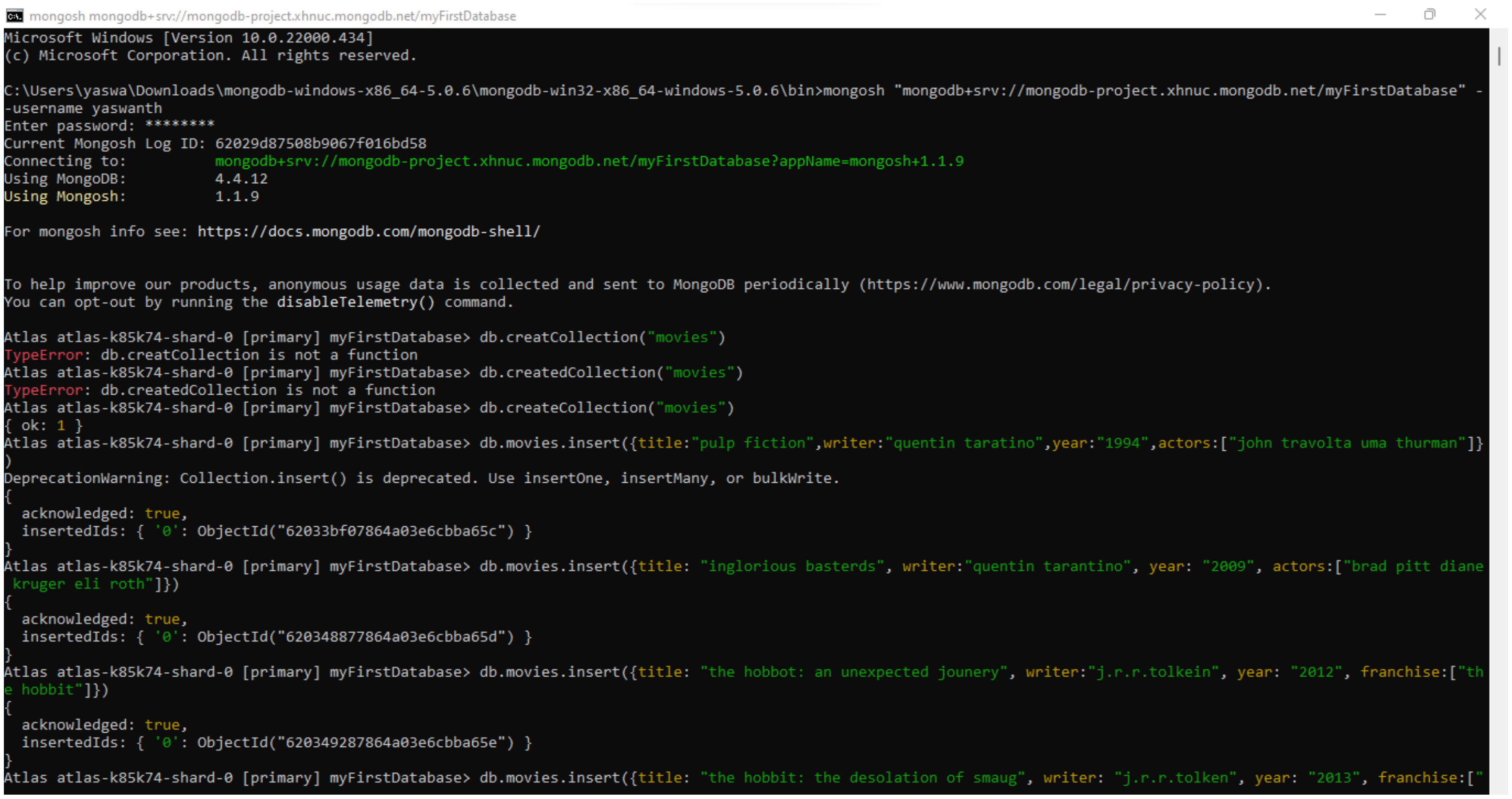
Gudipati yaswanth

MongoDB-Assignments1

MongoDB Exercise in mongo shell

Insert Documents

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



mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

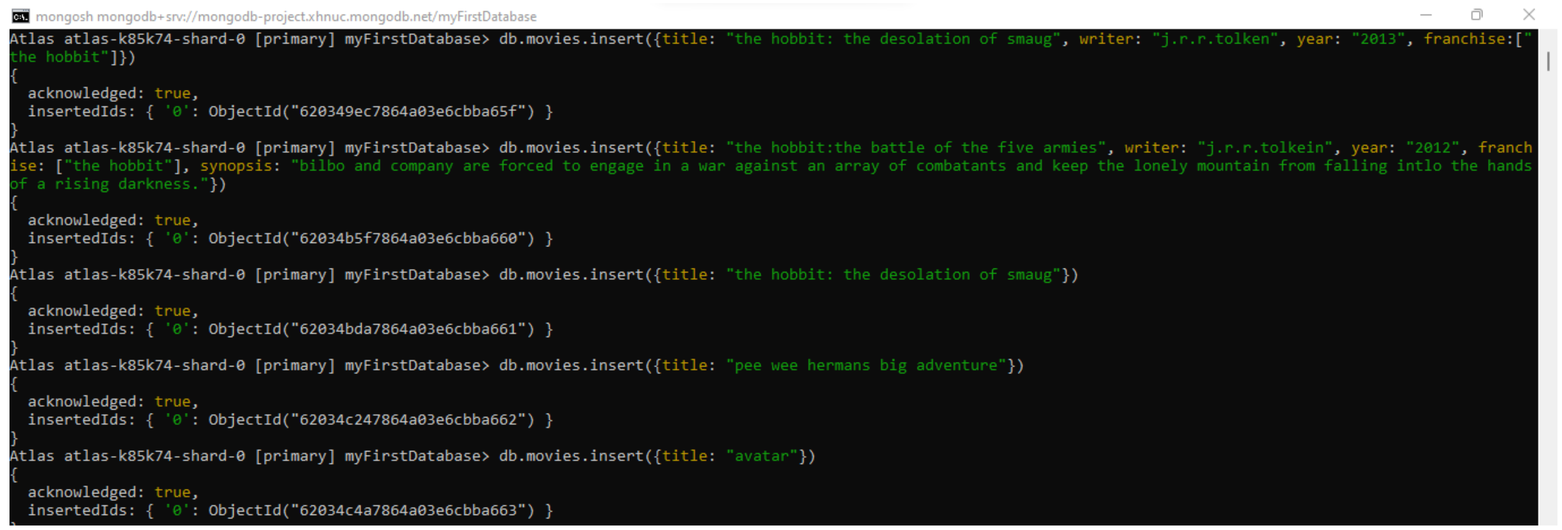
```
Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yaswa\Downloads\mongodb-windows-x86_64-5.0.6\mongodb-win32-x86_64-windows-5.0.6\bin>mongosh "mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase" -
-username yaswanth
Enter password: ********
Current Mongosh Log ID: 62029d87508b9067f016bd58
Connecting to:          mongodb+srv://mongodb-project.xhnuc.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/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.creatCollection("movies")
TypeError: db.creatCollection is not a function
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.createdCollection("movies")
TypeError: db.createdCollection is not a function
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.createCollection("movies")
{ ok: 1 }
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title:"pulp fiction",writer:"quentin taratino",year:"1994",actors:["john travolta uma thurman"]}
)
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62033bf07864a03e6cbba65c") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "inglorious basterds", writer:"quentin tarantino", year: "2009", actors:["brad pitt diane
 kruger eli roth"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620348877864a03e6cbba65d") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "the hobbot: an unexpected jounery", writer:"j.r.r.tolkein", year: "2012", franchise:["th
e hobbit"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620349287864a03e6cbba65e") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "the hobbit: the desolation of smaug", writer: "j.r.r.tolken", year: "2013", franchise:["
```



