MongoDB Lab Assignments -Day 1

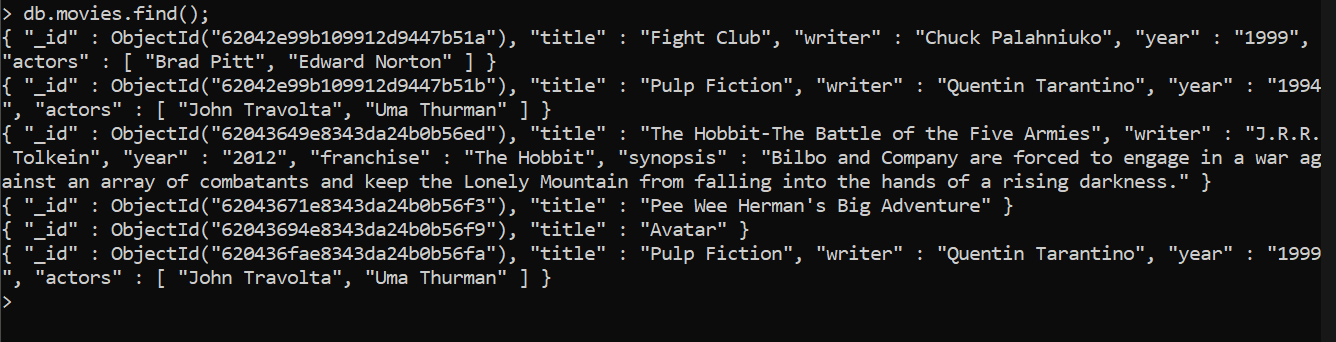
Use a database named **mongo\_practice**



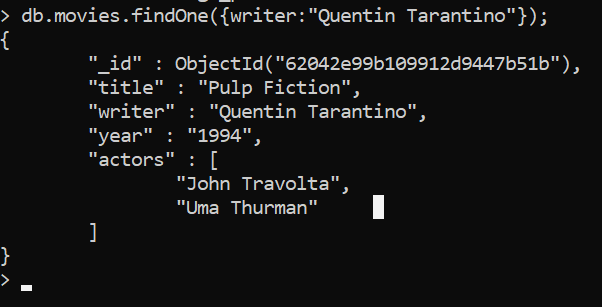


Query / Find Documents query the movies collection to

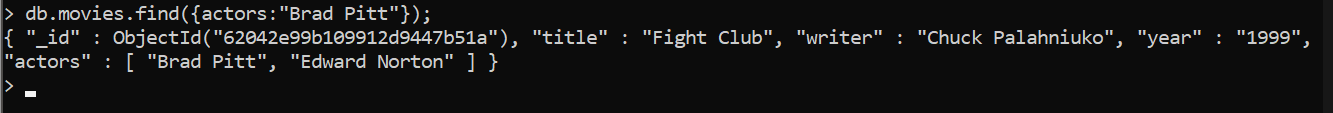
1. Get all documents



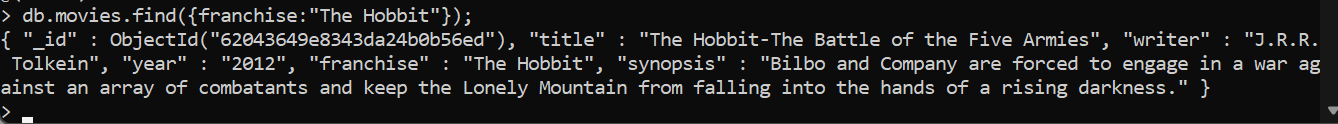
2. Get all documents with writer set to "Quentin Tarantino"



