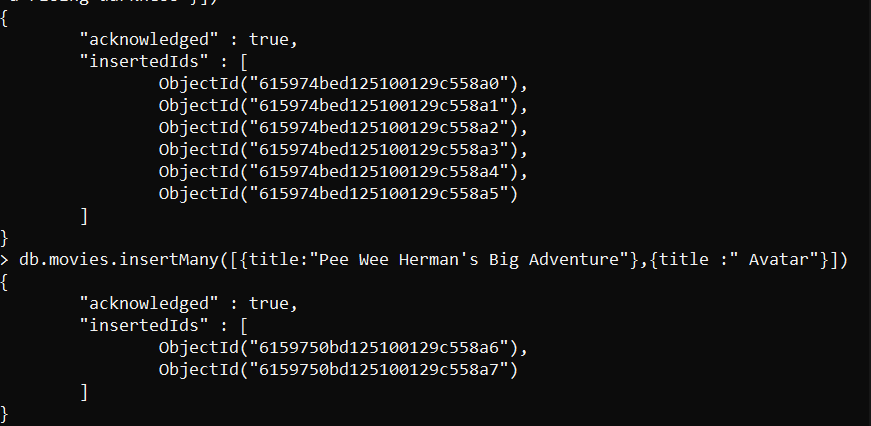
**MongoDB Assignment – 1**

**Insert Documents**

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

* db.movies.insertMany([{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"}])

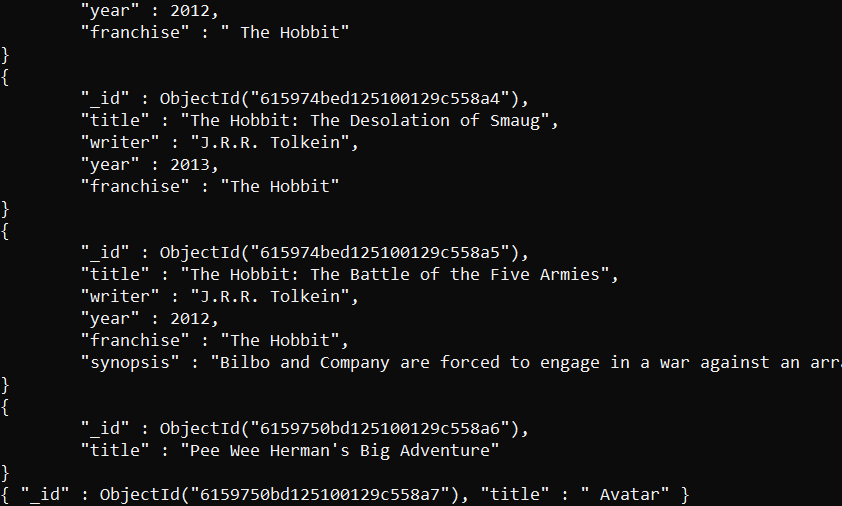


**Query / Find Documents**

1. Get all documents

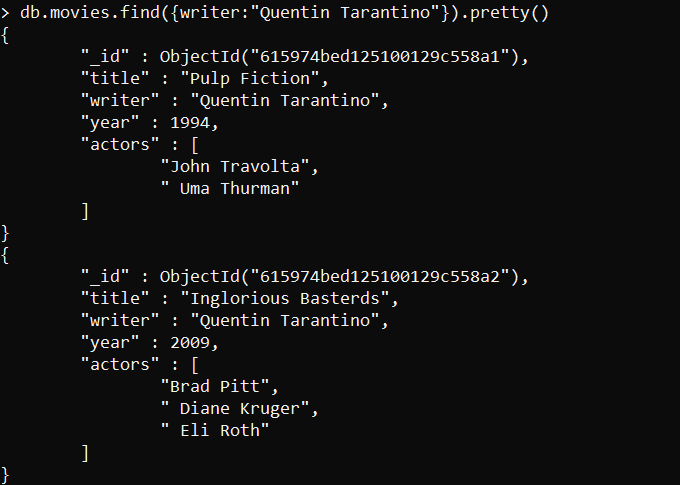
* db.movies.find().pretty()