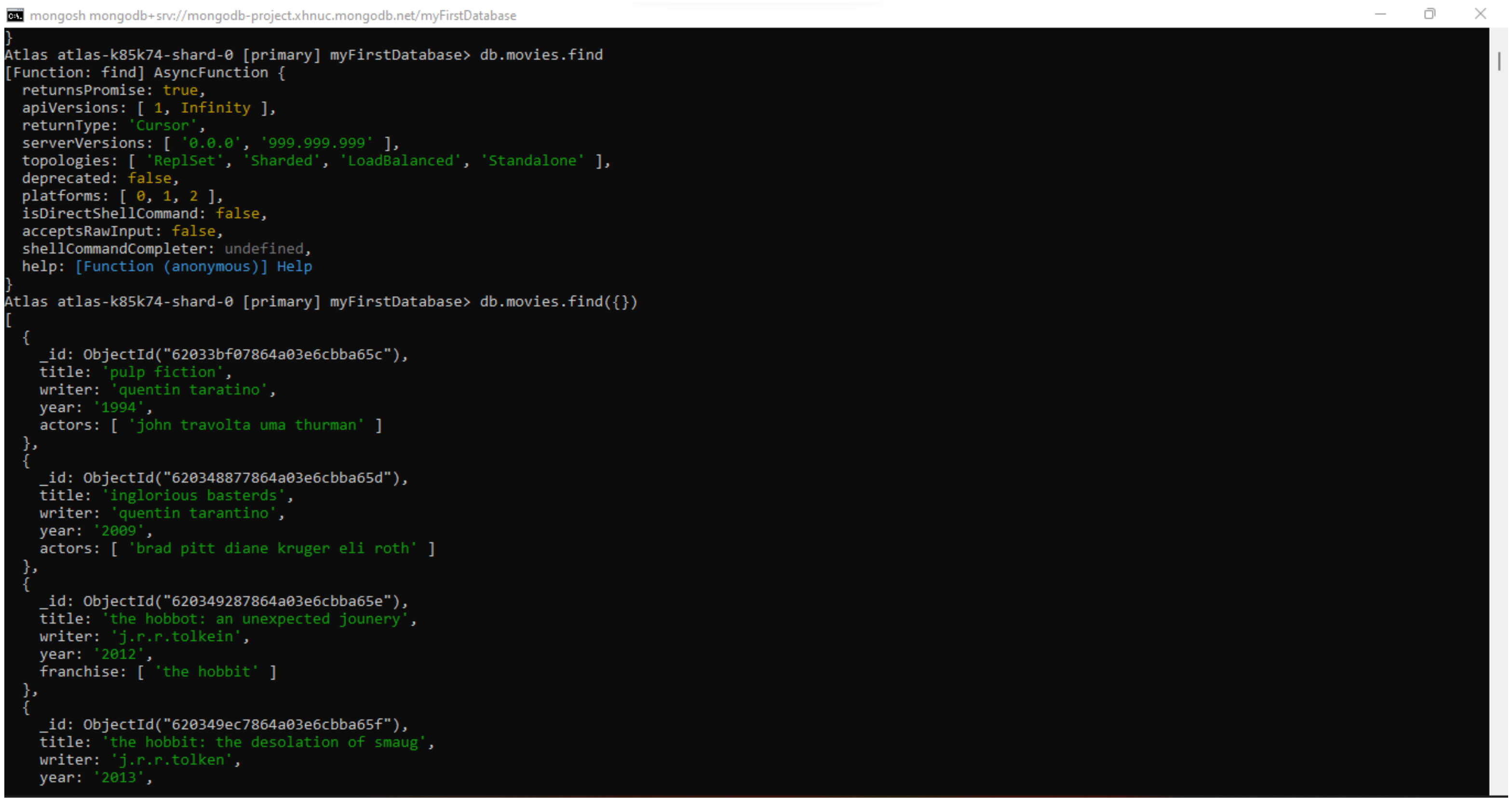
mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "the hobbit: the desolation of smaug", writer: "j.r.r.tolken", year: "2013", franchise:["
the hobbit"]})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("620349ec7864a03e6cbba65f") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "the hobbit:the battle of the five armies", writer: "j.r.r.tolkein", year: "2012", franch
ise: ["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 intlo the hands
of a rising darkness."})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62034b5f7864a03e6cbba660") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "the hobbit: the desolation of smaug"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62034bda7864a03e6cbba661") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "pee wee hermans big adventure"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62034c247864a03e6cbba662") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.insert({title: "avatar"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("62034c4a7864a03e6cbba663") }
```

Query / Find Documents

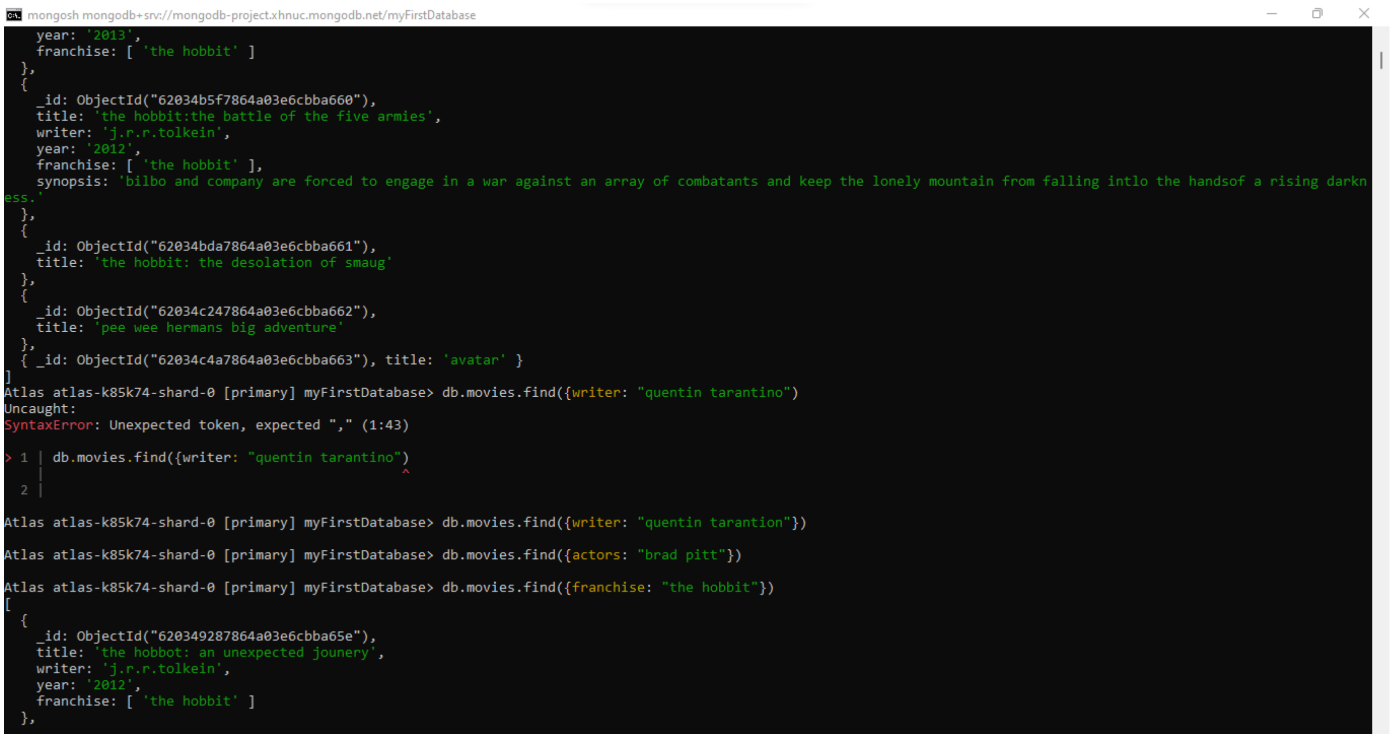
query the **movies** collection to

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://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.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-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({})
[
  {
    _id: ObjectId("62033bf07864a03e6cbba65c"),
    title: 'pulp fiction',
    writer: 'quentin taratino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  },
  {
    _id: ObjectId("620348877864a03e6cbba65d"),
    title: 'inglorious basterds',
    writer: 'quentin tarantino',
    year: '2009',
    actors: [ 'brad pitt diane kruger eli roth' ]
  },
  {
    _id: ObjectId("620349287864a03e6cbba65e"),
    title: 'the hobbot: an unexpected jounery',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("620349ec7864a03e6cbba65f"),
    title: 'the hobbit: the desolation of smaug',
    writer: 'j.r.r.tolken',
    year: '2013',
```



