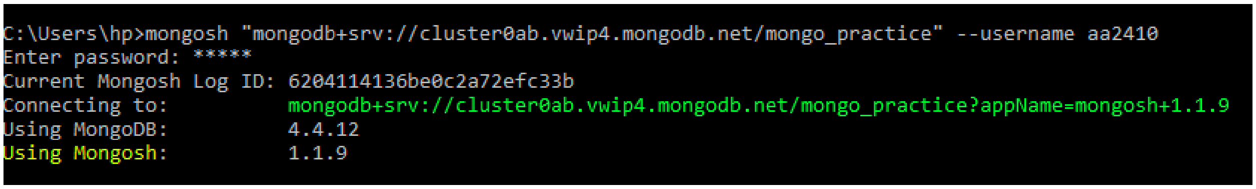
ASSIGNMENT 2.1

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.



```
C:\Users\hp>mongosh "mongodb+srv://cluster0ab.vwip4.mongodb.net/mongo_practice" --username aa2410
Enter password: *****
Current Mongosh Log ID: 6204114136be0c2a72efc33b
Connecting to:          mongodb+srv://cluster0ab.vwip4.mongodb.net/mongo_practice?appName=mongosh+1.1.9
Using MongoDB:          4.4.12
Using Mongosh:          1.1.9
```

Insert Documents

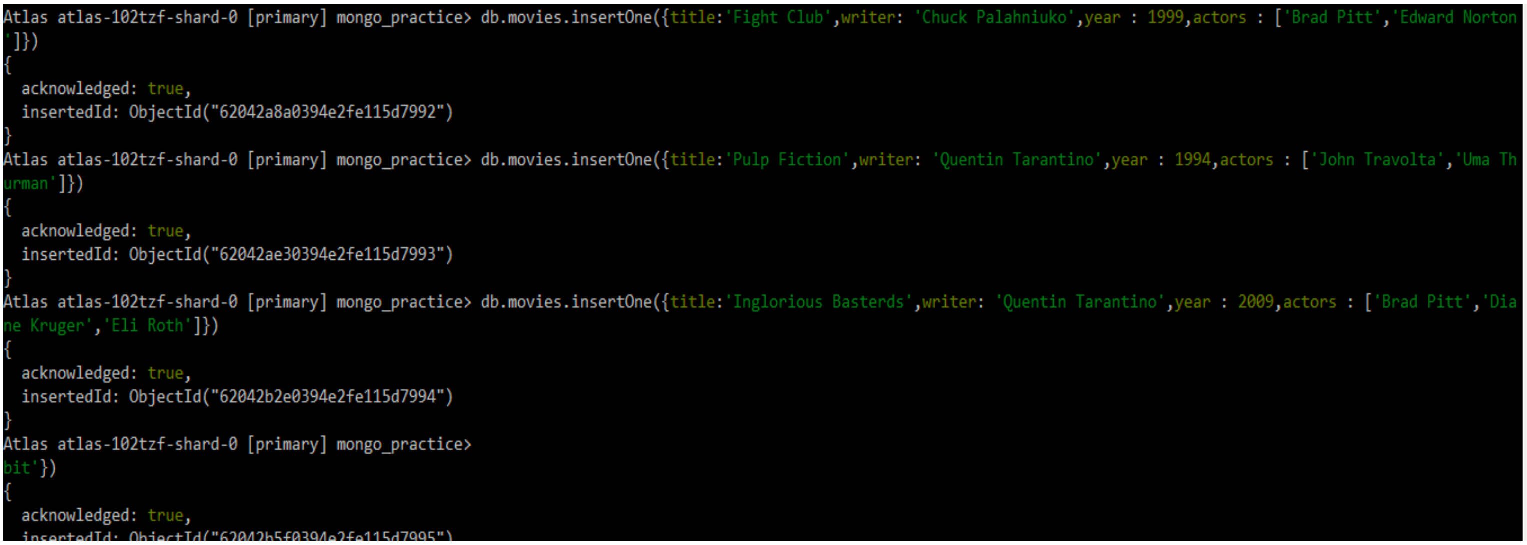
Insert the following documents into a movies collection.
title : Fight Club
writer: Chuck Palahniuko
year : 1999
actors : [
Brad Pitt
Edward Norton
]
title : Pulp Fiction
writer : Quentin Tarantino
year : 1994
actors : [
John Travolta
Uma Thurman
]

title : Inglorious Basterds
writer : Quentin Tarantino
year : 2009
actors : [
Brad Pitt
Diane Kruger
Eli Roth
]
title : The Hobbit: An Unexpected Journey
writer : J.R.R. Tolkein
year : 2012
franchise : The Hobbit

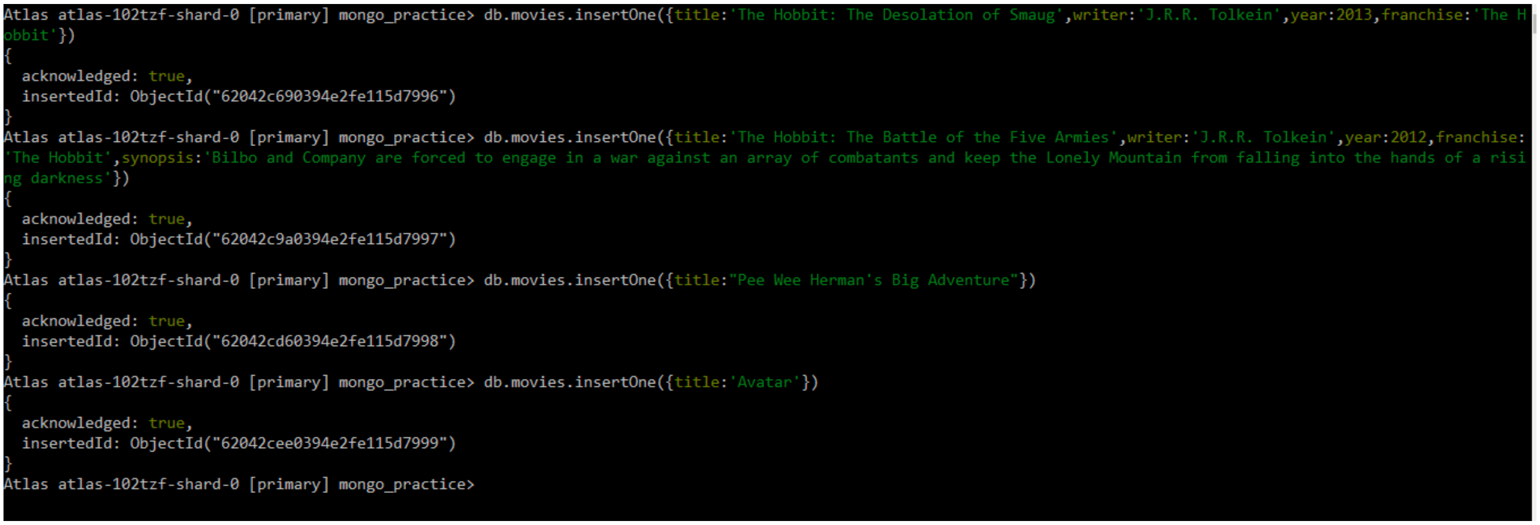
title : The Hobbit: The Desolation of Smaug
writer : J.R.R. Tolkein

year : 2013
franchise : The Hobbit
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.
title : Pee Wee Herman's Big Adventure
title : Avatar



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:'Fight Club',writer: 'Chuck Palahniuko',year : 1999,actors : ['Brad Pitt','Edward Norton
']})
{
  acknowledged: true,
  insertedId: ObjectId("62042a8a0394e2fe115d7992")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:'Pulp Fiction',writer: 'Quentin Tarantino',year : 1994,actors : ['John Travolta','Uma Th
urman']})
{
  acknowledged: true,
  insertedId: ObjectId("62042ae30394e2fe115d7993")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:'Inglorious Basterds',writer: 'Quentin Tarantino',year : 2009,actors : ['Brad Pitt','Dia
ne Kruger','Eli Roth']})
{
  acknowledged: true,
  insertedId: ObjectId("62042b2e0394e2fe115d7994")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
bit'})
{
  acknowledged: true,
  insertedId: ObjectId("62042b5f0394e2fe115d7995")
```

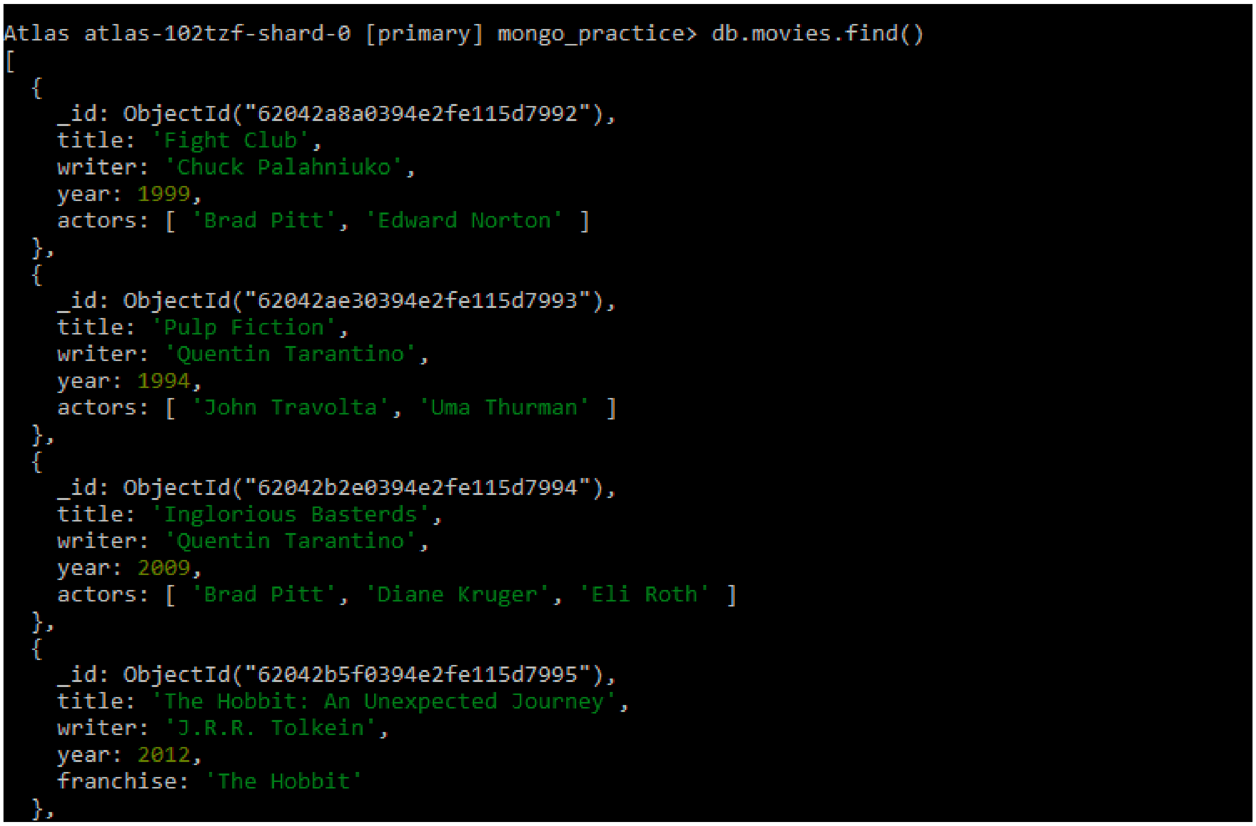


