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

Query the movies collection to

1. Get all documents

```
use Anusha
'switched to db Anusha'
db.movies.find()
{ _id: ObjectId("638d8f91b26788f1ac2094b4"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuko',
  year: 'int',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ _id: ObjectId("638d9215b26788f1ac2094b5"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: 'int',
  actors: [ 'John Travolta', 'Thurman' ] }
{ _id: ObjectId("638d9334b26788f1ac2094b6"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: 'int',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
{ _id: ObjectId("638d943bb26788f1ac2094b7"),
  title: 'The Hobbit: An Unexpected Journey',
  title: 'The Hobbit: An Unexpected Journey',
  writer: 'J.R.R. Tolkein',
  year: 'int',
  franchise: 'The Hobbit' }
{ _id: ObjectId("638d9a05b26788f1ac2094ba"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkein',
  year: 'int',
  franchies: '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 risin
{ _id: ObjectId("638d9c4ab26788f1ac2094bb"),
  'title ': 'Pee Wee Herman\'s Big Adventure' }
{ _id: ObjectId("638d9c99b26788f1ac2094bc"), title: 'Avatar' }
tlas atlas-s45y9d-shard-0 [primary] Anusha>
```

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

```
> db.movies.find({writer:"Quentin Tarantino"})
< { _id: ObjectId("638d9215b26788f1ac2094b5"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 'int',
    actors: [ 'John Travolta', 'Thurman' ] }
  { _id: ObjectId("638d9334b26788f1ac2094b6"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 'int',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
Atlas atlas-s45y9d-shard-0 [primary] Anusha>
```

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

```
> db.movies.find({actors:"Brad Pitt"})
< { _id: ObjectId("638d8f91b26788f1ac2094b4"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 'int',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("638d9334b26788f1ac2094b6"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 'int',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
Atlas atlas-s45y9d-shard-0 [primary] Anusha>
```

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

```
> db.movies.find({franchise:"The Hobbit"})
< { _id: ObjectId("638d943bb26788f1ac2094b7"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 'int',
    franchise: 'The Hobbit' }
Atlas atlas-s45y9d-shard-0 [primary] Anusha>
```

5. get all movies released in the 90s

```
> db.movies.find({year:{$gt:"1990", $lt:"2000"}})
< { _id: ObjectId("638d8f91b26788f1ac2094b4"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("638d9334b26788f1ac2094b6"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
```

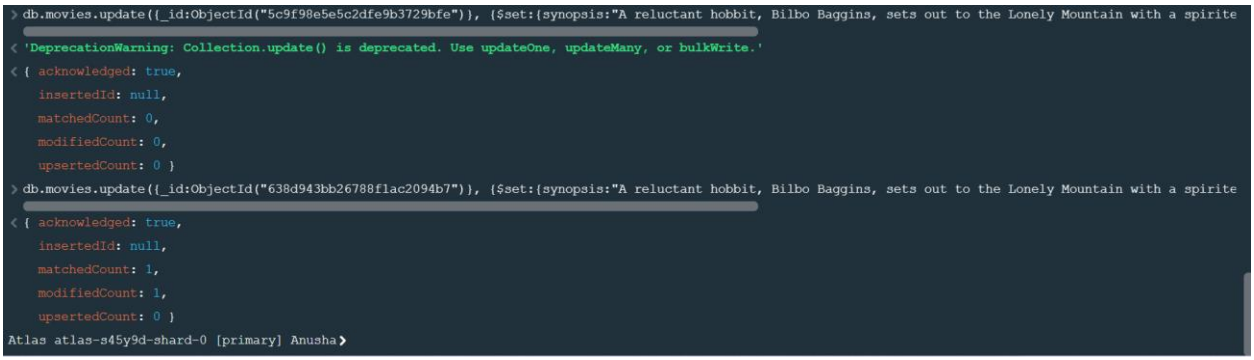
6. get all movies released before the year 2000 or after 2010

```
> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})
< { _id: ObjectId("638d8f91b26788f1ac2094b4"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("638d9215b26788f1ac2094b5"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 'int',
    actors: [ 'John Travolta', 'Thurman' ] }
  { _id: ObjectId("638d9334b26788f1ac2094b6"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
  { _id: ObjectId("638d943bb26788f1ac2094b7"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
```