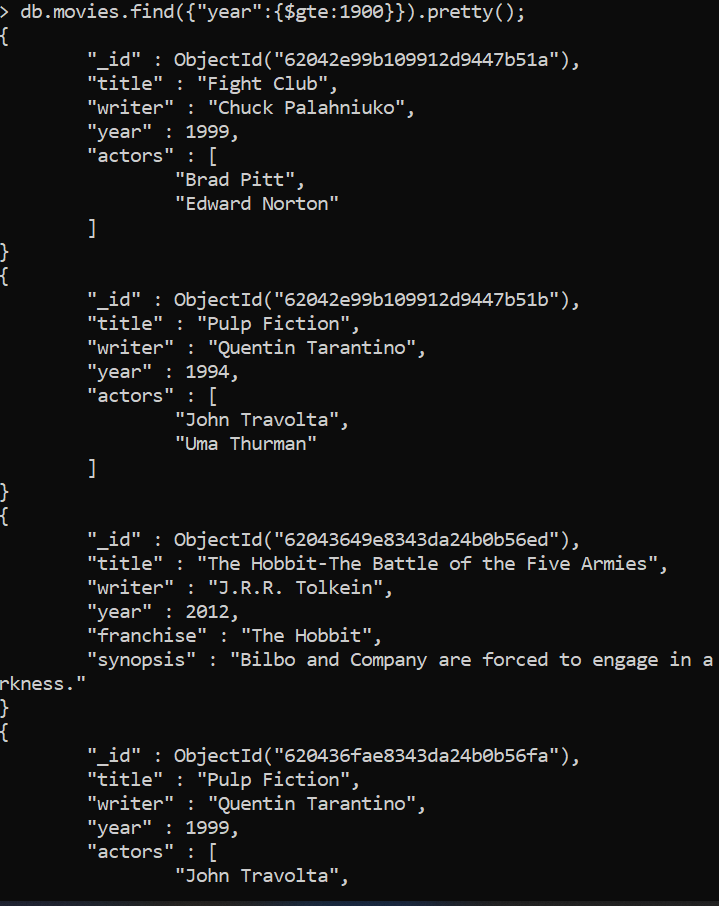
3. Get all documents where actors include "Brad Pitt"



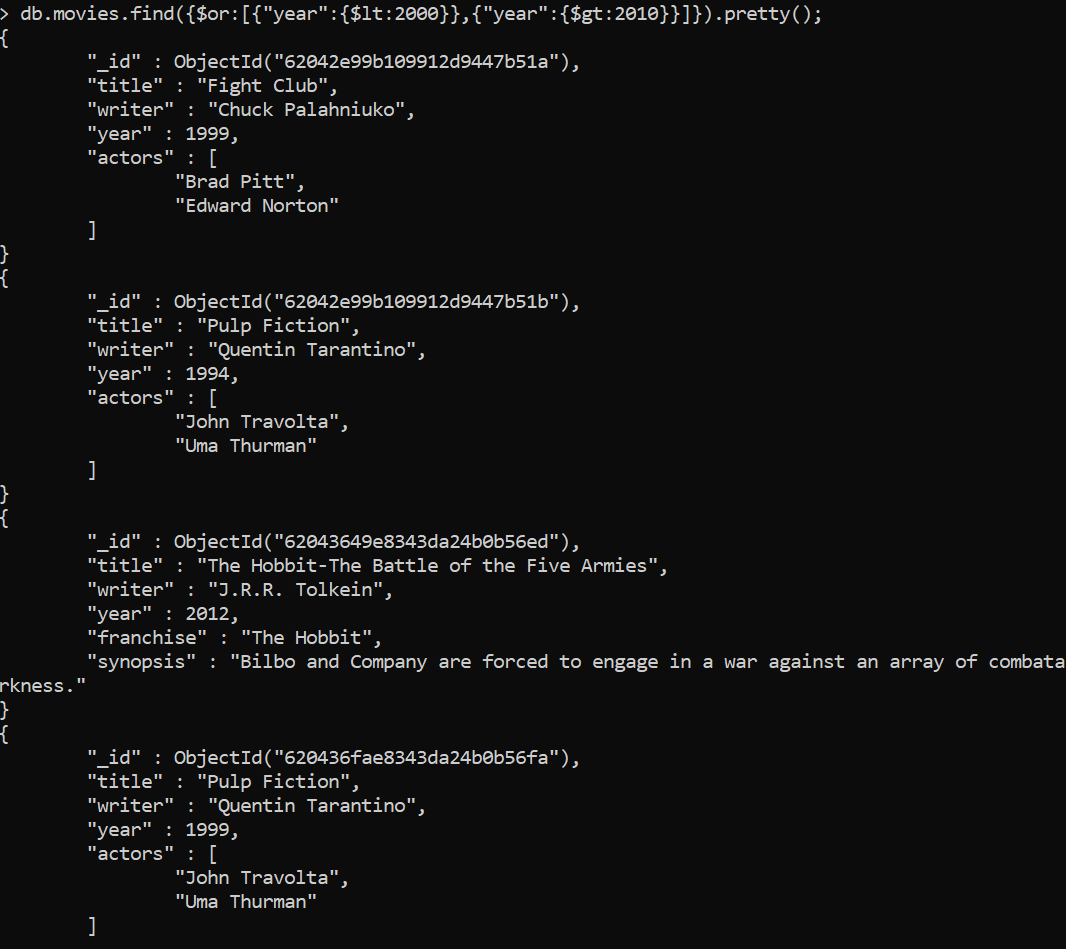
4. Get all documents with franchise set to "The Hobbit"