```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:'The Hobbit: The Desolation of Smaug',writer:'J.R.R. Tolkein',year:2013,franchise:'The H
obbit'})
{
  acknowledged: true,
  insertedId: ObjectId("62042c690394e2fe115d7996")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({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 risi
ng darkness'})
{
  acknowledged: true,
  insertedId: ObjectId("62042c9a0394e2fe115d7997")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:"Pee Wee Herman's Big Adventure"})
{
  acknowledged: true,
  insertedId: ObjectId("62042cd60394e2fe115d7998")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.insertOne({title:'Avatar'})
{
  acknowledged: true,
  insertedId: ObjectId("62042cee0394e2fe115d7999")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

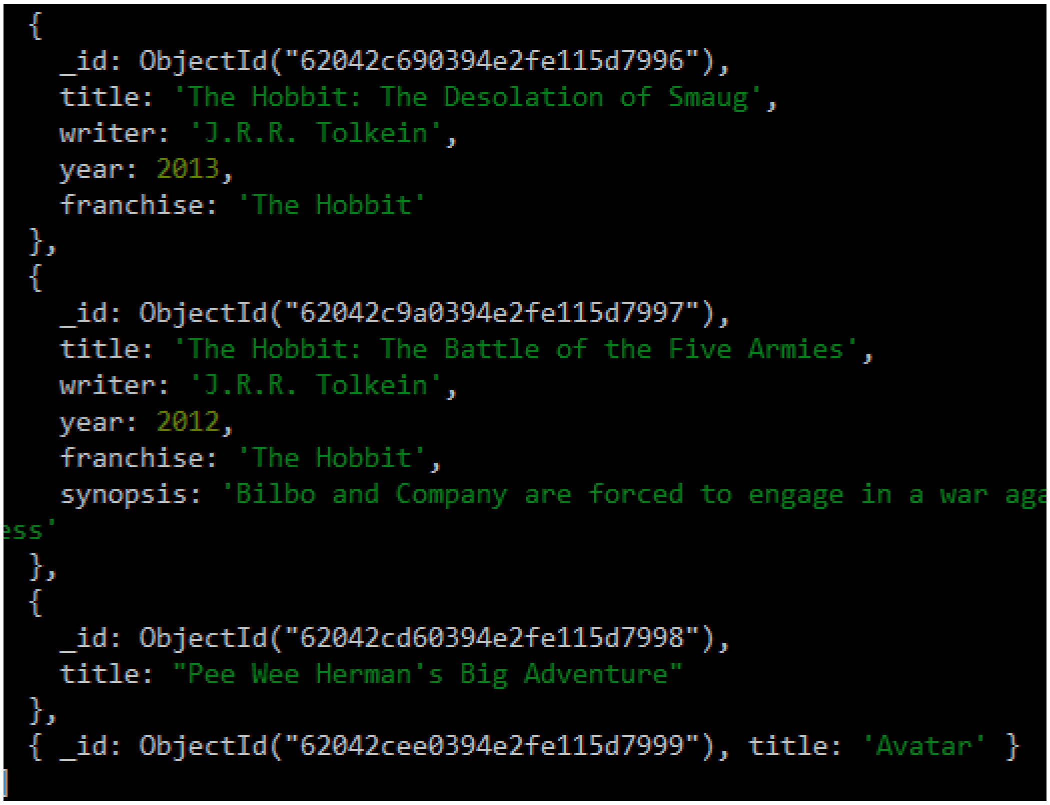
Query / Find Documents

query the movies collection to

1. get all documents

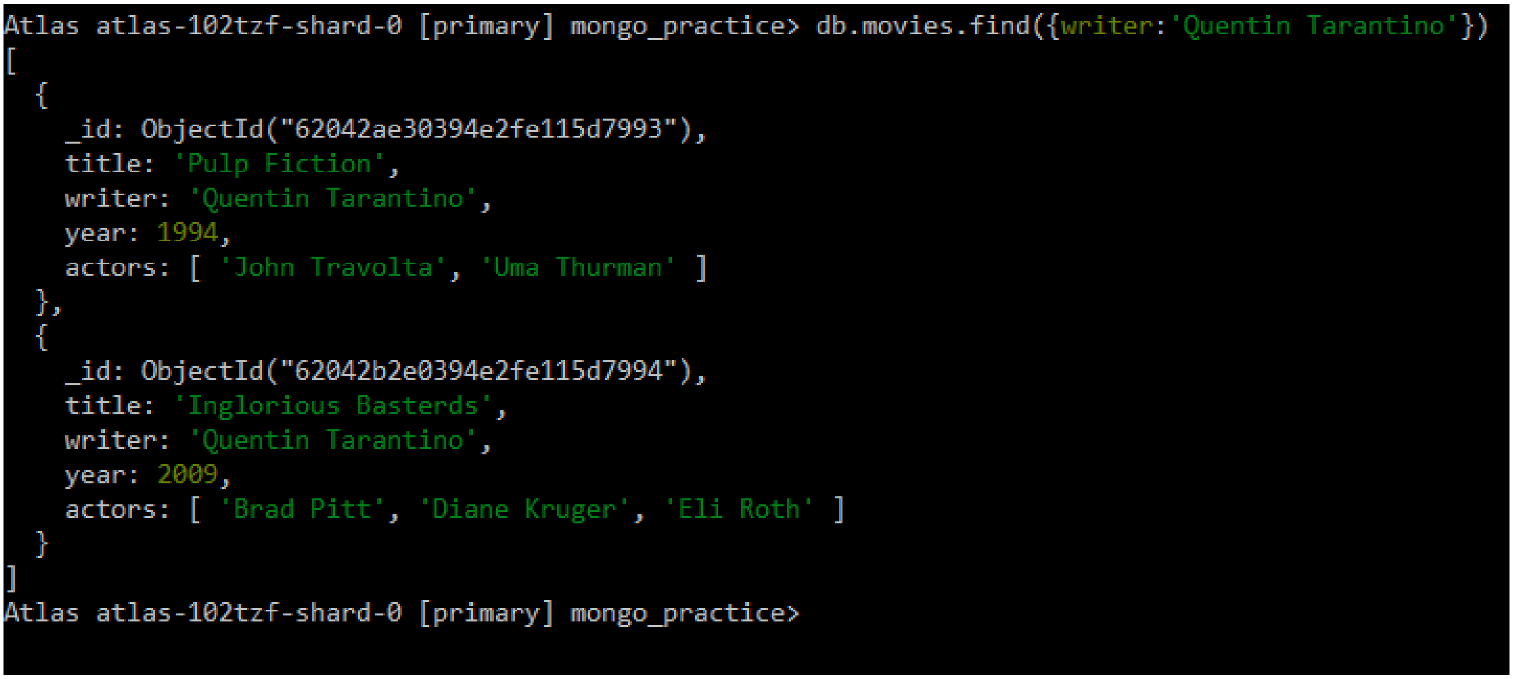


```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find()
[
  {
    _id: ObjectId("62042a8a0394e2fe115d7992"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62042ae30394e2fe115d7993"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("62042b2e0394e2fe115d7994"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  },
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
```



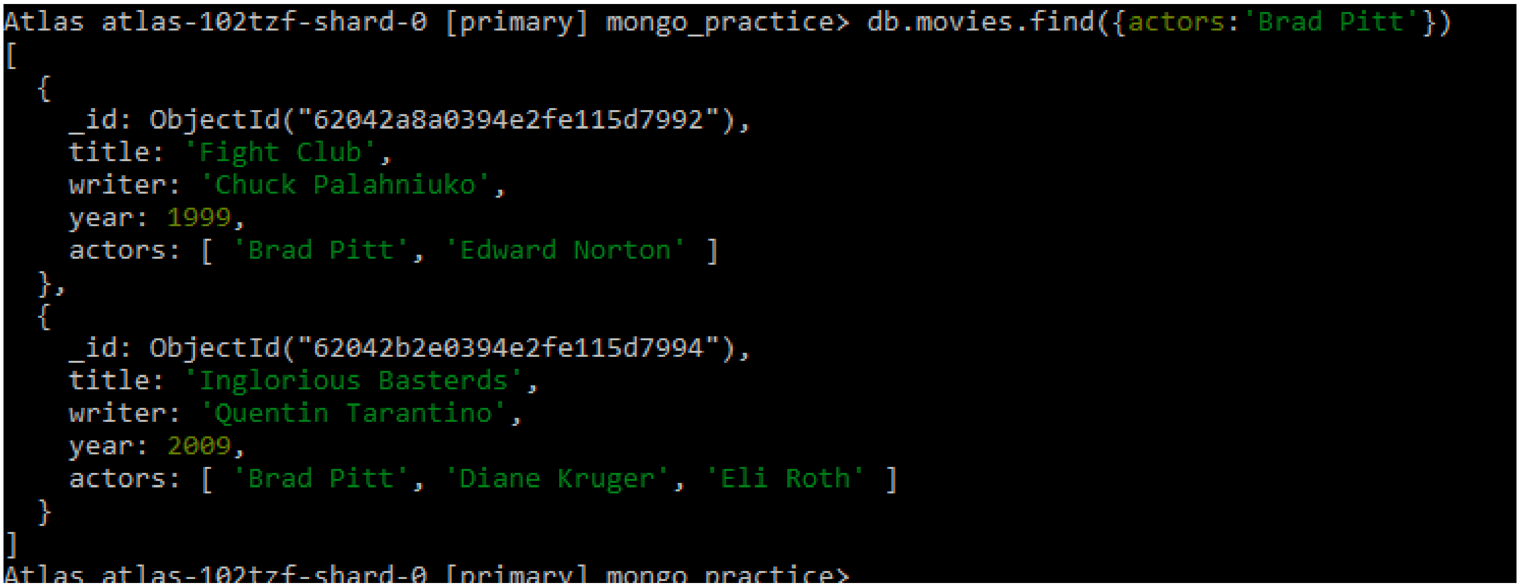
```
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("62042c9a0394e2fe115d7997"),
    title: 'The Hobbit: The Battle of the Five Armies',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war aga
ess'
  },
  {
    _id: ObjectId("62042cd60394e2fe115d7998"),
    title: "Pee Wee Herman's Big Adventure"
  },
  { _id: ObjectId("62042cee0394e2fe115d7999"), title: 'Avatar' }
]
```