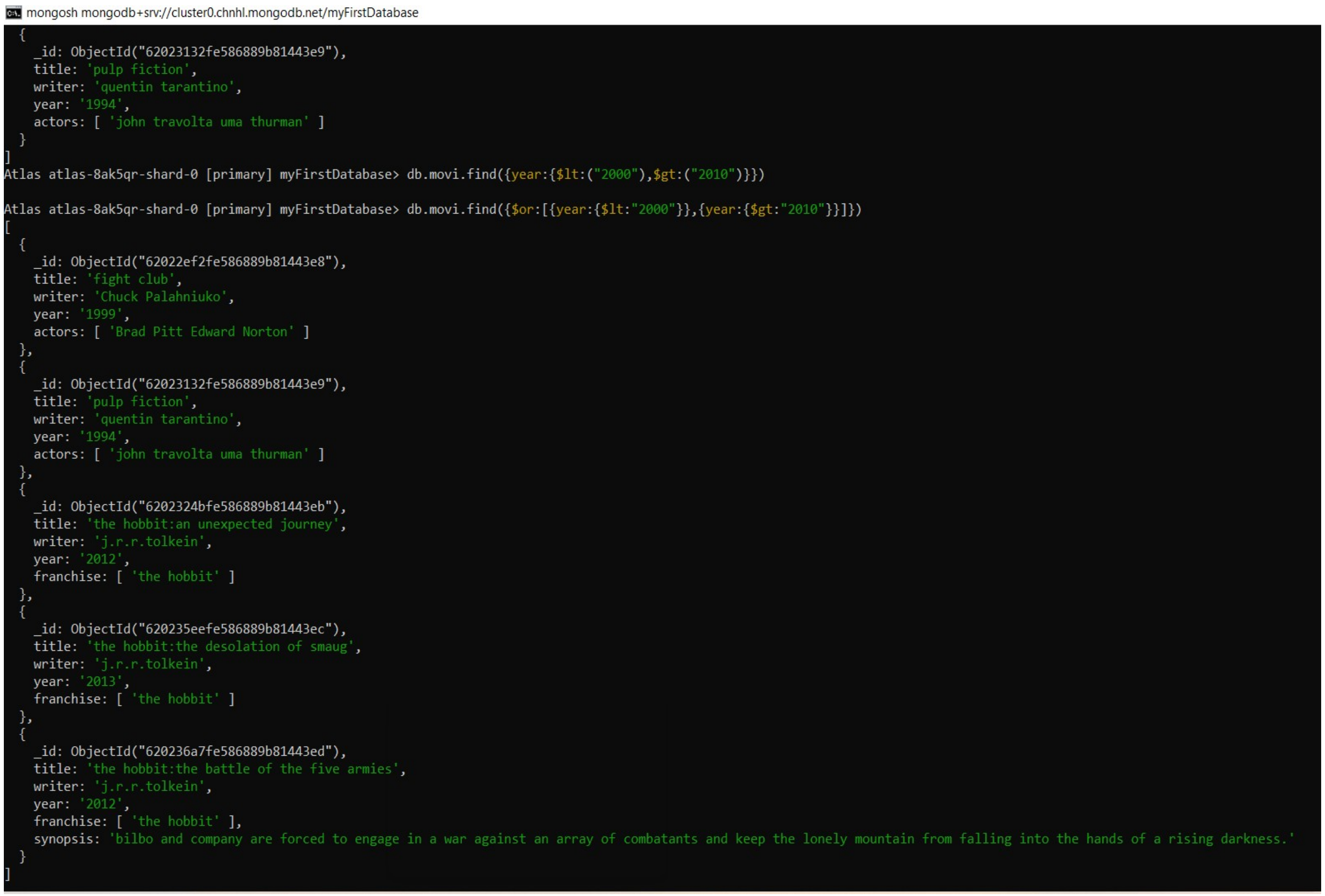
```
mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase
    year: '2013',
    franchise: [ 'the hobbit' ]
  },
  {
    _id: ObjectId("62034b5f7864a03e6cbba660"),
    title: 'the hobbit:the battle of the five armies',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ '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 intlo the handsof a rising darkn
ess.'
  },
  {
    _id: ObjectId("62034bda7864a03e6cbba661"),
    title: 'the hobbit: the desolation of smaug'
  },
  {
    _id: ObjectId("62034c247864a03e6cbba662"),
    title: 'pee wee hermans big adventure'
  },
  { _id: ObjectId("62034c4a7864a03e6cbba663"), title: 'avatar' }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({writer: "quentin tarantino")
Uncaught:
SyntaxError: Unexpected token, expected "," (1:43)

> 1 | db.movies.find({writer: "quentin tarantino")
    |                                            ^
  2 |

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({writer: "quentin tarantion"})

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({actors: "brad pitt"})

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({franchise: "the hobbit"})
[
  {
    _id: ObjectId("620349287864a03e6cbba65e"),
    title: 'the hobbot: an unexpected jounery',
    writer: 'j.r.r.tolkein',
    year: '2012',
    franchise: [ 'the hobbit' ]
  },
```