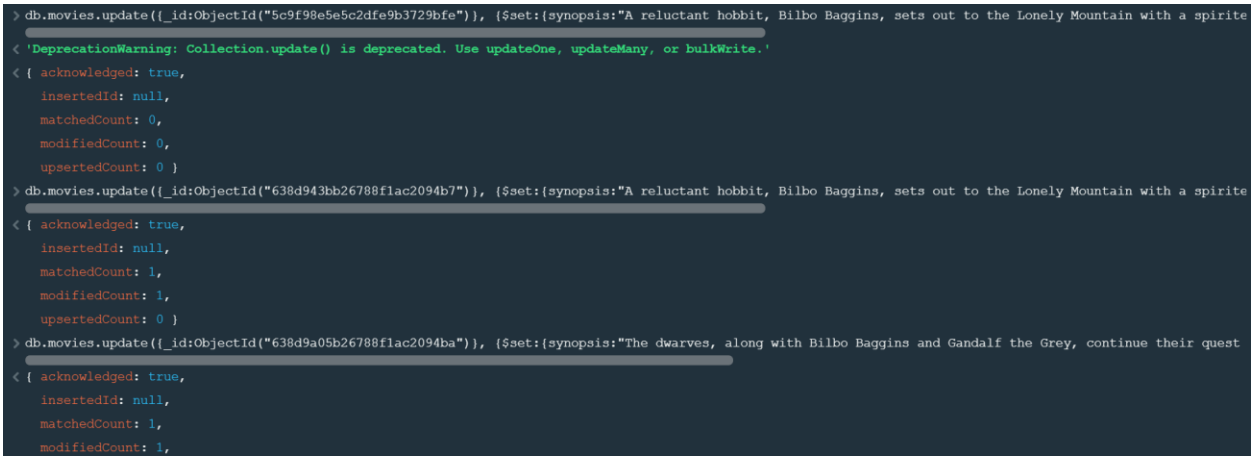
```
{ _id: ObjectId("638d943bb26788f1ac2094b7"),
  title: 'The Hobbit: An Unexpected Journey',
  writer: 'J.R.R. Tolkein',
  year: '2012',
  franchise: 'The Hobbit',
  synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold
{ _id: ObjectId("638d9a05b26788f1ac2094ba"),
  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 i
{ _id: ObjectId("638dc463b26788f1ac2094bd"),
  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 risin
```

Update Documents

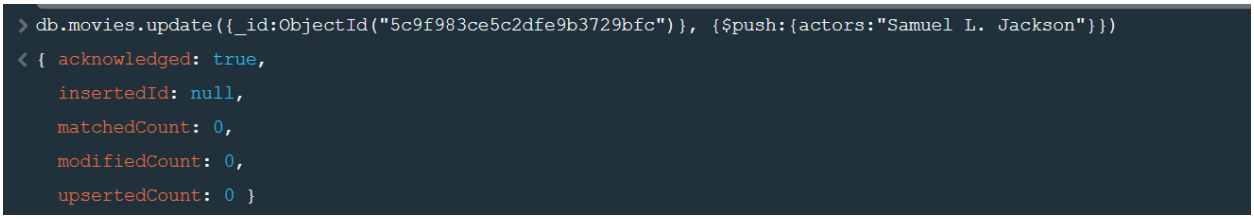
1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."



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."