5.Get all movies released in the 90s

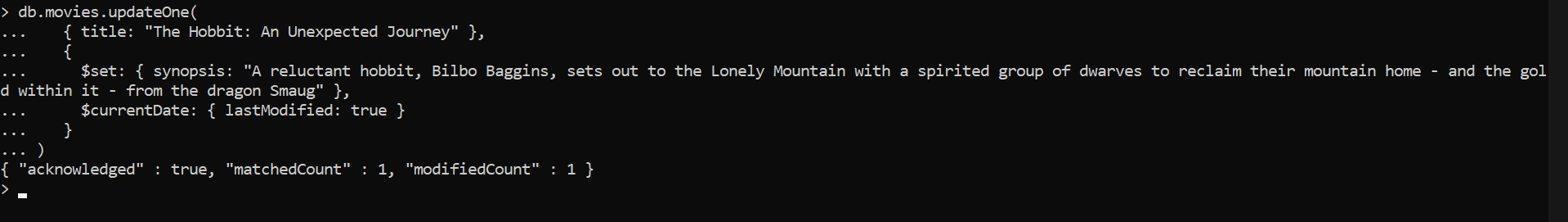


6.Get all movies released before the year 2000 or after 2010

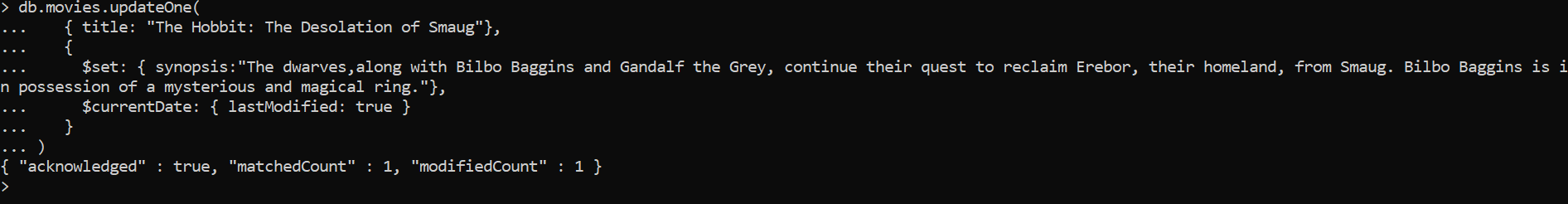


Update Documents

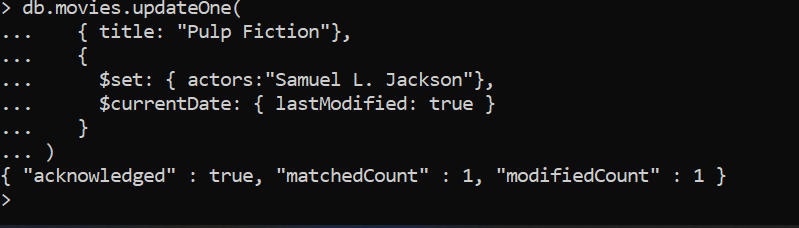
1.Add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."



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."

