId("62025e21fe586889b8144408"),
    id: '2',
    username: 'goodguygreg',
    title: 'steals your identity',
    body: 'raise your credit score'
  },
  {
    _id: ObjectId("62025e3bfe586889b8144409"),
    id: '3',
    username: 'goodguygreg',
    title: 'reports a bug in your code',
    body: 'sends you a pull request'
  },
  {
    _id: ObjectId("62025e57fe586889b814440a"),
    id: '4',
    username: 'scumbagsteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("62025e72fe586889b814440b"),
    id: '5',
    username: 'scumbagsteve',
    title: 'borrows everything',
    body: 'the end'
  },
  {
    _id: ObjectId("62025e90fe586889b814440c"),
    id: '6',
    username: 'scumbagsteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> _
```



mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase

```
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.find({})
[
  {
    _id: ObjectId("620255d0fe586889b81443fc"),
    username: 'goodguygreg',
    comment: 'hope you got a good deal!',
    post: '1'
  },
  {
    _id: ObjectId("620255f9fe586889b81443fd"),
    username: 'goodguygreg',
    comment: 'whats mine is yours!',
    post: '2'
  },
  {
    _id: ObjectId("62025630fe586889b81443fe"),
    username: 'goodguygreg',
    comment: 'dont violate the licensing agreement!',
    post: '3'
  },
  {
    _id: ObjectId("6202566afe586889b81443ff"),
    username: 'scumbagsteve',
    comment: 'it still isnt clean',
    post: '4'
  },
  {
    _id: ObjectId("620256b0fe586889b8144400"),
    username: 'scumbagsteve',
    comment: 'denied your pr cause i found a hack',
    post: '5'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.find({username:"goodguygreg"})
[
  {
    _id: ObjectId("620255d0fe586889b81443fc"),
    username: 'goodguygreg',
    comment: 'hope you got a good deal!',
    post: '1'
  },
  {
    _id: ObjectId("620255f9fe586889b81443fd"),
    username: 'goodguygreg',
    comment: 'whats mine is yours!',
    post: '2'
  },
  {
```



```
    username: 'goodguygreg',
    comment: 'whats mine is yours!',
    post: '2'
  },
  {
    _id: ObjectId("62025630fe586889b81443fe"),
    username: 'goodguygreg',
    comment: 'dont violate the licensing agreement!',
    post: '3'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> db.comments.find({username:"scumbagsteve"})
[
  {
    _id: ObjectId("6202566afe586889b81443ff"),
    username: 'scumbagsteve',
    comment: 'it still isnt clean',
    post: '4'
  },
  {
    _id: ObjectId("620256b0fe586889b8144400"),
    username: 'scumbagsteve',
    comment: 'denied your pr cause i found a hack',
    post: '5'
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> _
```