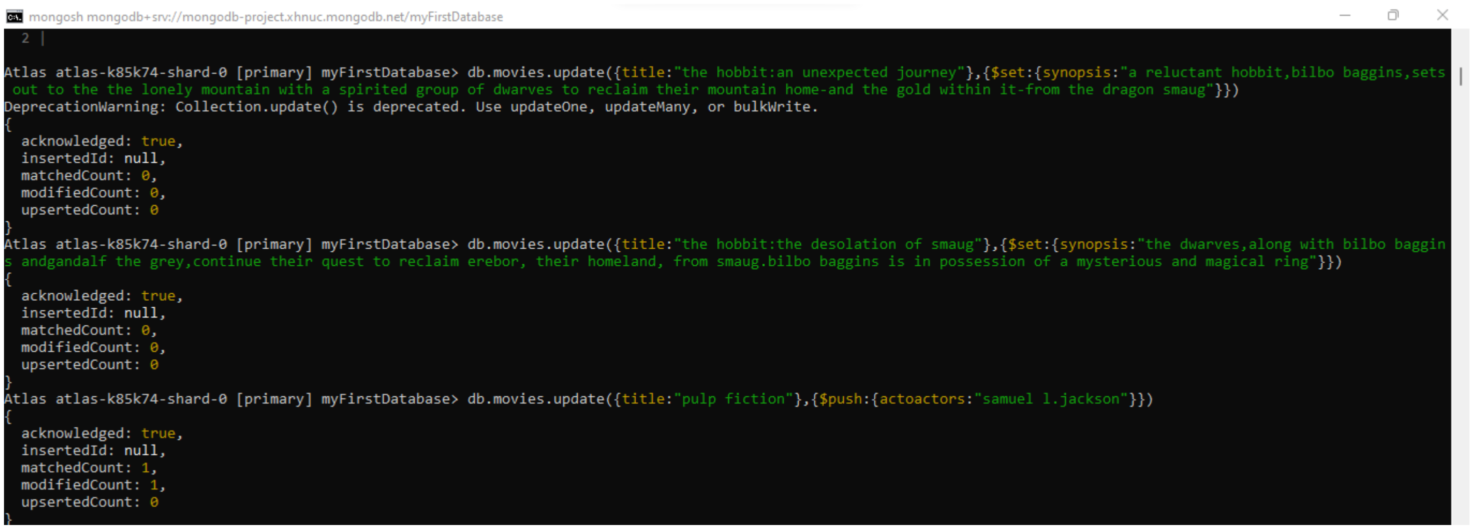


```
mongosh mongodb+srv://cluster0.chnhl.mongodb.net/myFirstDatabase
  {
    _id: ObjectId("62023132fe586889b81443e9"),
    title: 'pulp fiction',
    writer: 'quentin tarantino',
    year: '1994',
    actors: [ 'john travolta uma thurman' ]
  }
]
Atlas atlas-8ak5qr-shard-0 [primary] myFirstDatabase> 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"



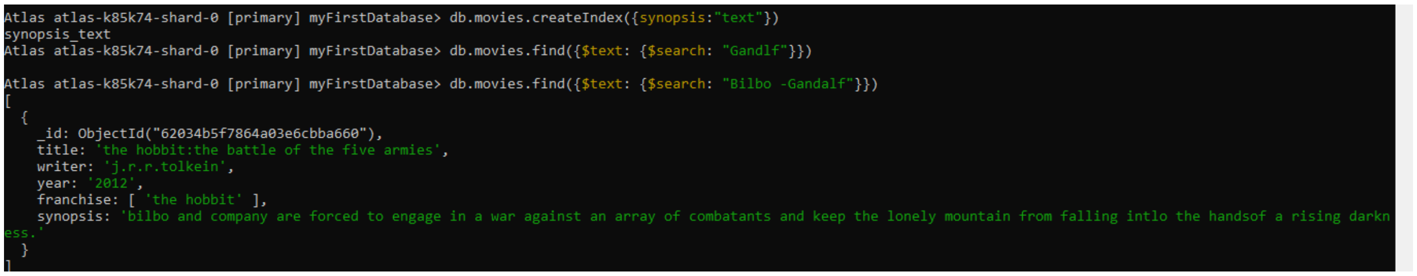
```
mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase
  2 |