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



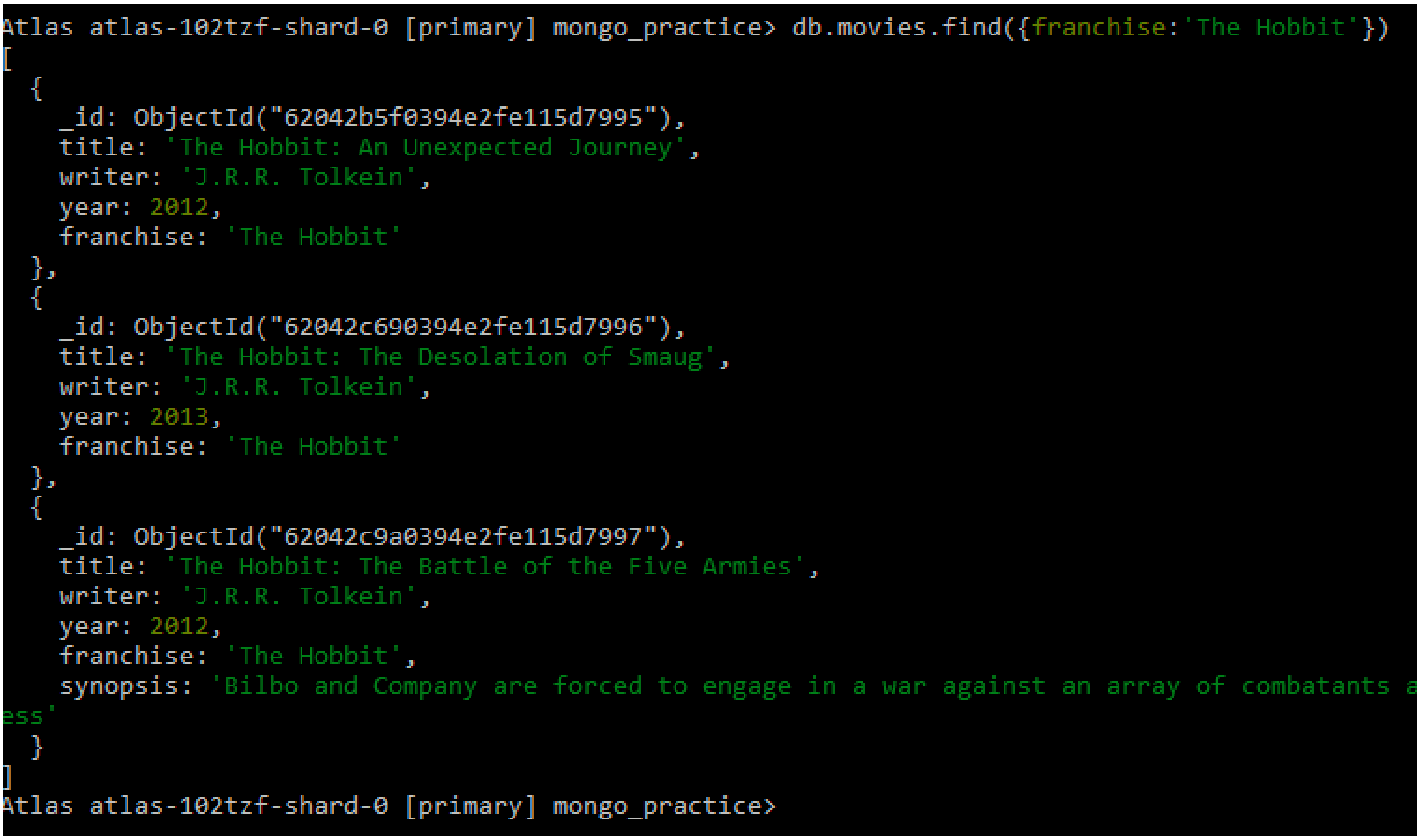
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({writer:'Quentin Tarantino'})
[
  {
    _id: ObjectId("62042ae30394e2fe115d7993"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("62042b2e0394e2fe115d7994"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

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



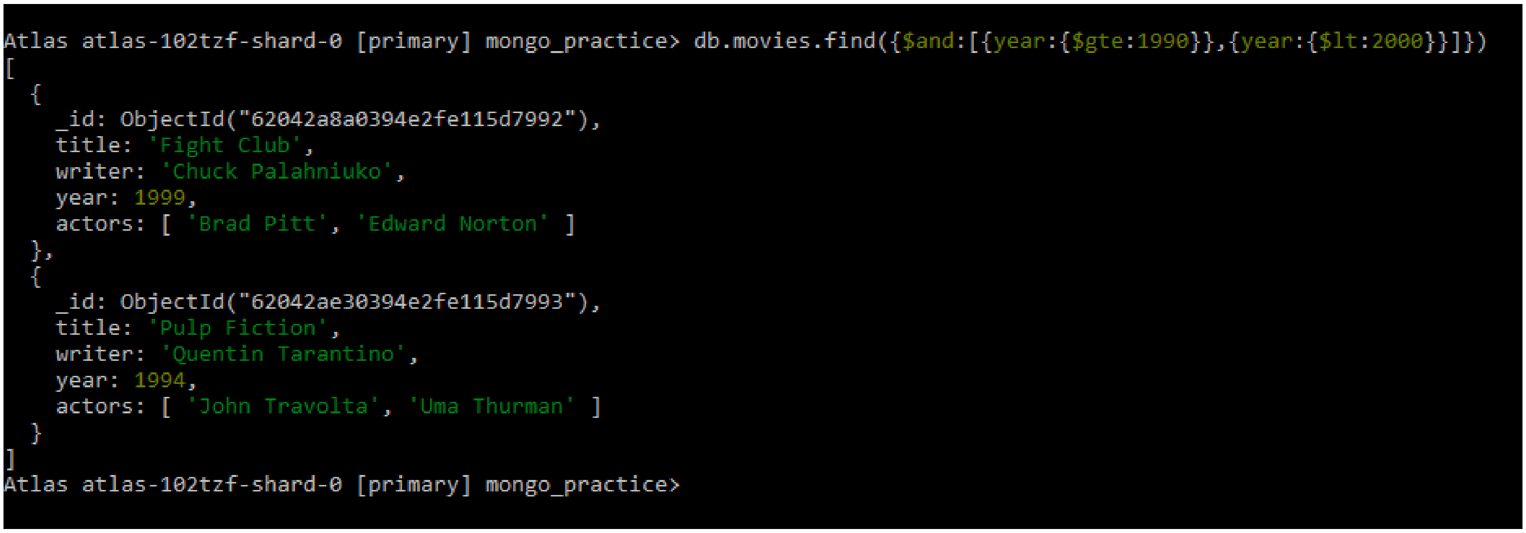
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({actors:'Brad Pitt'})
[
  {
    _id: ObjectId("62042a8a0394e2fe115d7992"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62042b2e0394e2fe115d7994"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

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



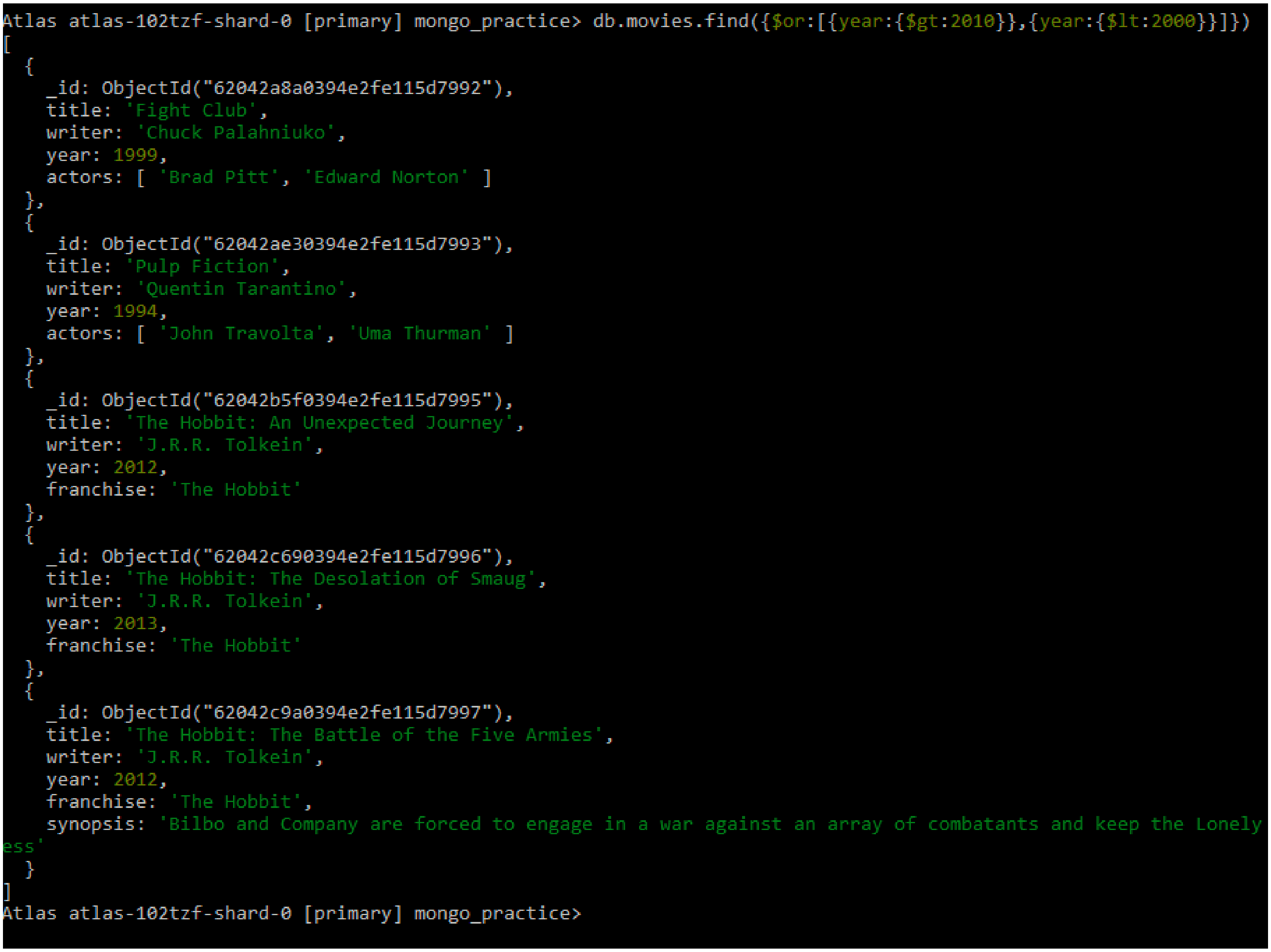
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({franchise:'The Hobbit'})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("62042c9a0394e2fe115d7997"),
    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 a
ess'
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

5. get all movies released in the 90s



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({$and:[{year:{$gte:1990}},{year:{$lt:2000}}]})
[
  {
    _id: ObjectId("62042a8a0394e2fe115d7992"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62042ae30394e2fe115d7993"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

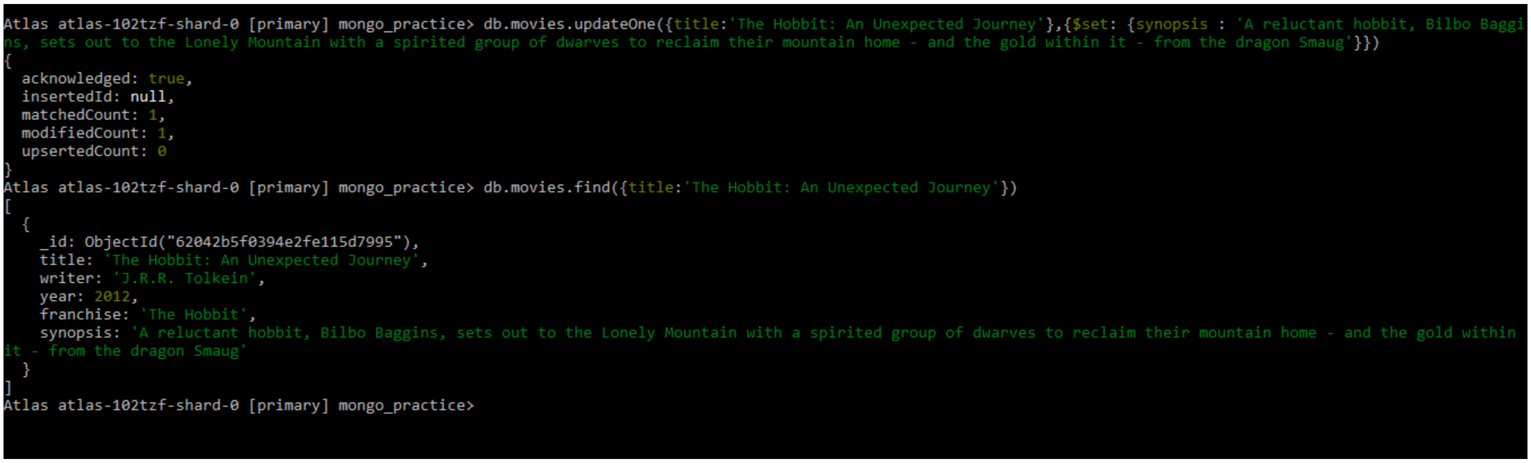
6. get all movies released before the year 2000 or after 2010