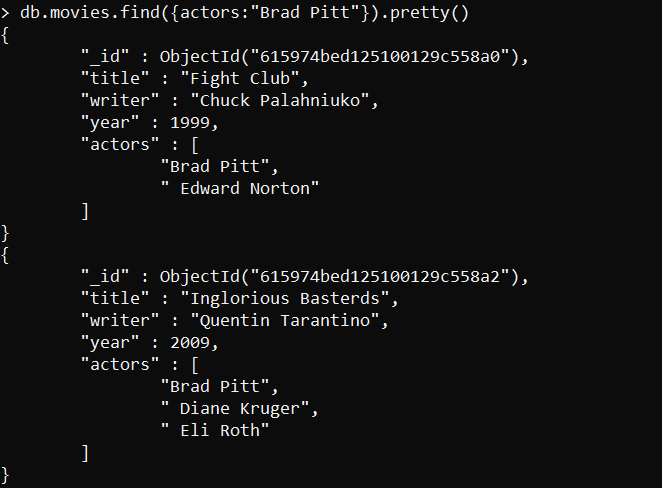
1. Get all documents with writer set to "Quentin Tarantino"

* db.movies.find({writer:"Quentin Tarantino"}).pretty()



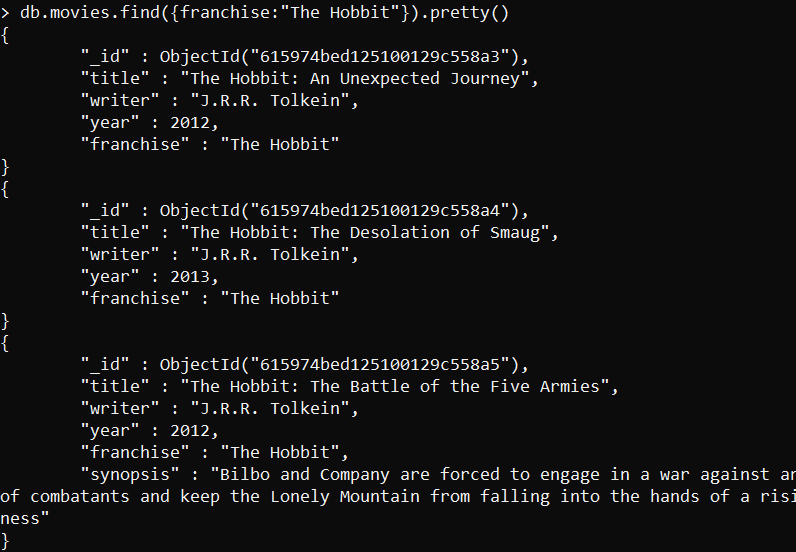
1. Get all documents where actors include "Brad Pitt"

* db.movies.find({actors:"Brad Pitt"}).pretty()



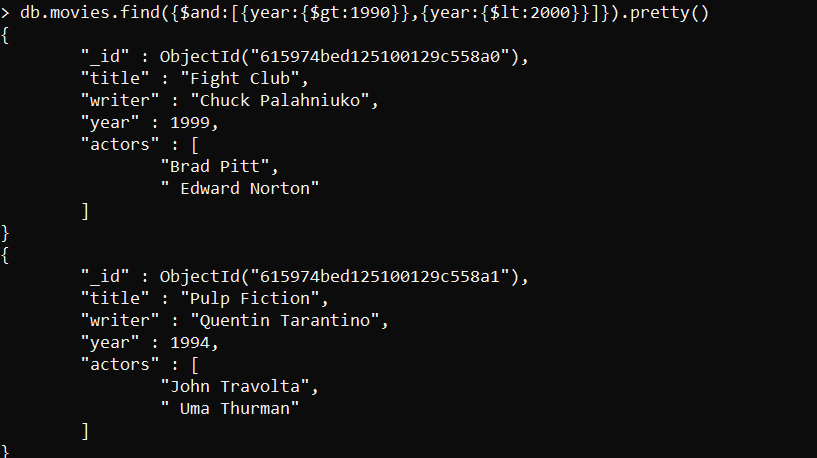
1. Get all documents with franchise set to "The Hobbit"