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.update({title:"the hobbit:an unexpected journey"},{$set:{synopsis:"a reluctant hobbit,bilbo baggins,sets
 out to the the lonely mountain with a spirited group of dwarves to reclaim their mountain home-and the gold within it-from the dragon smaug"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.update({title:"the hobbit:the desolation of smaug"},{$set:{synopsis:"the dwarves,along with bilbo baggin
s andgandalf 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: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.update({title:"pulp fiction"},{$push:{actoactors:"samuel l.jackson"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Text Search

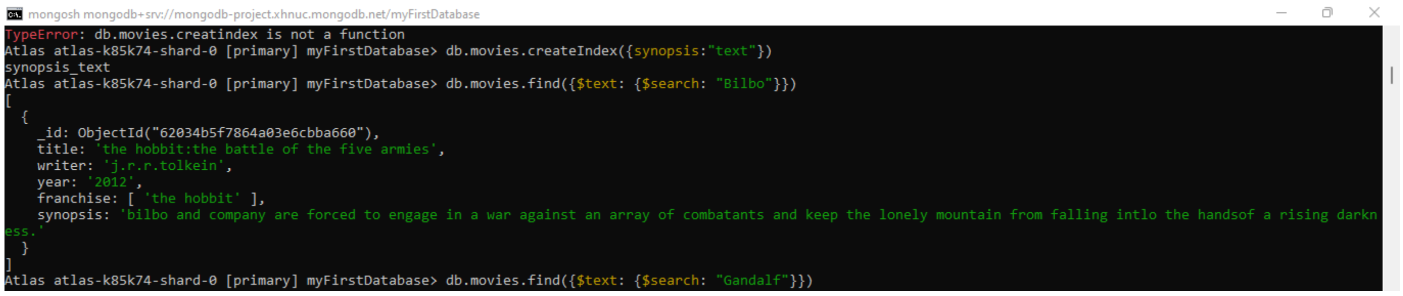
1. find all movies that have a synopsis that contains the word "Bilbo"
2. find all movies that have a synopsis that contains the word "Gandalf"
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"
4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

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



```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.createIndex({synopsis:"text"})
synopsis_text
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({$text: {$search: "Gandlf"}})

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({$text: {$search: "Bilbo -Gandalf"}})
[
  {
    _id: ObjectId("62034b5f7864a03e6cbba660"),
    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 intlo the handsof a rising darkn
ess.'
  }
]
```

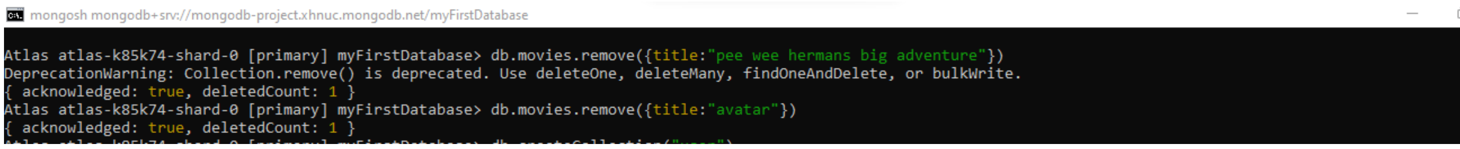


mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
TypeError: db.movies.creatindex is not a function
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.createIndex({synopsis:"text"})
synopsis_text
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({$text: {$search: "Bilbo"}})
[
  {
    _id: ObjectId("62034b5f7864a03e6cbba660"),
    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 intlo the handsof a rising darkn
ess.'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.find({$text: {$search: "Gandalf"}})
```

Delete Documents

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

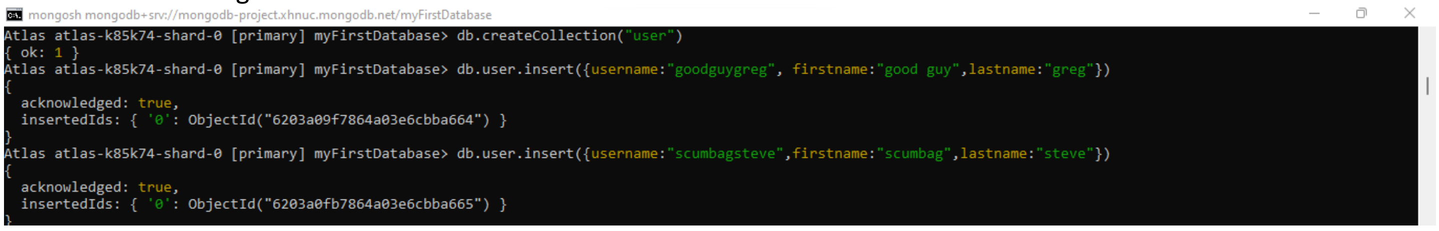


mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.remove({title:"pee wee hermans big adventure"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.movies.remove({title:"avatar"})
{ acknowledged: true, deletedCount: 1 }
```