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({$or:[{year:{$gt:2010}},{year:{$lt:2000}}]})
[
  {
    _id: ObjectId("62042a8a0394e2fe115d7992"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("62042ae30394e2fe115d7993"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("62042c9a0394e2fe115d7997"),
    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
ess'
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

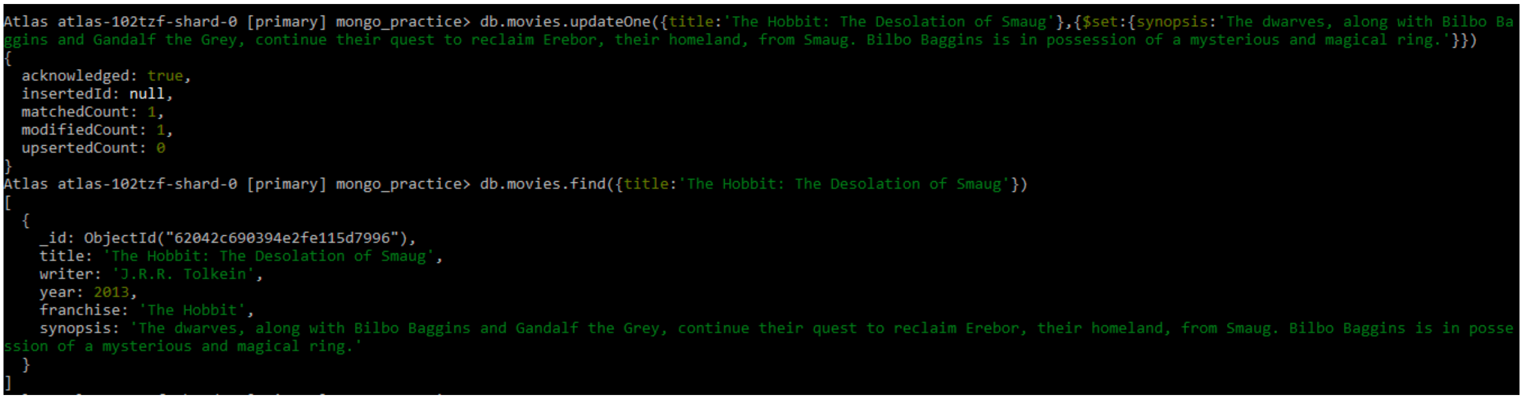
Update Documents

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."



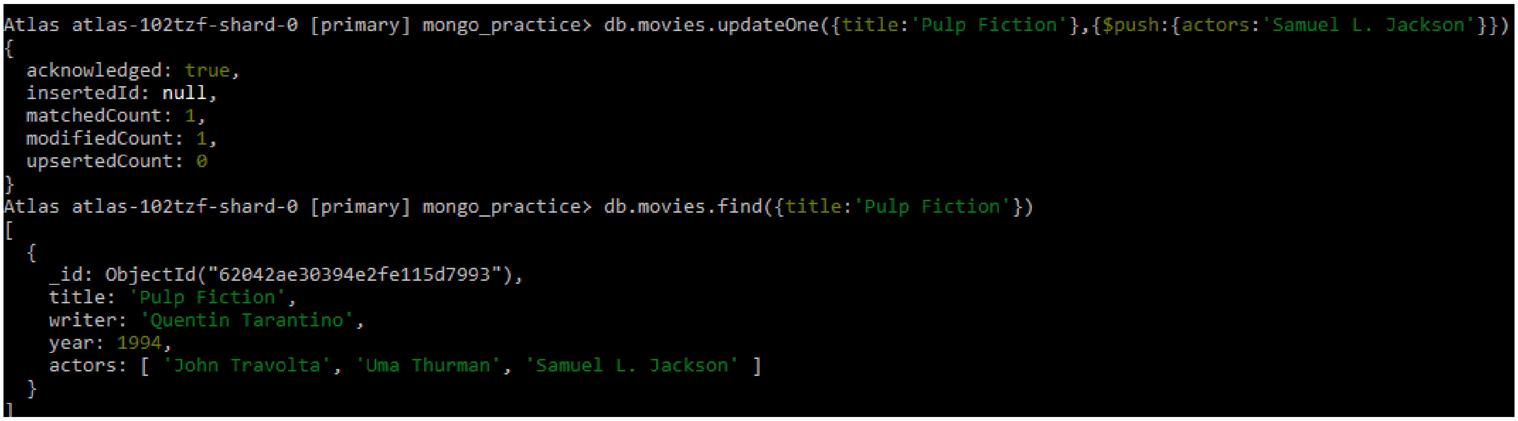
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.updateOne({title:'The Hobbit: An Unexpected Journey'},{$set: {synopsis : 'A reluctant hobbit, Bilbo Baggi
ns, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({title:'The Hobbit: An Unexpected Journey'})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    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-102tzf-shard-0 [primary] mongo_practice>
```

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."



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.updateOne({title:'The Hobbit: The Desolation of Smaug'},{$set:{synopsis:'The dwarves, along with Bilbo Ba
ggins 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.'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({title:'The Hobbit: The Desolation of Smaug'})
[
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    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 posse
ssion of a mysterious and magical ring.'
  }
]
```

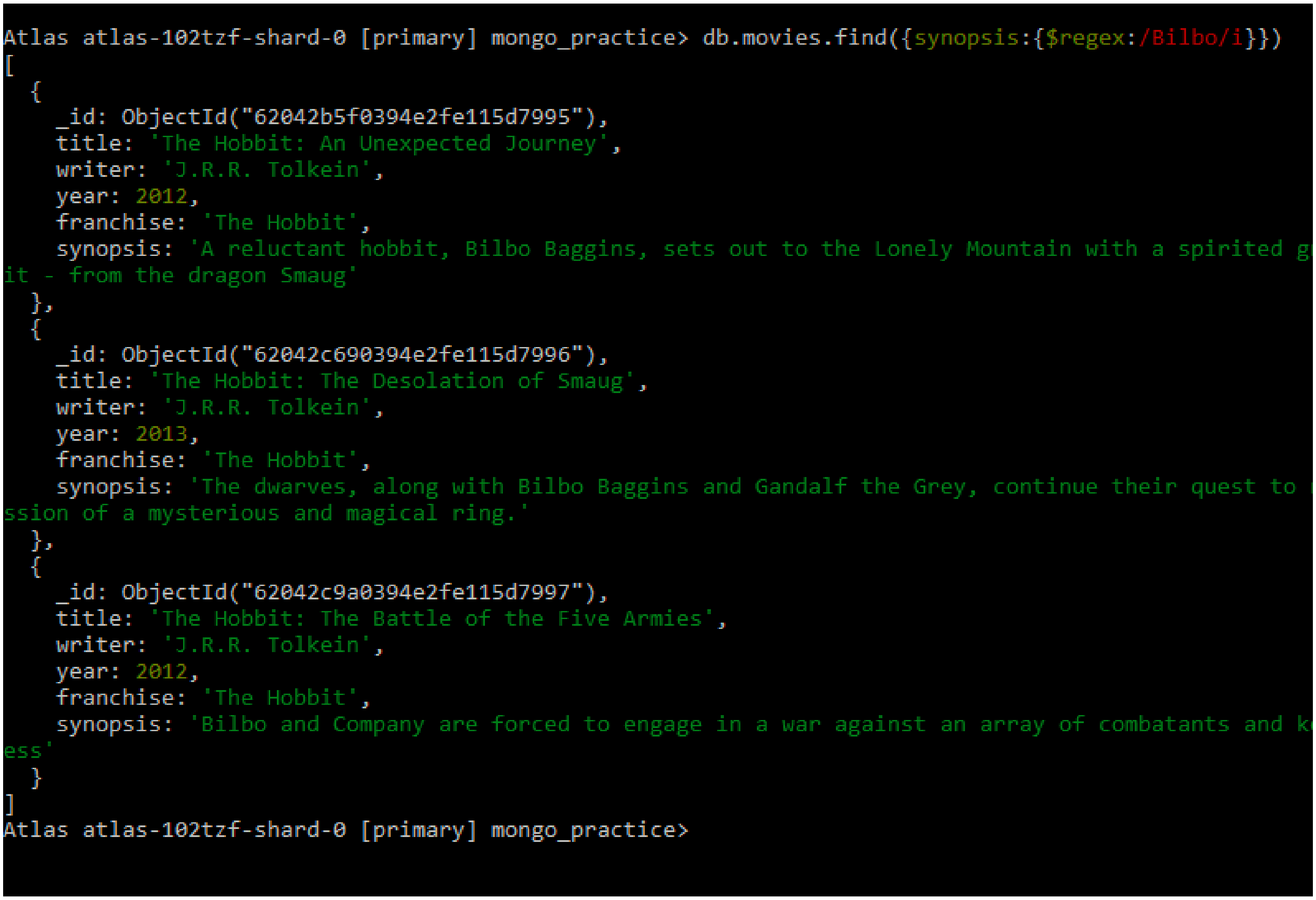
3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.updateOne({title:'Pulp Fiction'},{$push:{actors:'Samuel L. Jackson'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({title:'Pulp Fiction'})
[
  {
    _id: ObjectId("62042ae30394e2fe115d7993"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman', 'Samuel L. Jackson' ]
  }
]
```

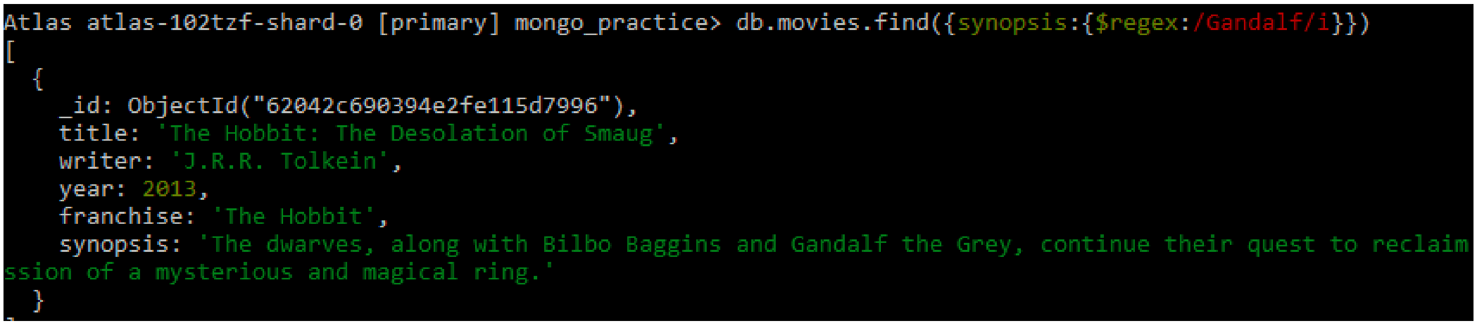
Text Search

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



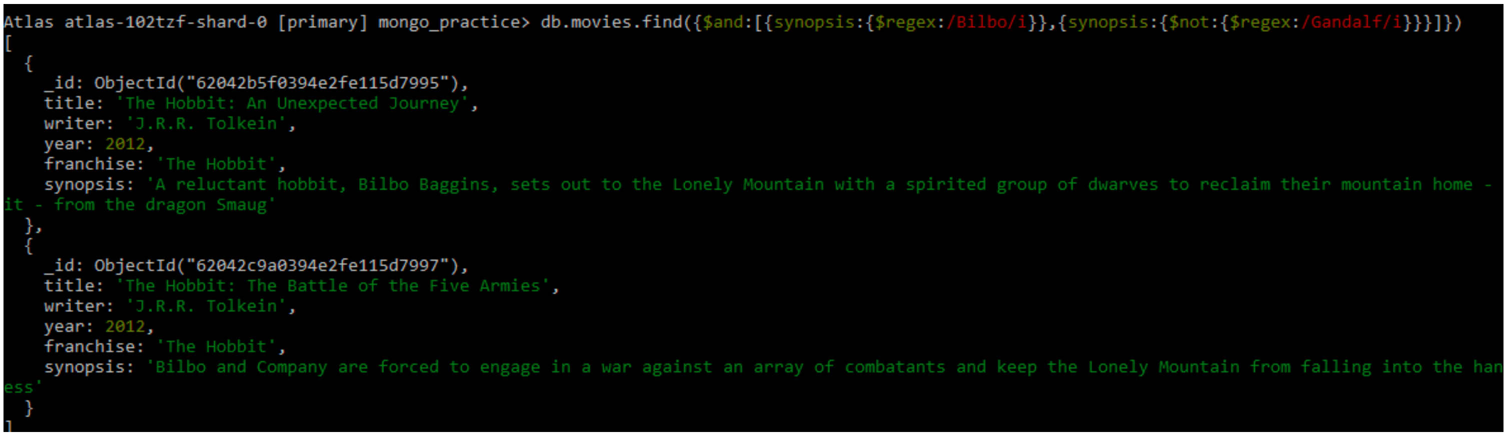
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({synopsis:{$regex:/Bilbo/i}})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited g
it - from the dragon Smaug'
  },
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    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
ssion of a mysterious and magical ring.'
  },
  {
    _id: ObjectId("62042c9a0394e2fe115d7997"),
    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 k
ess'
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

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



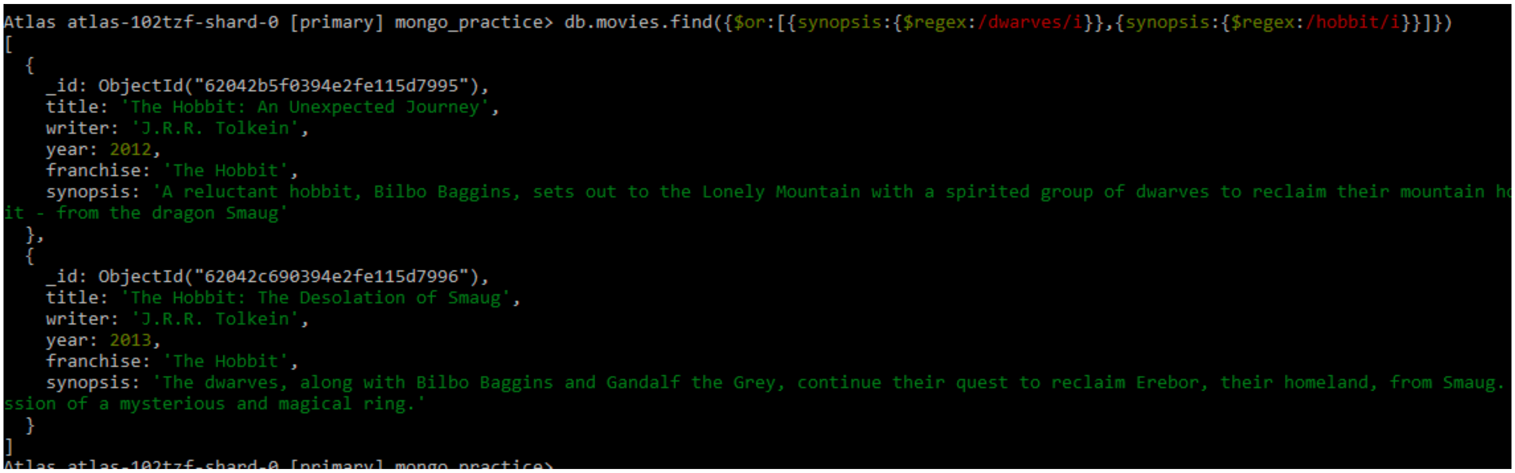
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({synopsis:{$regex:/Gandalf/i}})
[
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim
ssion of a mysterious and magical ring.'
  }
```