* db.movies.find({franchise:"The Hobbit"}).pretty()



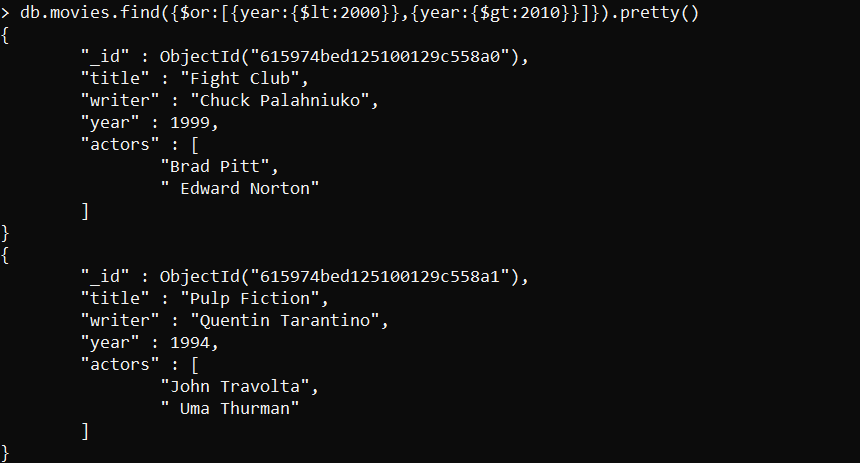
1. Get all movies released in the 90s

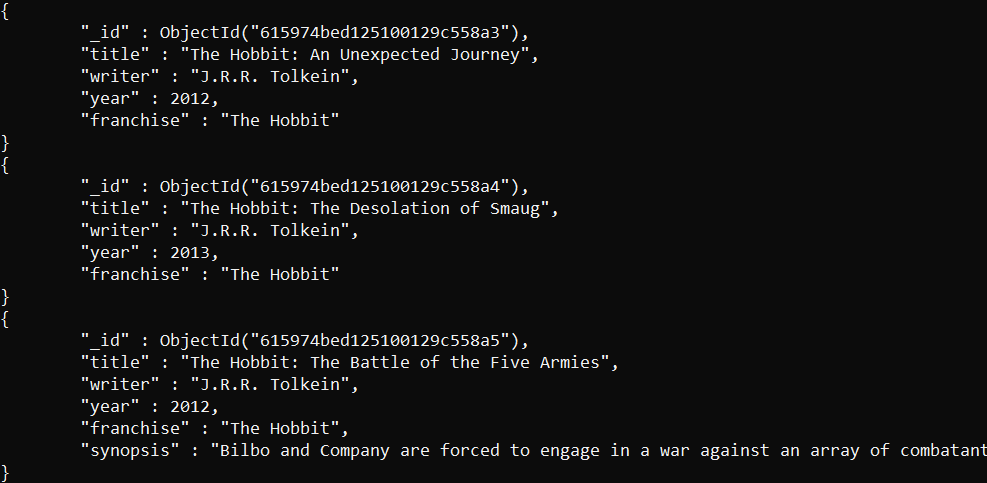
* db.movies.find({$and:[{year:{$gt:1990}},{year:{$lt:2000}}]}).pretty()



1. Get all movies released before the year 2000 or after 2010

* db.movies.find({$or:[{year:{$lt:2000}},{year:{$gt:2010}}]}).pretty()