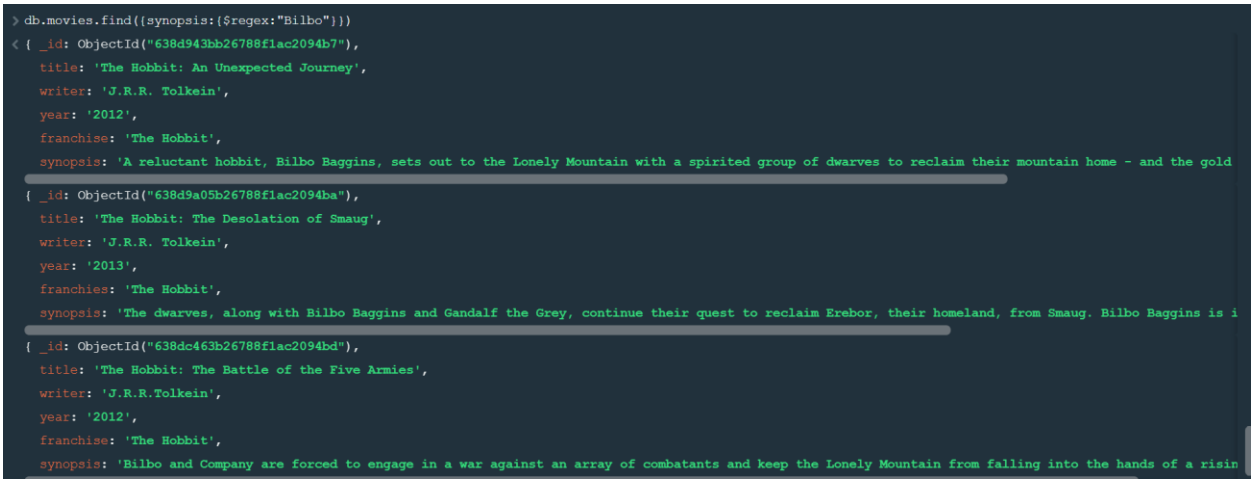


3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

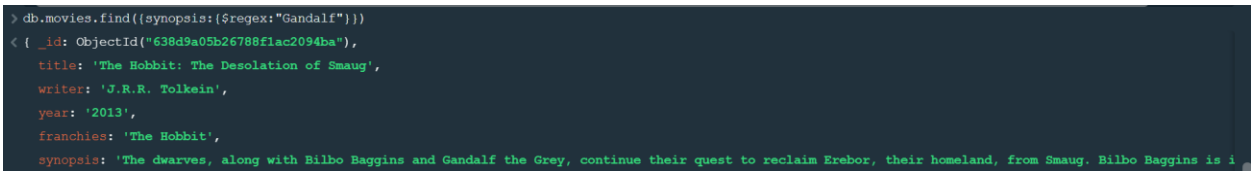


Text Search

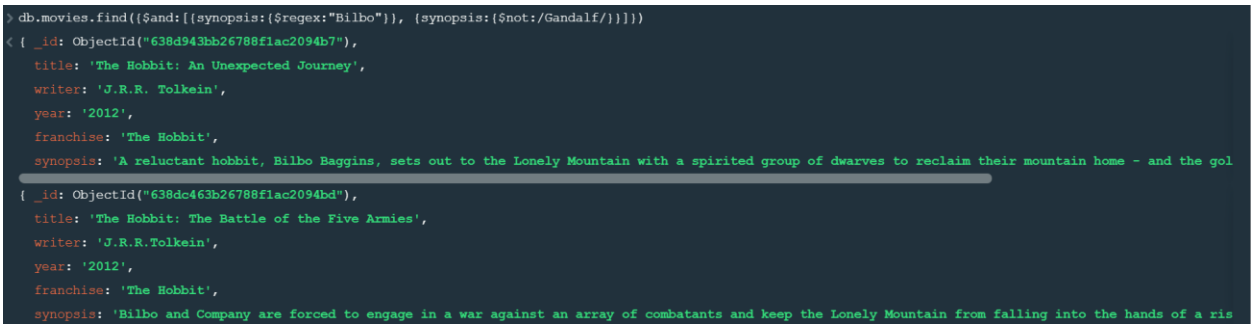
1. find all movies that have a synopsis that contains the word "Bilbo"



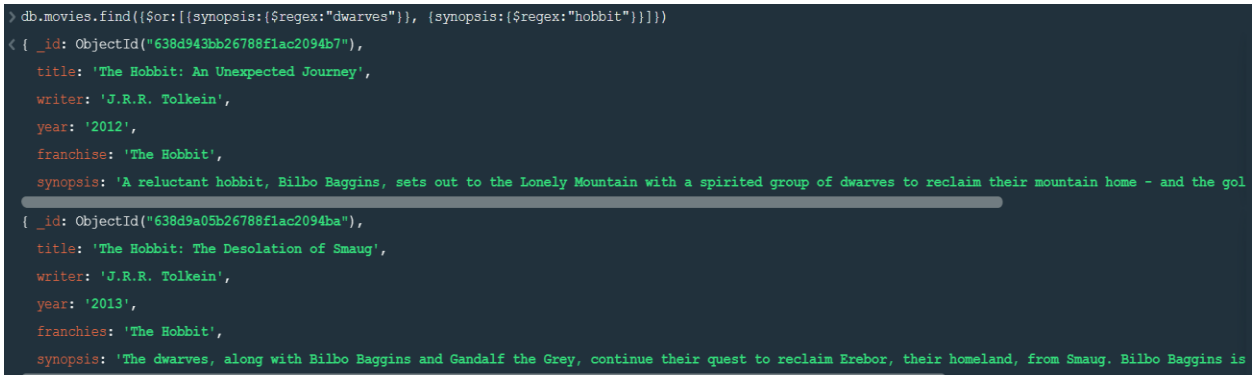
2. find all movies that have a synopsis that contains the word "Gandalf"