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



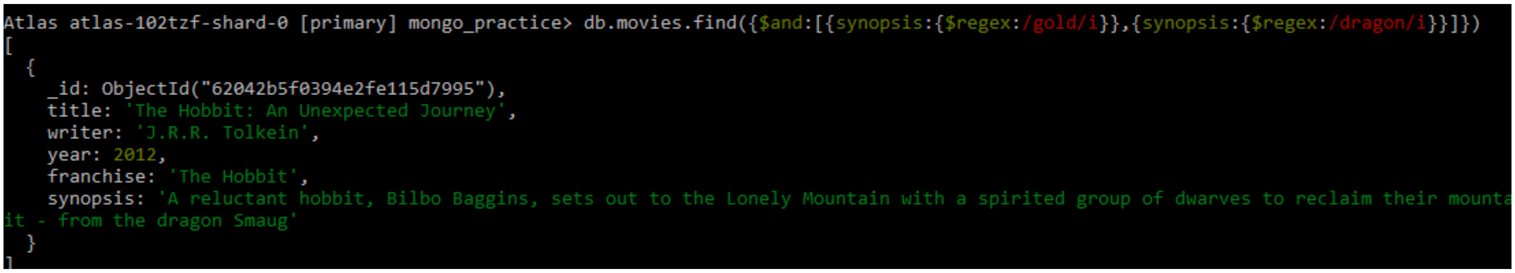
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:{$regex:/Bilbo/i}},{synopsis:{$not:{$regex:/Gandalf/i}}}]})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: '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 -
it - from the dragon Smaug'
  },
  {
    _id: ObjectId("62042c9a0394e2fe115d7997"),
    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 han
ess'
  }
]
```

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



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({$or:[{synopsis:{$regex:/dwarves/i}},{synopsis:{$regex:/hobbit/i}}]})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    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 ho
it - from the dragon Smaug'
  },
  {
    _id: ObjectId("62042c690394e2fe115d7996"),
    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.
ssion of a mysterious and magical ring.'
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

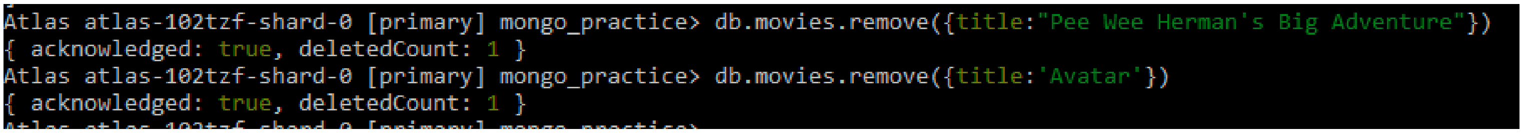
5. find all movies that have a synopsis that contains the word "gold" and "dragon"



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:{$regex:/gold/i}},{synopsis:{$regex:/dragon/i}}]})
[
  {
    _id: ObjectId("62042b5f0394e2fe115d7995"),
    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 mounta
it - from the dragon Smaug'
  }
]
```

Delete Documents

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

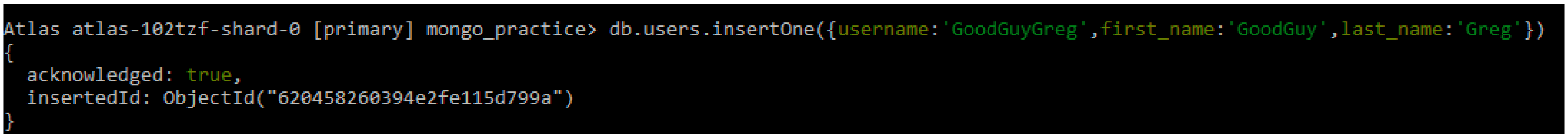


```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.remove({title:"Pee Wee Herman's Big Adventure"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.movies.remove({title:'Avatar'})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

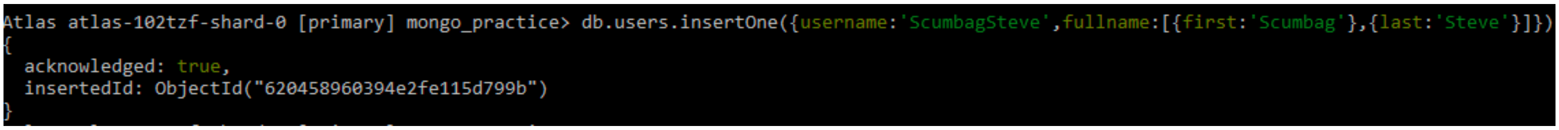
Relationships

Insert the following documents into a users collection

username : GoodGuyGreg
first_name : "Good Guy"
last_name : "Greg"
username : ScumbagSteve
full_name :
first : "Scumbag"
last : "Steve"



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.users.insertOne({username:'GoodGuyGreg',first_name:'GoodGuy',last_name:'Greg'})
{
  acknowledged: true,
  insertedId: ObjectId("620458260394e2fe115d799a")
}
```

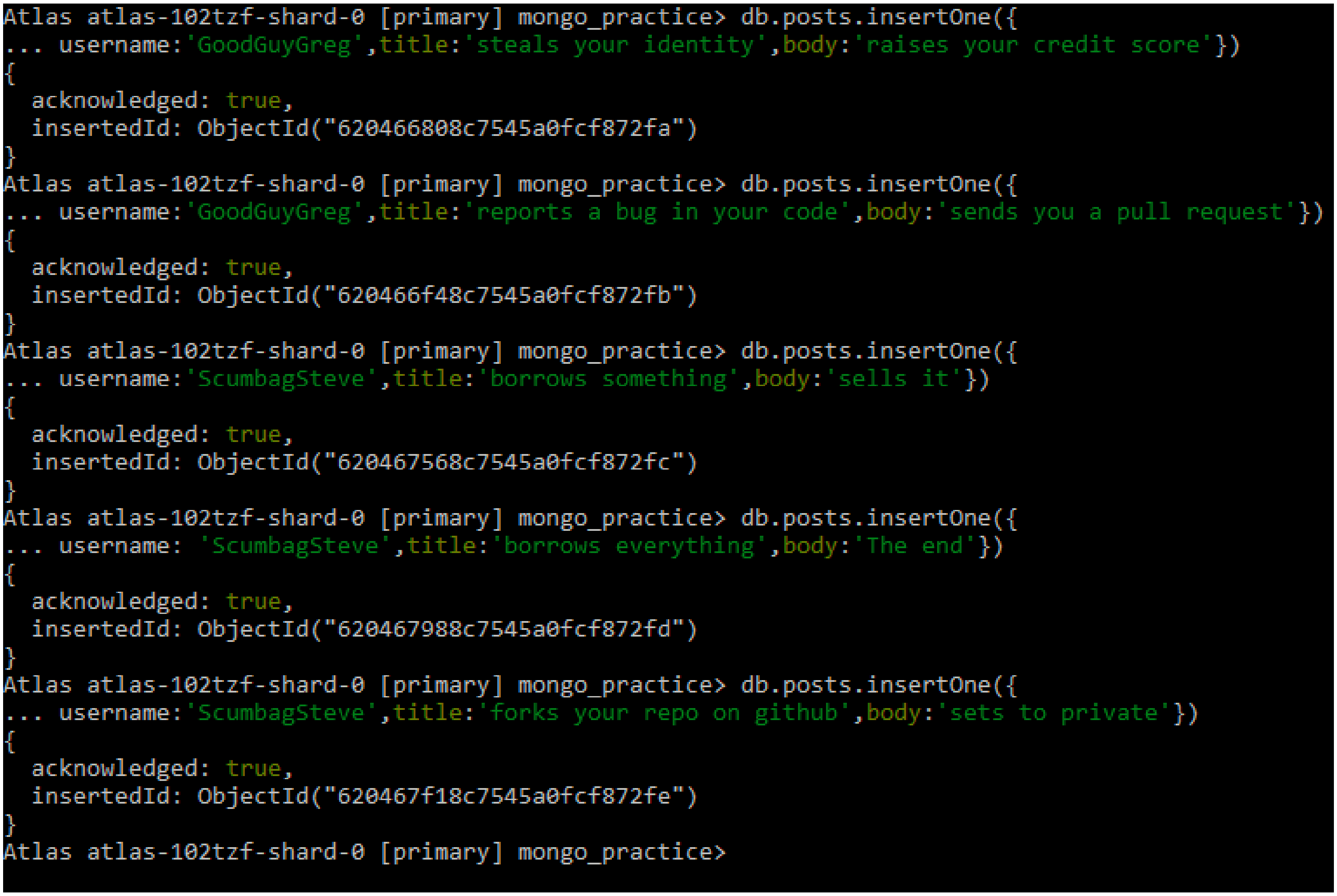


```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.users.insertOne({username:'ScumbagSteve',fullname:[{first:'Scumbag'},{last:'Steve'}]})
{
  acknowledged: true,
  insertedId: ObjectId("620458960394e2fe115d799b")
}
```