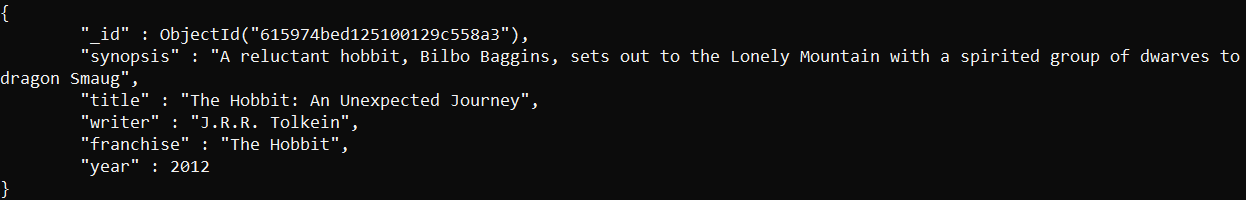




**Update Documents**

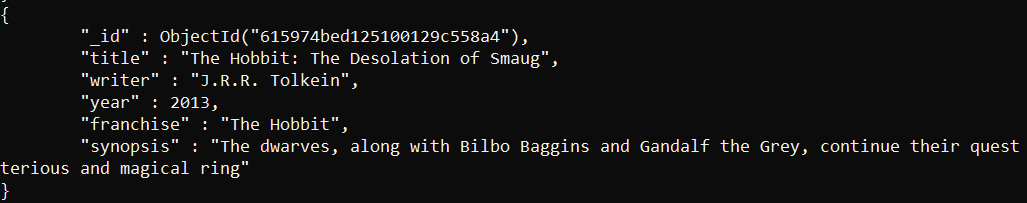
1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

* db.movies.update({title:"The Hobbit: An Unexpected Journey"},{$set:{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"}})



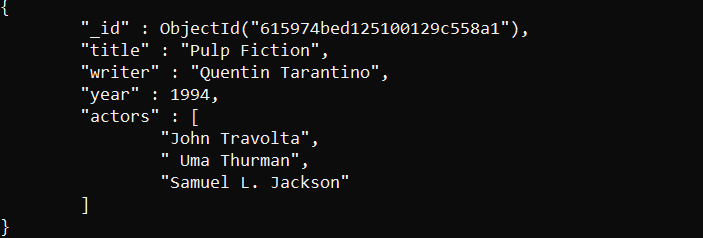
1. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

* db.movies.update({title:"The Hobbit: The Desolation of Smaug"},{$set:{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"}})



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

* db.movies.update({title:"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})