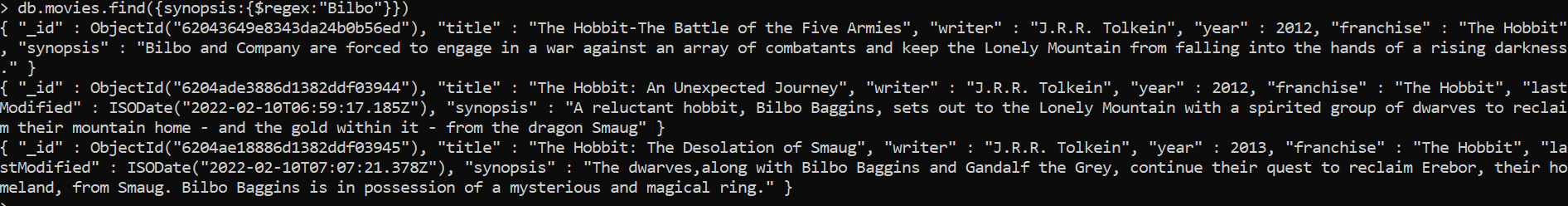


3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

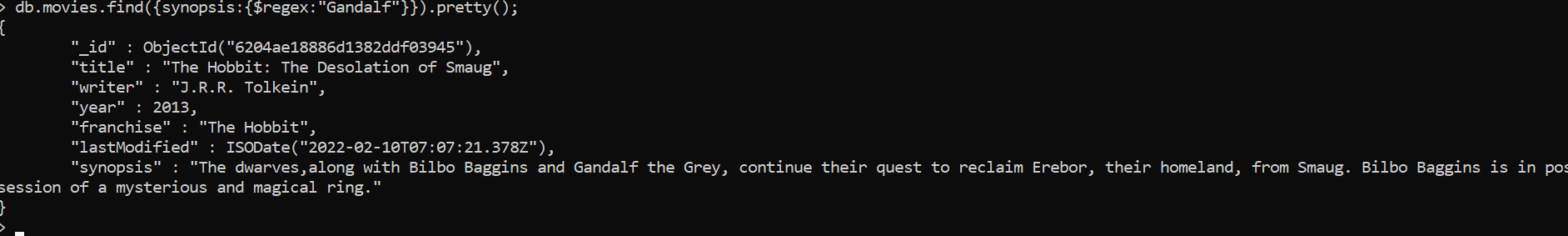


**Text Search**

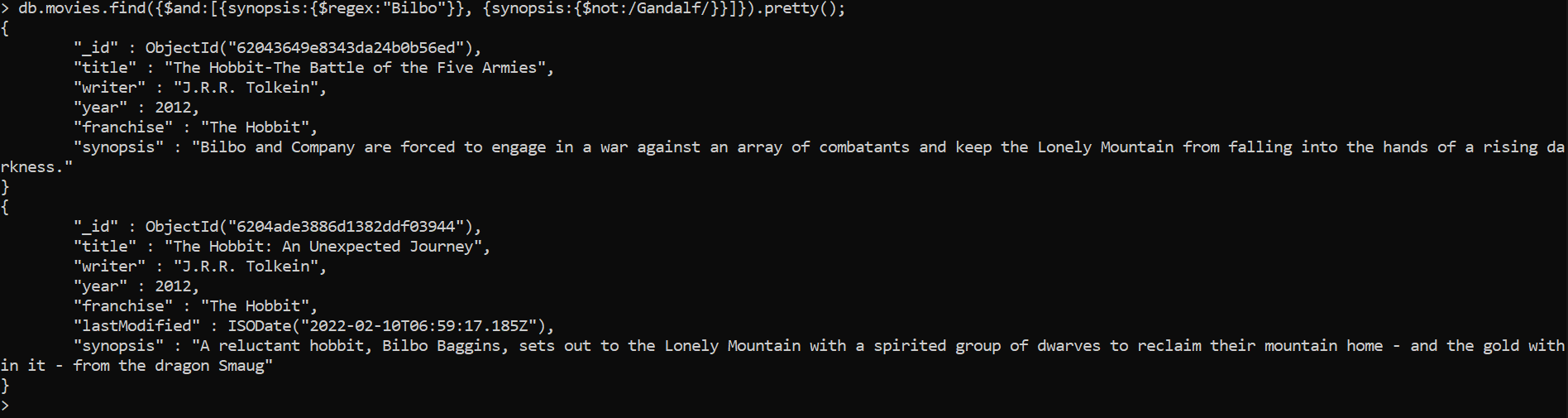
1.find all movies that have a synopsis that contains the word "Bilbo"



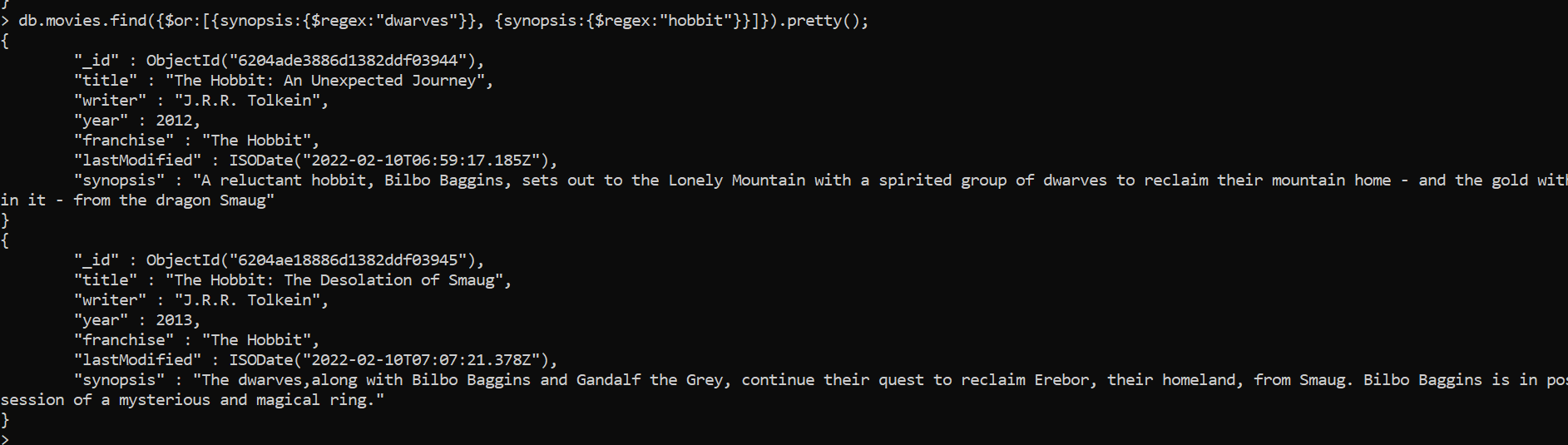
2. find all movies that have a synopsis that contains the word "Gandalf"