Insert the following documents into a posts collection
username : GoodGuyGreg
title : Passes out at party

body : Wakes up early and cleans house

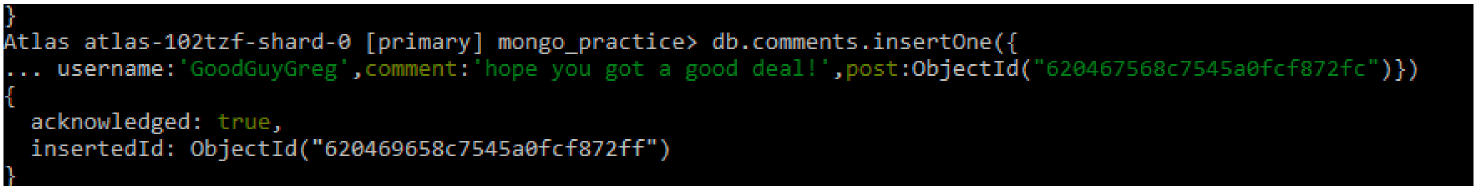
username : GoodGuyGreg
title : Steals your identity
body : Raises your credit score
username : GoodGuyGreg
title : Reports a bug in your code
body : Sends you a Pull Request
username : ScumbagSteve
title : Borrows something
body : Sells it
username : ScumbagSteve
title : Borrows everything
body : The end
username : ScumbagSteve
title : Forks your repo on github
body : Sets to private



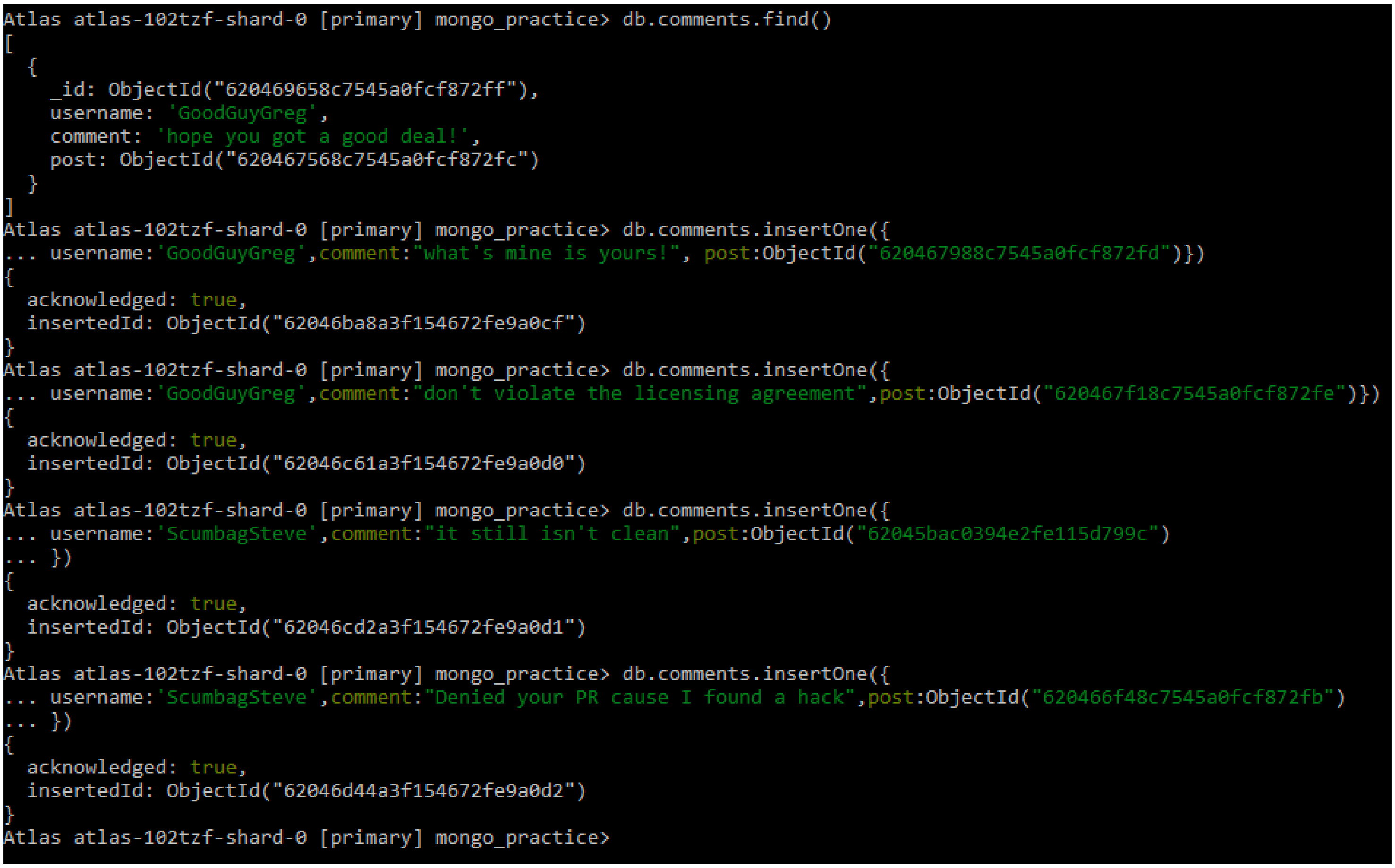
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.insertOne({
... username:'GoodGuyGreg',title:'steals your identity',body:'raises your credit score'})
{
  acknowledged: true,
  insertedId: ObjectId("620466808c7545a0fcf872fa")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.insertOne({
... username:'GoodGuyGreg',title:'reports a bug in your code',body:'sends you a pull request'})
{
  acknowledged: true,
  insertedId: ObjectId("620466f48c7545a0fcf872fb")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.insertOne({
... username:'ScumbagSteve',title:'borrows something',body:'sells it'})
{
  acknowledged: true,
  insertedId: ObjectId("620467568c7545a0fcf872fc")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.insertOne({
... username: 'ScumbagSteve',title:'borrows everything',body:'The end'})
{
  acknowledged: true,
  insertedId: ObjectId("620467988c7545a0fcf872fd")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.insertOne({
... username:'ScumbagSteve',title:'forks your repo on github',body:'sets to private'})
{
  acknowledged: true,
  insertedId: ObjectId("620467f18c7545a0fcf872fe")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

Insert the following documents into a comments collection

username : GoodGuyGreg
comment : Hope you got a good deal!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Borrows something"
username : GoodGuyGreg
comment : What's mine is yours!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Borrows everything"
username : GoodGuyGreg
comment : Don't violate the licensing agreement!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Forks your repo on github
username : ScumbagSteve
comment : It still isn't clean
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Passes out at party"
username : ScumbagSteve
comment : Denied your PR cause I found a hack
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Reports a bug in your code"



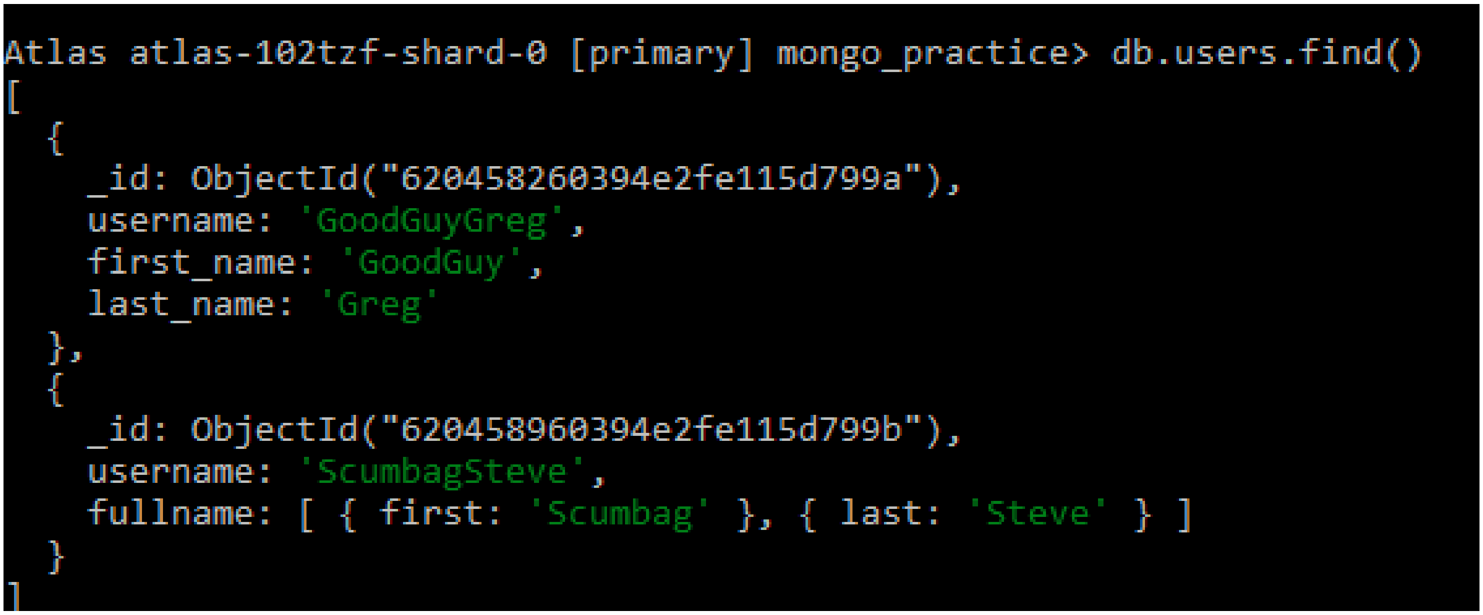
```
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.insertOne({
... username:'GoodGuyGreg',comment:'hope you got a good deal!',post:ObjectId("620467568c7545a0fcf872fc")})
{
  acknowledged: true,
  insertedId: ObjectId("620469658c7545a0fcf872ff")
}
```