Relationships

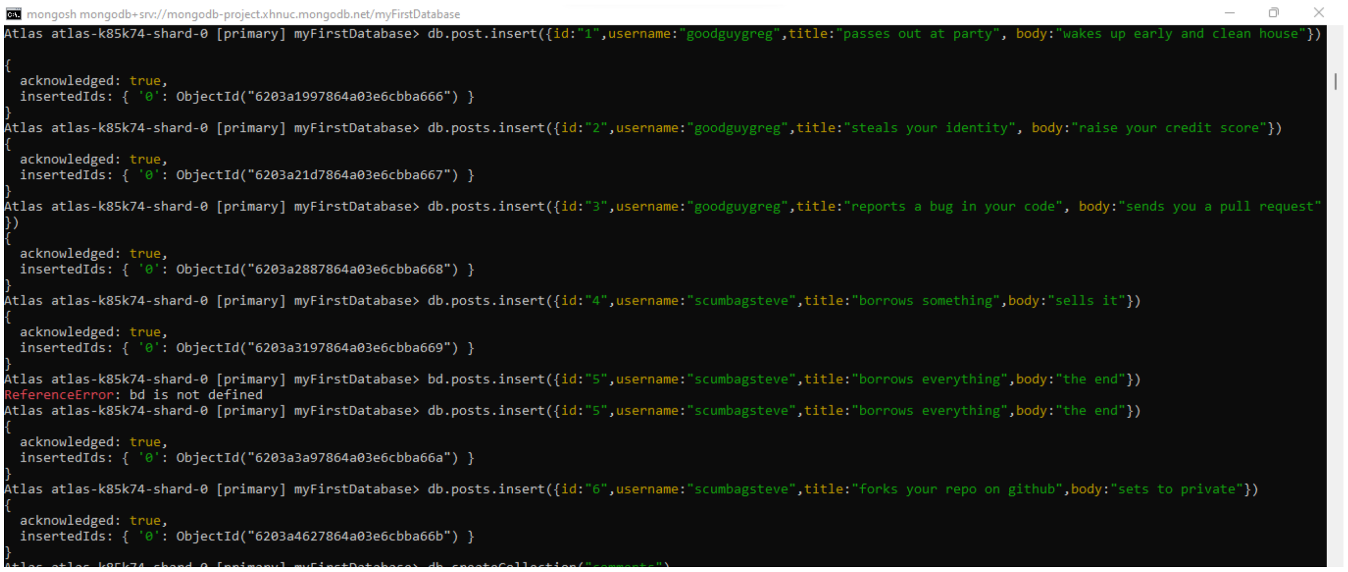
Insert the following documents into a **users** collection



mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.createCollection("user")
{ ok: 1 }
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.user.insert({username:"goodguygreg", firstname:"good guy",lastname:"greg"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203a09f7864a03e6cbba664") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.user.insert({username:"scumbagsteve",firstname:"scumbag",lastname:"steve"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203a0fb7864a03e6cbba665") }
```

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

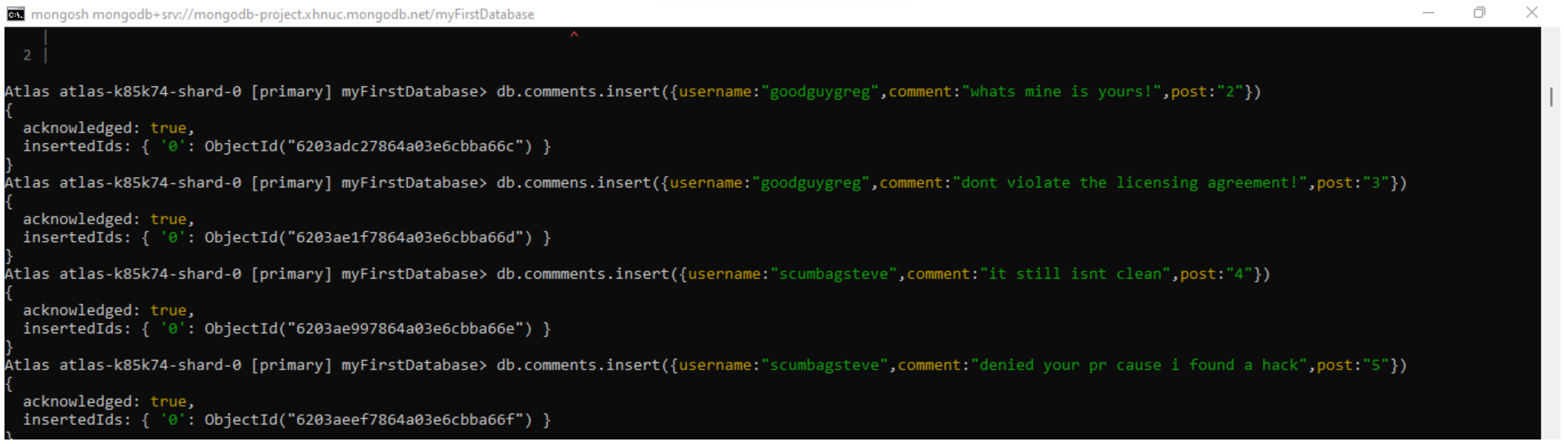


mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.post.insert({id:"1",username:"goodguygreg",title:"passes out at party", body:"wakes up early and clean house"})

{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203a1997864a03e6cbba666") }
}
Atlas atlas-k85k74-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("6203a21d7864a03e6cbba667") }
}
Atlas atlas-k85k74-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("6203a2887864a03e6cbba668") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"4",username:"scumbagsteve",title:"borrows something",body:"sells it"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203a3197864a03e6cbba669") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> bd.posts.insert({id:"5",username:"scumbagsteve",title:"borrows everything",body:"the end"})
ReferenceError: bd is not defined
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.posts.insert({id:"5",username:"scumbagsteve",title:"borrows everything",body:"the end"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203a3a97864a03e6cbba66a") }
}
Atlas atlas-k85k74-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("6203a4627864a03e6cbba66b") }
}
```