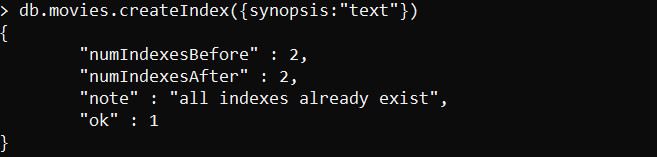


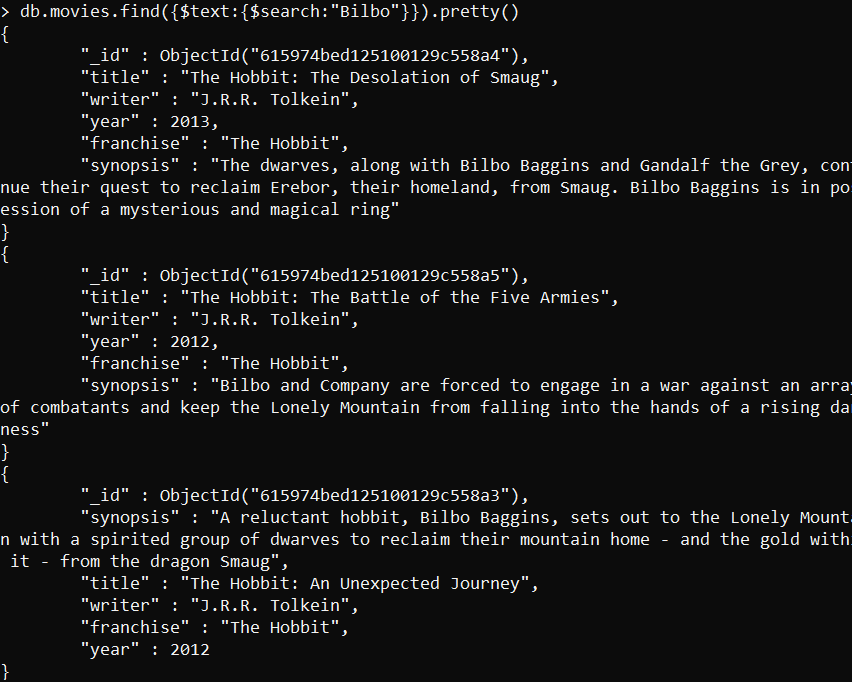
**Text Search**

* **db.movies.createIndex({synopsis:"text"}).pretty()**

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

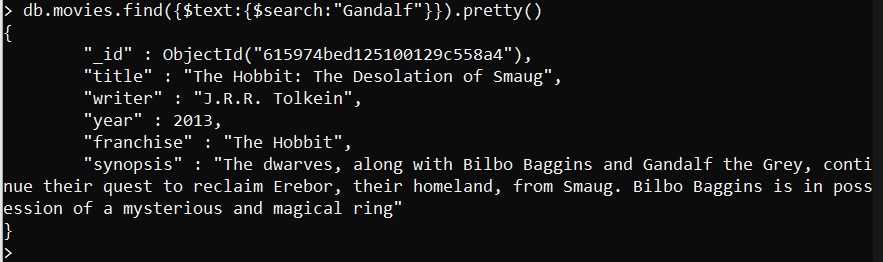
* db.movies.find({$text:{$search:"Bilbo"}}).pretty()





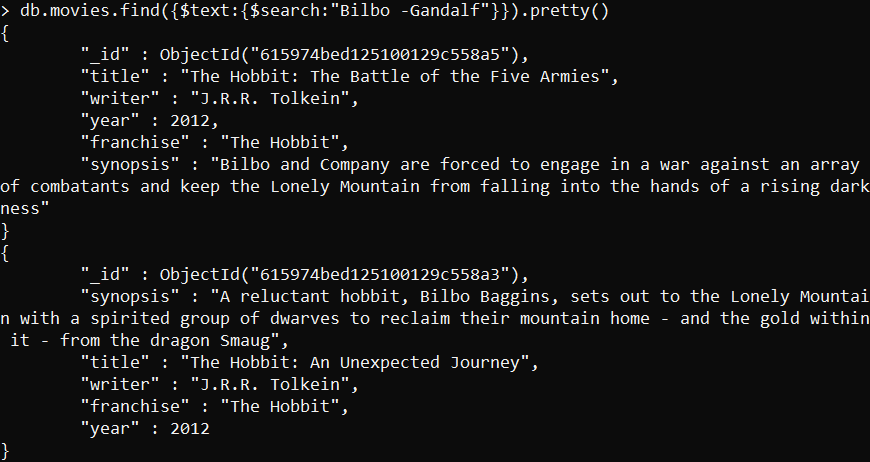
1. find all movies that have a synopsis that contains the word "Gandalf"

* db.movies.find({$text:{$search:"Gandalf"}}).pretty()



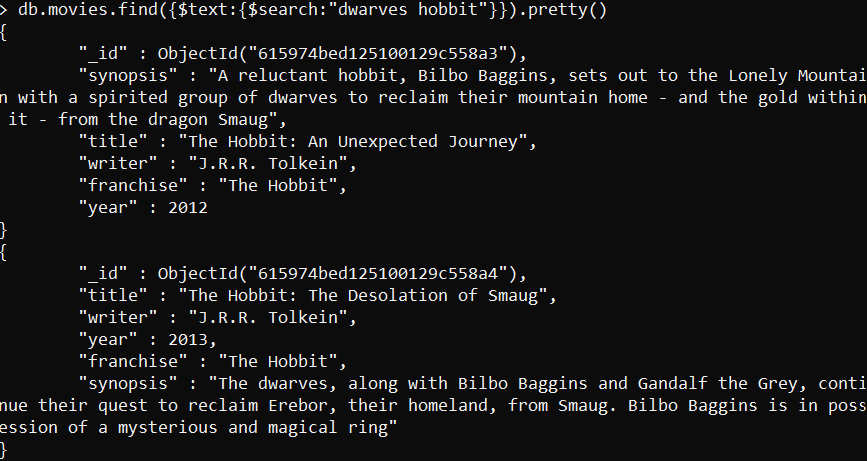
1. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

* db.movies.find({$text:{$search:"Bilbo -Gandalf"}}).pretty()