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.find()
[
  {
    _id: ObjectId("620469658c7545a0fcf872ff"),
    username: 'GoodGuyGreg',
    comment: 'hope you got a good deal!',
    post: ObjectId("620467568c7545a0fcf872fc")
  }
]
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.insertOne({
... username:'GoodGuyGreg',comment:"what's mine is yours!", post:ObjectId("620467988c7545a0fcf872fd")})
{
  acknowledged: true,
  insertedId: ObjectId("62046ba8a3f154672fe9a0cf")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.insertOne({
... username:'GoodGuyGreg',comment:"don't violate the licensing agreement",post:ObjectId("620467f18c7545a0fcf872fe")})
{
  acknowledged: true,
  insertedId: ObjectId("62046c61a3f154672fe9a0d0")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.insertOne({
... username:'ScumbagSteve',comment:"it still isn't clean",post:ObjectId("62045bac0394e2fe115d799c")
... })
{
  acknowledged: true,
  insertedId: ObjectId("62046cd2a3f154672fe9a0d1")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.insertOne({
... username:'ScumbagSteve',comment:"Denied your PR cause I found a hack",post:ObjectId("620466f48c7545a0fcf872fb")
... })
{
  acknowledged: true,
  insertedId: ObjectId("62046d44a3f154672fe9a0d2")
}
Atlas atlas-102tzf-shard-0 [primary] mongo_practice>
```

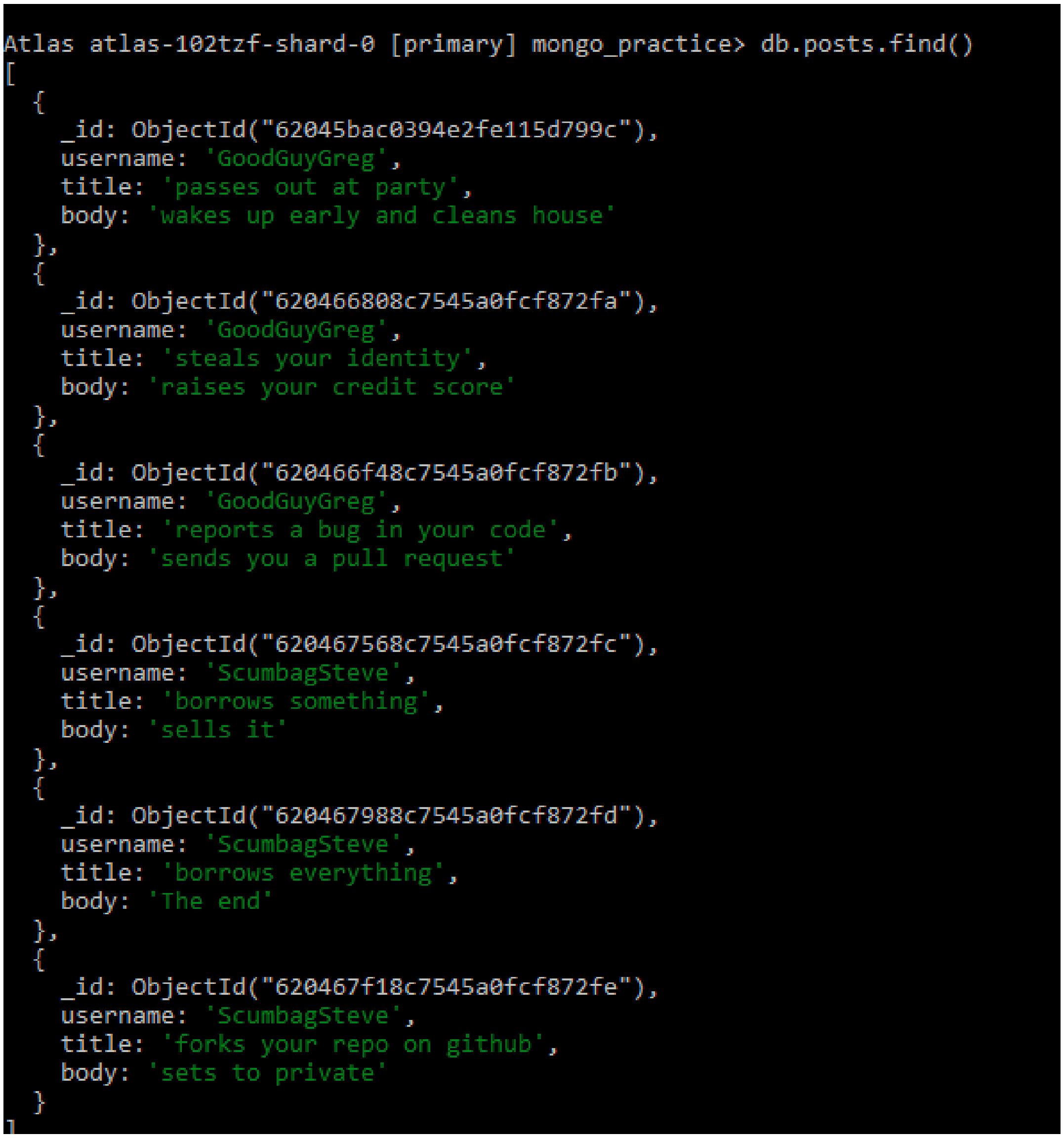
Querying related collections

1. find all users



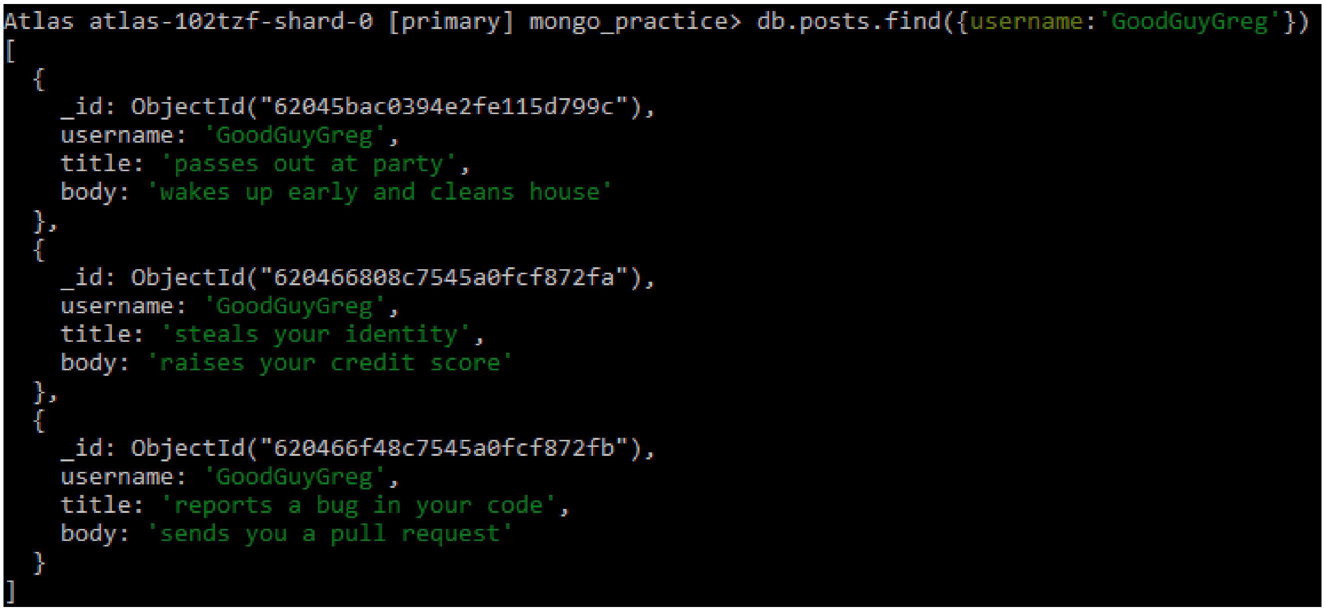
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.users.find()
[
  {
    _id: ObjectId("620458260394e2fe115d799a"),
    username: 'GoodGuyGreg',
    first_name: 'GoodGuy',
    last_name: 'Greg'
  },
  {
    _id: ObjectId("620458960394e2fe115d799b"),
    username: 'ScumbagSteve',
    fullname: [ { first: 'Scumbag' }, { last: 'Steve' } ]
  }
]
```

2. find all posts



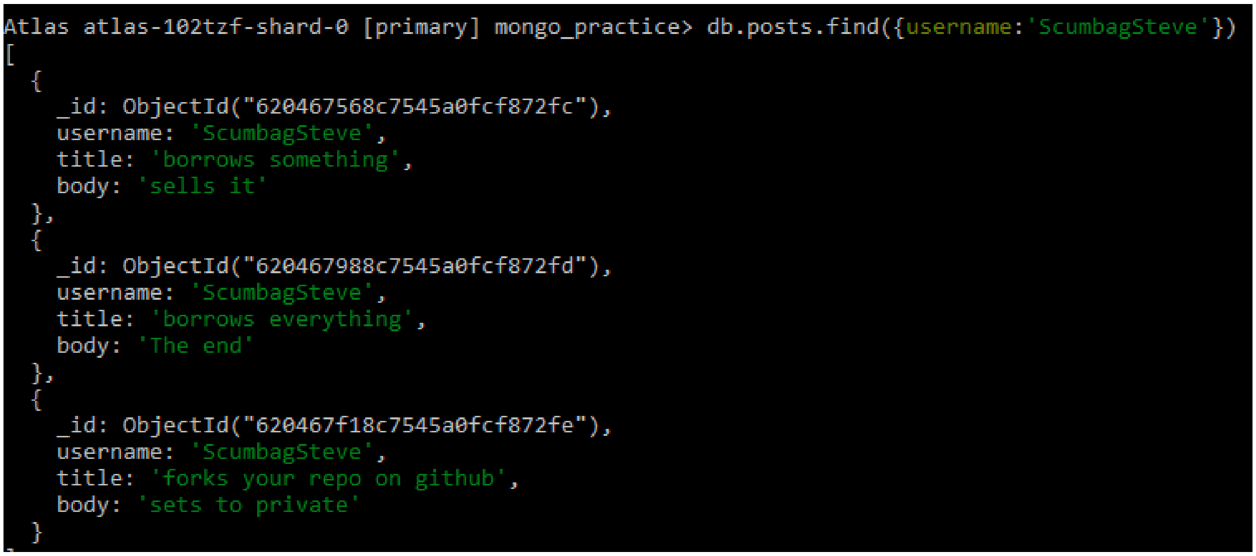
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.find()
[
  {
    _id: ObjectId("62045bac0394e2fe115d799c"),
    username: 'GoodGuyGreg',
    title: 'passes out at party',
    body: 'wakes up early and cleans house'
  },
  {
    _id: ObjectId("620466808c7545a0fcf872fa"),
    username: 'GoodGuyGreg',
    title: 'steals your identity',
    body: 'raises your credit score'
  },
  {
    _id: ObjectId("620466f48c7545a0fcf872fb"),
    username: 'GoodGuyGreg',
    title: 'reports a bug in your code',
    body: 'sends you a pull request'
  },
  {
    _id: ObjectId("620467568c7545a0fcf872fc"),
    username: 'ScumbagSteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("620467988c7545a0fcf872fd"),
    username: 'ScumbagSteve',
    title: 'borrows everything',
    body: 'The end'
  },
  {
    _id: ObjectId("620467f18c7545a0fcf872fe"),
    username: 'ScumbagSteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  }
]
```

3. find all posts that was authored by "GoodGuyGreg"



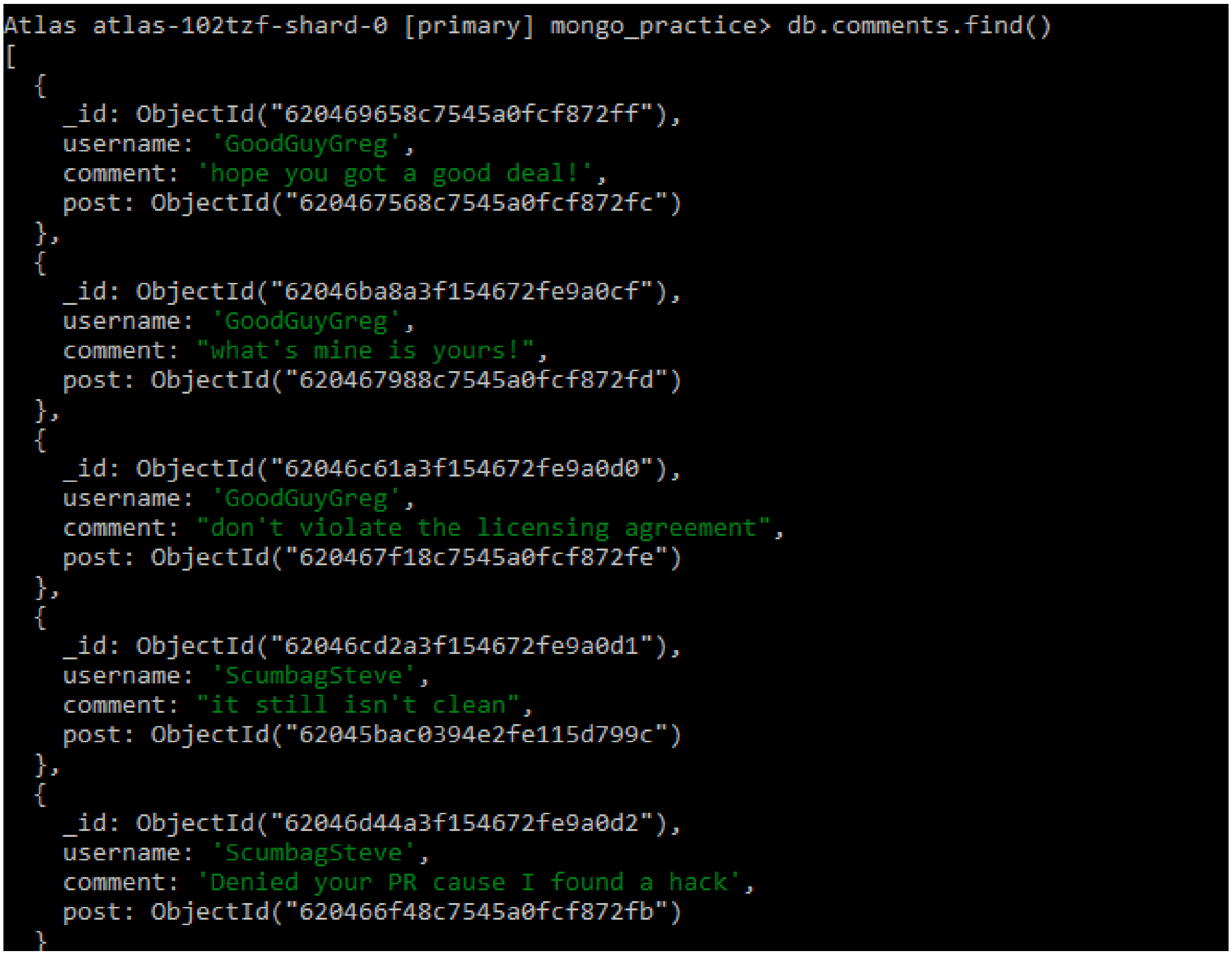
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.find({username:'GoodGuyGreg'})
[
  {
    _id: ObjectId("62045bac0394e2fe115d799c"),
    username: 'GoodGuyGreg',
    title: 'passes out at party',
    body: 'wakes up early and cleans house'
  },
  {
    _id: ObjectId("620466808c7545a0fcf872fa"),
    username: 'GoodGuyGreg',
    title: 'steals your identity',
    body: 'raises your credit score'
  },
  {
    _id: ObjectId("620466f48c7545a0fcf872fb"),
    username: 'GoodGuyGreg',
    title: 'reports a bug in your code',
    body: 'sends you a pull request'
  }
]
```