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



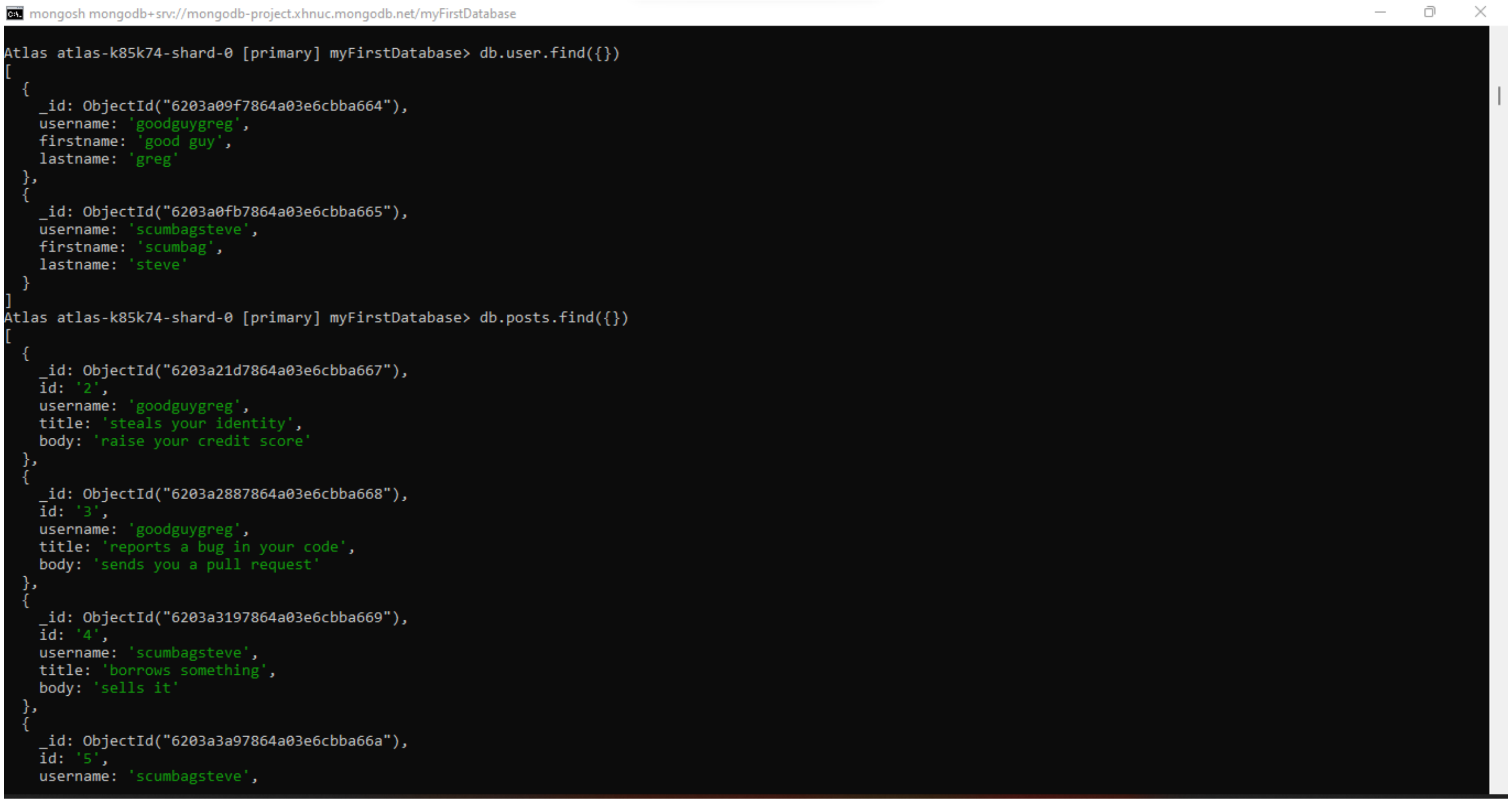
```
mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase
  |
2 |

Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.comments.insert({username:"goodguygreg",comment:"whats mine is yours!",post:"2"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203adc27864a03e6cbba66c") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.commens.insert({username:"goodguygreg",comment:"dont violate the licensing agreement!",post:"3"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203ae1f7864a03e6cbba66d") }
}
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.commments.insert({username:"scumbagsteve",comment:"it still isnt clean",post:"4"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6203ae997864a03e6cbba66e") }
}
Atlas atlas-k85k74-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("6203aeef7864a03e6cbba66f") }
```