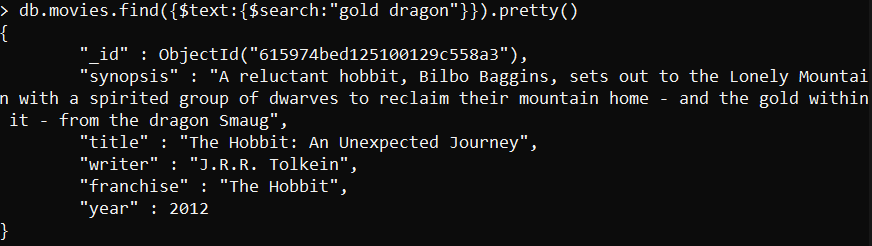
1. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

* db.movies.find({$text:{$search:"dwarves hobbit"}}).pretty()



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

* db.movies.find({$text:{$search:"gold dragon"}}).pretty()



**Delete Documents**

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

* db.movies.remove({title:"Pee Wee Herman's Big Adventure"})

1. delete the movie "Avatar"

* db.movies.remove({title:" Avatar"})

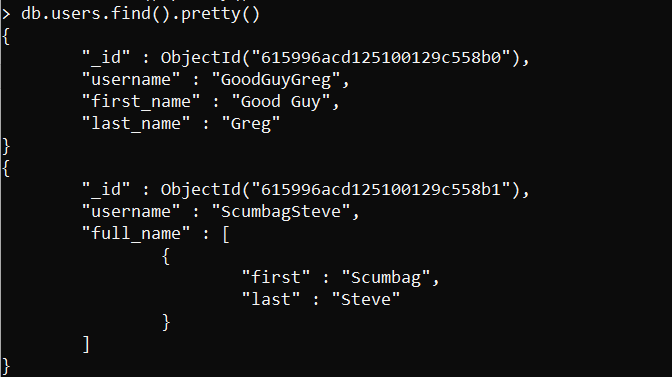
**Relationships**

* **users** collection
* db.users.insertMany([{username : "GoodGuyGreg",first\_name : "Good Guy",last\_name : "Greg"},{username : "ScumbagSteve",full\_name : [{"first" : "Scumbag","last" : "Steve"}]}])
* **posts** collection
* db.posts.insertMany([{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"}])
* **comments** collection
* db.comments.insertMany([{username : "GoodGuyGreg",comment : "Hope you got a good deal!",post : "615996ddd125100129c558b5"},{username : "GoodGuyGreg",comment : "What's mine is yours!",post : "615996ddd125100129c558b6"},{username : "GoodGuyGreg",comment : "Don't violate the licensing agreement!",post : "615996ddd125100129c558b7"},{username : "ScumbagSteve",comment : "It still isn't clean",post : "615996ddd125100129c558b2"},{username : "ScumbagSteve",comment : "Denied your PR cause I found a hack",post : "615996ddd125100129c558b4"}])