4. find all posts that was authored by "ScumbagSteve"



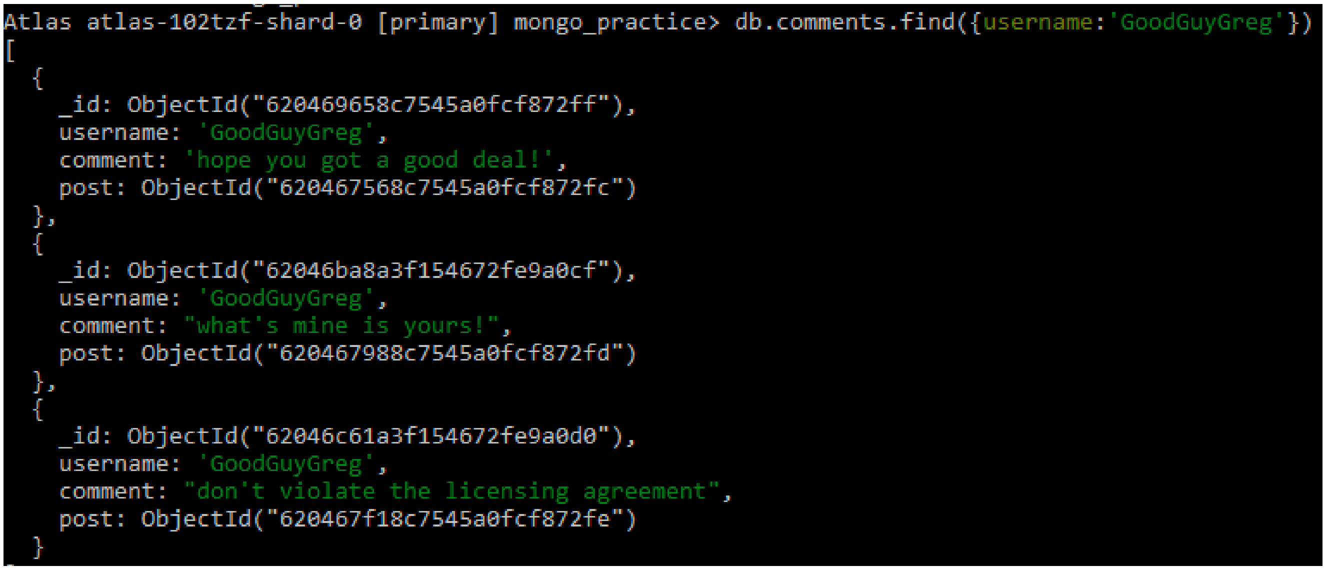
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.posts.find({username:'ScumbagSteve'})
[
  {
    _id: ObjectId("620467568c7545a0fcf872fc"),
    username: 'ScumbagSteve',
    title: 'borrows something',
    body: 'sells it'
  },
  {
    _id: ObjectId("620467988c7545a0fcf872fd"),
    username: 'ScumbagSteve',
    title: 'borrows everything',
    body: 'The end'
  },
  {
    _id: ObjectId("620467f18c7545a0fcf872fe"),
    username: 'ScumbagSteve',
    title: 'forks your repo on github',
    body: 'sets to private'
  }
```

5. find all comments



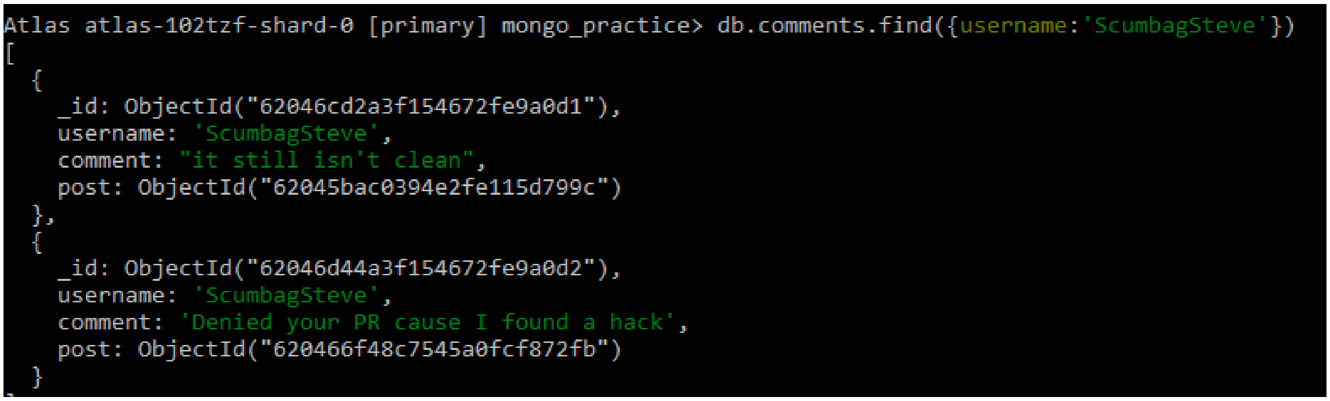
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.find()
[
  {
    _id: ObjectId("620469658c7545a0fcf872ff"),
    username: 'GoodGuyGreg',
    comment: 'hope you got a good deal!',
    post: ObjectId("620467568c7545a0fcf872fc")
  },
  {
    _id: ObjectId("62046ba8a3f154672fe9a0cf"),
    username: 'GoodGuyGreg',
    comment: "what's mine is yours!",
    post: ObjectId("620467988c7545a0fcf872fd")
  },
  {
    _id: ObjectId("62046c61a3f154672fe9a0d0"),
    username: 'GoodGuyGreg',
    comment: "don't violate the licensing agreement",
    post: ObjectId("620467f18c7545a0fcf872fe")
  },
  {
    _id: ObjectId("62046cd2a3f154672fe9a0d1"),
    username: 'ScumbagSteve',
    comment: "it still isn't clean",
    post: ObjectId("62045bac0394e2fe115d799c")
  },
  {
    _id: ObjectId("62046d44a3f154672fe9a0d2"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("620466f48c7545a0fcf872fb")
  }
```

6. find all comments that was authored by "GoodGuyGreg"



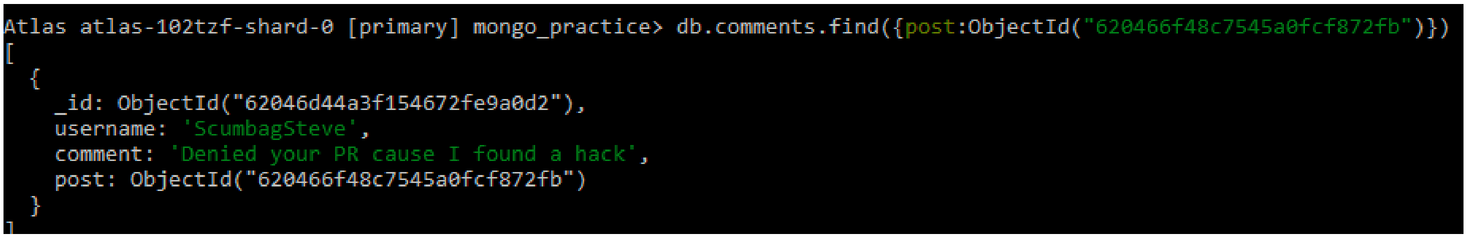
```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.find({username:'GoodGuyGreg'})
[
  {
    _id: ObjectId("620469658c7545a0fcf872ff"),
    username: 'GoodGuyGreg',
    comment: 'hope you got a good deal!',
    post: ObjectId("620467568c7545a0fcf872fc")
  },
  {
    _id: ObjectId("62046ba8a3f154672fe9a0cf"),
    username: 'GoodGuyGreg',
    comment: "what's mine is yours!",
    post: ObjectId("620467988c7545a0fcf872fd")
  },
  {
    _id: ObjectId("62046c61a3f154672fe9a0d0"),
    username: 'GoodGuyGreg',
    comment: "don't violate the licensing agreement",
    post: ObjectId("620467f18c7545a0fcf872fe")
  }
```

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



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.find({username:'ScumbagSteve'})
[
  {
    _id: ObjectId("62046cd2a3f154672fe9a0d1"),
    username: 'ScumbagSteve',
    comment: "it still isn't clean",
    post: ObjectId("62045bac0394e2fe115d799c")
  },
  {
    _id: ObjectId("62046d44a3f154672fe9a0d2"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("620466f48c7545a0fcf872fb")
  }
```

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



```
Atlas atlas-102tzf-shard-0 [primary] mongo_practice> db.comments.find({post:ObjectId("620466f48c7545a0fcf872fb")})
[
  {
    _id: ObjectId("62046d44a3f154672fe9a0d2"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("620466f48c7545a0fcf872fb")
  }
]
```