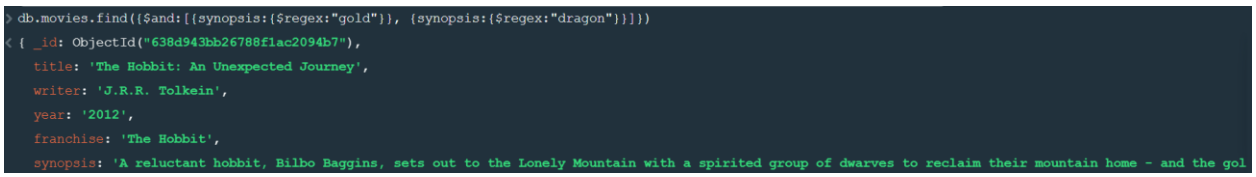
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"



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

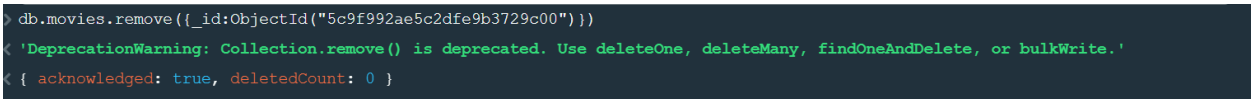


5. find all movies that have a synopsis that contains the word "gold" and "dragon"

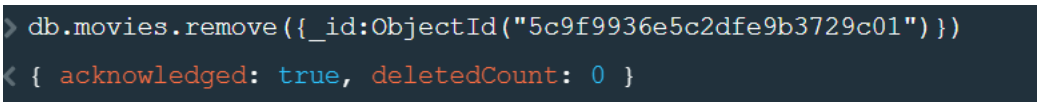


Delete Documents

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

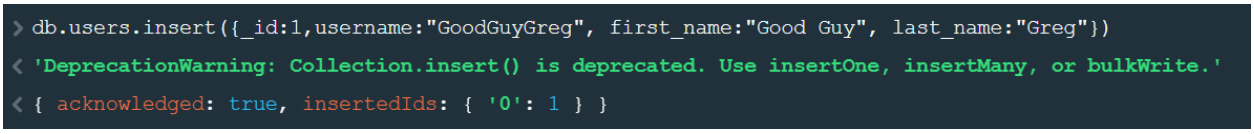


2. delete the movie "Avatar"



Relationships

username : GoodGuyGreg
first_name : "Good Guy"
last_name : "Greg"



username : ScumbagSteve
full_name :

first : "Scumbag"
last : "Steve"

```
> db.users.insert({_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})
< { acknowledged: true, insertedIds: { '0': 2 } }
```

Insert the following documents into a `posts` collection

username : GoodGuyGreg
title : Passes out at party
body : Wakes up early and cleans house

```
> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e09e64ca176ed620cc782") } }
```

username : GoodGuyGreg
title : Steals your identity
body : Raises your credit score

```
> db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0a534ca176ed620cc783") } }
```

username : GoodGuyGreg
title : Reports a bug in your code
body : Sends you a Pull Request

```
> db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0ab14ca176ed620cc784") } }
```

username : ScumbagSteve
title : Borrows something
body : Sells it

```
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0b014ca176ed620cc785") } }
```

username : ScumbagSteve
title : Borrows everything
body : The end

```
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0b594ca176ed620cc786") } }
```

username : ScumbagSteve
title : Forks your repo on github
body : Sets to private

```
> db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0bf54ca176ed620cc787") } }
```