Querying related collections

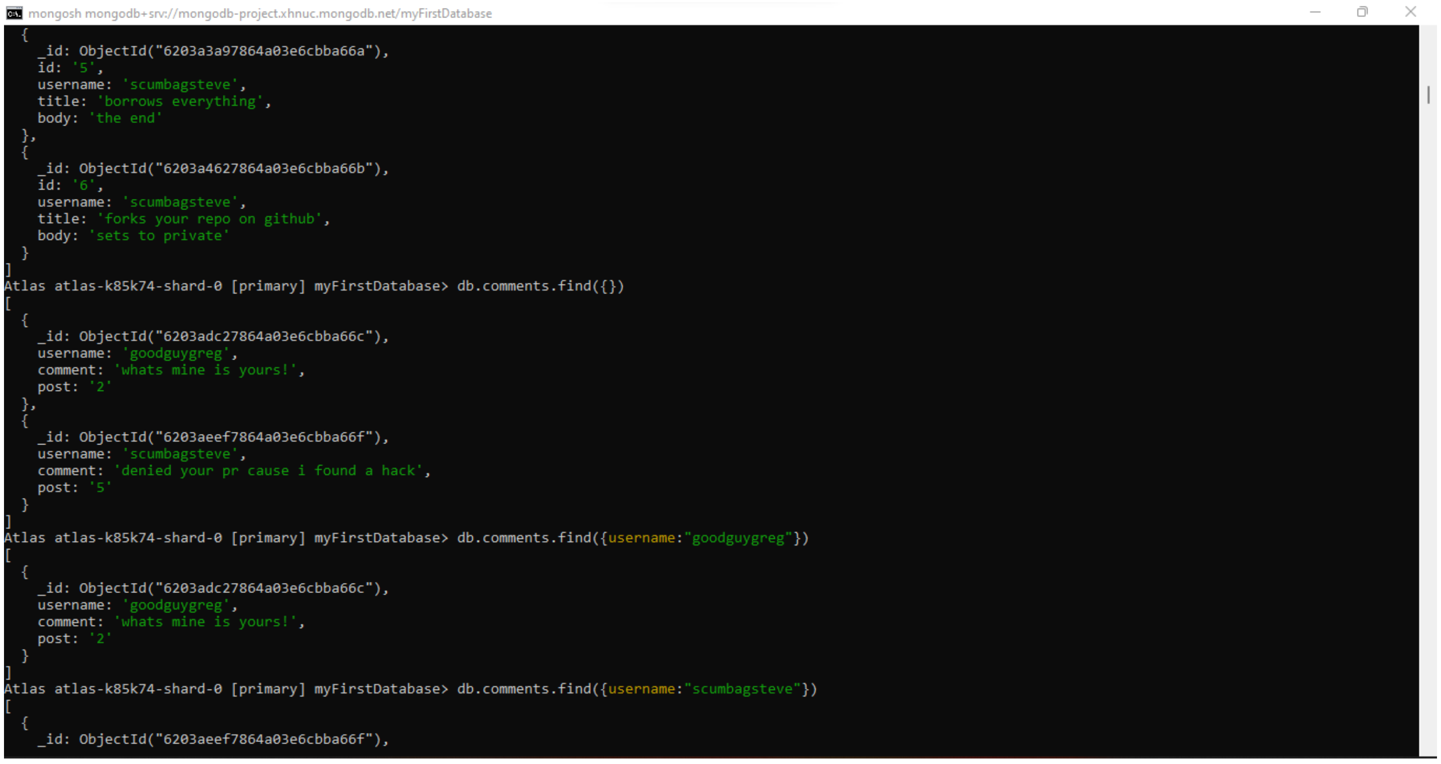
1. find all users
2. find all posts
3. find all posts that was authored by "GoodGuyGreg"
4. find all posts that was authored by "ScumbagSteve"
5. find all comments
6. find all comments that was authored by "GoodGuyGreg"

7. find all comments that was authored by "ScumbagSteve"
8. find all comments belonging to the post "Reports a bug in your code"



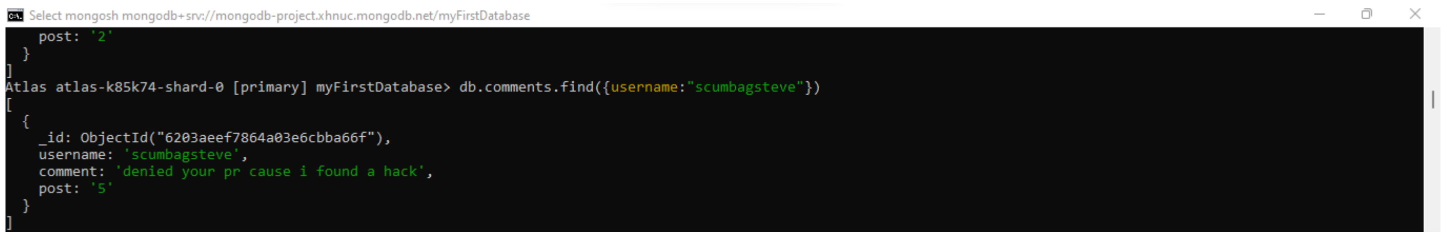
mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.user.find({})
[
  {
    _id: ObjectId("6203a09f7864a03e6cbba664"),
    username: 'goodguygreg',
    firstname: 'good guy',
    lastname: 'greg'
  },
  {
    _id: ObjectId("6203a0fb7864a03e6cbba665"),
    username: 'scumbagsteve',
    firstname: 'scumbag',
    lastname: 'steve'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.posts.find({})
[
  {
    _id: ObjectId("6203a21d7864a03e6cbba667"),
    id: '2',
    username: 'goodguygreg',
    title: 'steals your identity',
    body: 'raise your credit score'
  },
  {
    _id: ObjectId("6203a2887864a03e6cbba668"),
    id: '3',
    username: 'goodguygreg',
    title: 'reports a bug in your code',
    body: 'sends you a pull request'
  },
  {
    _id: ObjectId("6203a3197864a03e6cbba669"),
    id: '4',
    username: 'scumbagsteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("6203a3a97864a03e6cbba66a"),
    id: '5',
    username: 'scumbagsteve',
```



mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
  {
    _id: ObjectId("6203a3a97864a03e6cbba66a"),
    id: '5',
    username: 'scumbagsteve',
    title: 'borrows everything',
    body: 'the end'
  },
  {
    _id: ObjectId("6203a4627864a03e6cbba66b"),
    id: '6',
    username: 'scumbagsteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.comments.find({})
[
  {
    _id: ObjectId("6203adc27864a03e6cbba66c"),
    username: 'goodguygreg',
    comment: 'whats mine is yours!',
    post: '2'
  },
  {
    _id: ObjectId("6203aeef7864a03e6cbba66f"),
    username: 'scumbagsteve',
    comment: 'denied your pr cause i found a hack',
    post: '5'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.comments.find({username:"goodguygreg"})
[
  {
    _id: ObjectId("6203adc27864a03e6cbba66c"),
    username: 'goodguygreg',
    comment: 'whats mine is yours!',
    post: '2'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.comments.find({username:"scumbagsteve"})
[
  {
    _id: ObjectId("6203aeef7864a03e6cbba66f"),
```



Select mongosh mongodb+srv://mongodb-project.xhnuc.mongodb.net/myFirstDatabase

```
    post: '2'
  }
]
Atlas atlas-k85k74-shard-0 [primary] myFirstDatabase> db.comments.find({username:"scumbagsteve"})
[
  {
    _id: ObjectId("6203aeef7864a03e6cbba66f"),
    username: 'scumbagsteve',
    comment: 'denied your pr cause i found a hack',
    post: '5'
  }
]
```