**Querying related collections**

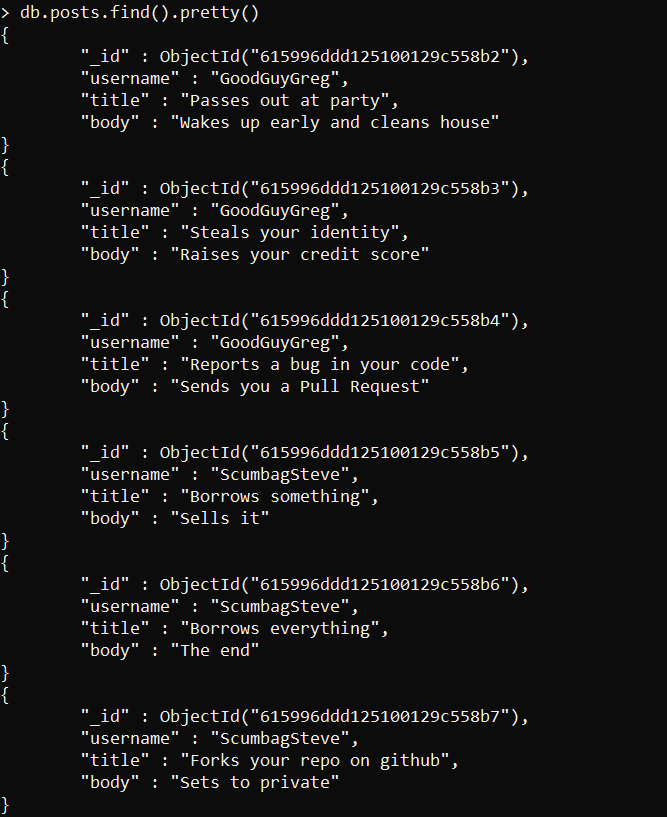
1. find all users

* db.users.find().pretty()



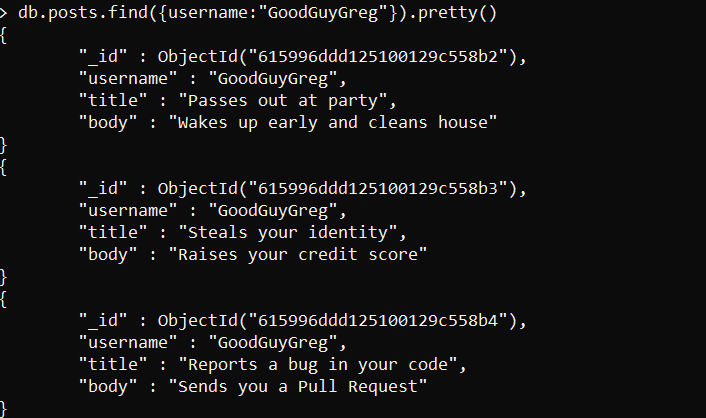
1. find all posts

* db.posts.find().pretty()



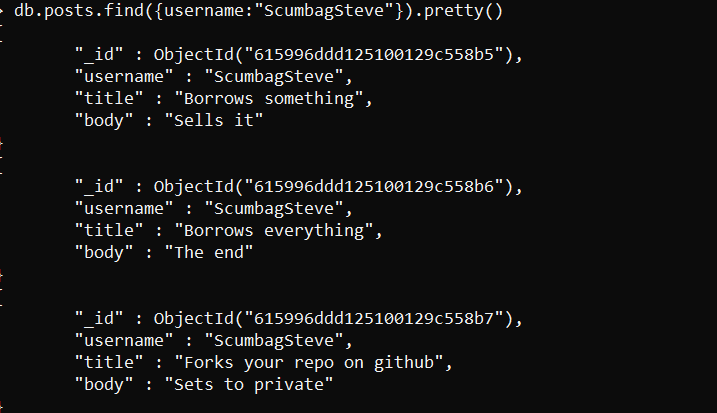
1. find all posts that was authored by "GoodGuyGreg"

* db.posts.find({username:"GoodGuyGreg"}).pretty()



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

* db.posts.find({username:"ScumbagSteve"}).pretty()



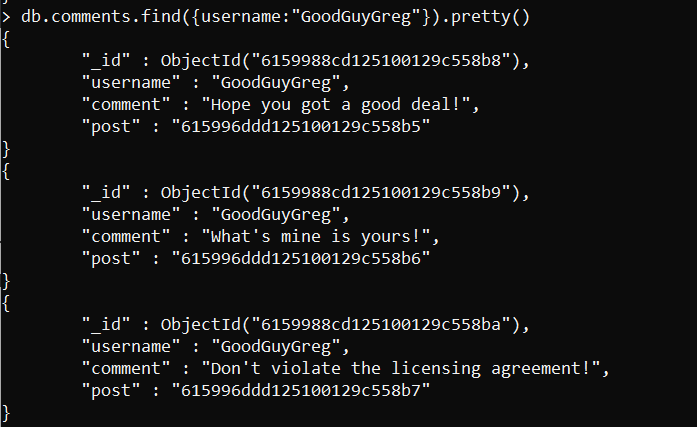
1. find all comments

* db.comments.find().pretty()



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

* db.comments.find({username:"GoodGuyGreg"}).pretty()



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

* db.comments.find({username:"ScumbagSteve"}).pretty()