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"

```
> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("5ca0b7e96435f98b5901f463")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0cc84ca176ed620cc788") } }
```

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"

```
> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("5ca0b9706435f98b5901f46a")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0d574ca176ed620cc789") } }
```

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

```
> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("5ca0b8766435f98b5901f467")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0dc34ca176ed620cc78a") } }
```

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"

```
> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("5ca0b8546435f98b5901f466")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0e394ca176ed620cc78b") } }
```

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"

```
> db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("5ca0b9256435f98b5901f469")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("638e0ead4ca176ed620cc78c") } }
```

Querying related collections

1. find all users

```
> db.users.find().pretty()
< { _id: 1,
    username: 'GoodGuyGreg',
    first_name: 'Good Guy',
    last_name: 'Greg' }
  { _id: 2,
    username: 'ScumbagSteve',
    fullname: { first: 'Scumbag', last: 'Steve' } }
```

2. find all posts

```
> db.posts.find().pretty()
< { _id: ObjectId("638e09e64ca176ed620cc782"),
    username: 'GoodGuyGreg',
    title: 'Passes out at Party',
    body: 'Raises your credit score' }
  { _id: ObjectId("638e0a534ca176ed620cc783"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score' }
  { _id: ObjectId("638e0ab14ca176ed620cc784"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request' }
  { _id: ObjectId("638e0b014ca176ed620cc785"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it' }
  { _id: ObjectId("638e0b594ca176ed620cc786"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
```

```
  body: 'The end' }
{ _id: ObjectId("638e0bf54ca176ed620cc787"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private' }
```

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

```
> db.posts.find({username:"GoodGuyGreg"})
< { _id: ObjectId("638e09e64ca176ed620cc782"),
    username: 'GoodGuyGreg',
    title: 'Passes out at Party',
    body: 'Raises your credit score' }
  { _id: ObjectId("638e0a534ca176ed620cc783"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score' }
  { _id: ObjectId("638e0ab14ca176ed620cc784"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request' }
```

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

```
> db.posts.find({username:"ScumbagSteve"})
< { _id: ObjectId("638e0b014ca176ed620cc785"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it' }
  { _id: ObjectId("638e0b594ca176ed620cc786"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end' }
  { _id: ObjectId("638e0bf54ca176ed620cc787"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private' }
```

5. find all comments

```
> db.comments.find().pretty()
< { _id: ObjectId("638e0cc84ca176ed620cc788"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("5ca0b7e96435f98b5901f463") }
  { _id: ObjectId("638e0d574ca176ed620cc789"),
    username: 'GoodGuyGreg',
    comment: 'What\'s mine is yours!',
    post: ObjectId("5ca0b9706435f98b5901f46a") }
  { _id: ObjectId("638e0dc34ca176ed620cc78a"),
    username: 'GoodGuyGreg',
    comment: 'Don\'t violate the licensing agreement!',
    post: ObjectId("5ca0b8766435f98b5901f467") }
  { _id: ObjectId("638e0e394ca176ed620cc78b"),
    username: 'ScumbagSteve',
    comment: 'It still isn\'t clean',
    post: ObjectId("5ca0b8546435f98b5901f466") }
  { _id: ObjectId("638e0ead4ca176ed620cc78c"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
```

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

```
> db.comments.find({username:"GoodGuyGreg"})
< { _id: ObjectId("638e0cc84ca176ed620cc788"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("5ca0b7e96435f98b5901f463") }
  { _id: ObjectId("638e0d574ca176ed620cc789"),
    username: 'GoodGuyGreg',
    comment: 'What\'s mine is yours!',
    post: ObjectId("5ca0b9706435f98b5901f46a") }
  { _id: ObjectId("638e0dc34ca176ed620cc78a"),
    username: 'GoodGuyGreg',
    comment: 'Don\'t violate the licensing agreement!',
    post: ObjectId("5ca0b8766435f98b5901f467") }
```

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

```
> db.comments.find({username:"ScumbagSteve"})
< { _id: ObjectId("638e0e394ca176ed620cc78b"),
    username: 'ScumbagSteve',
    comment: 'It still isn\'t clean',
    post: ObjectId("5ca0b8546435f98b5901f466") }
  { _id: ObjectId("638e0ead4ca176ed620cc78c"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("5ca0b9256435f98b5901f469") }
```