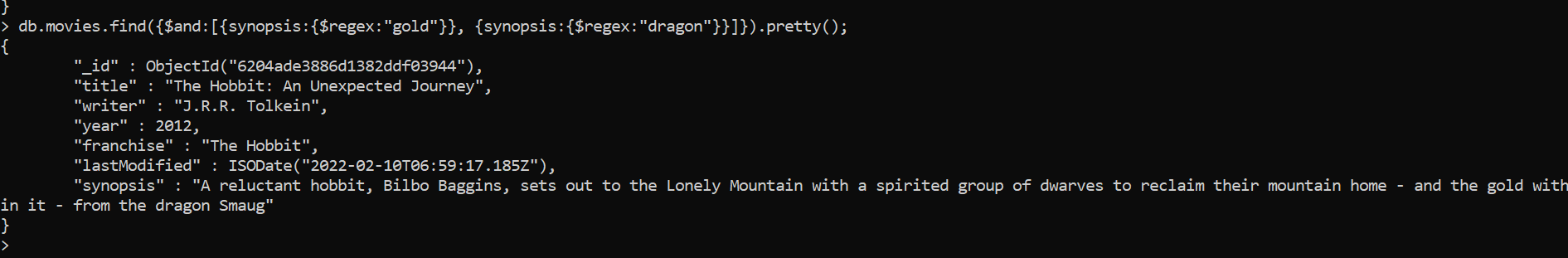
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"



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

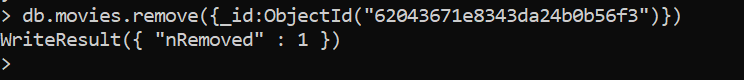


5. find all movies that have a synopsis that contains the word "gold" and "dragon"



**Delete Documents**

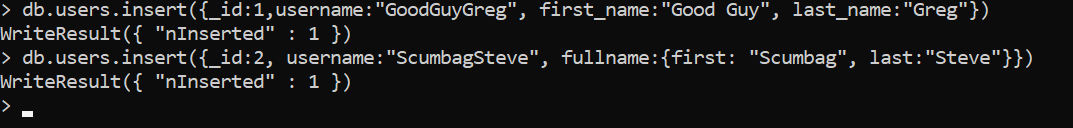
1.delete the movie "Pee Wee Herman's Big Adventure"



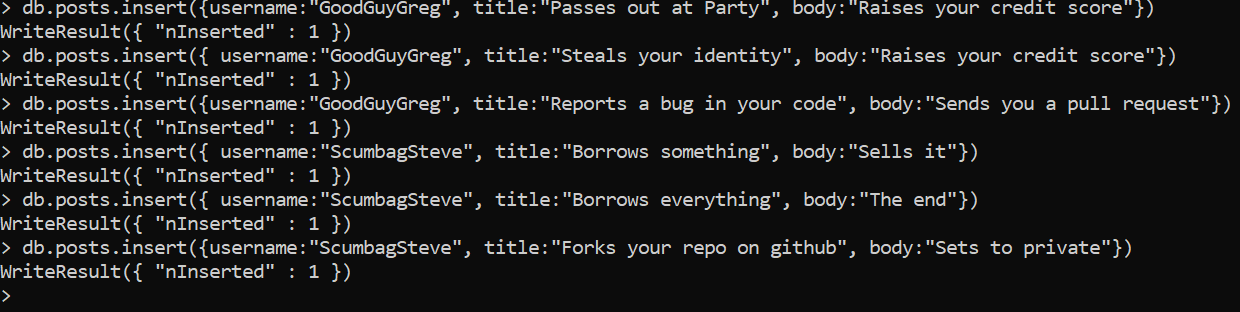
2. delete the movie "Avatar"



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



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

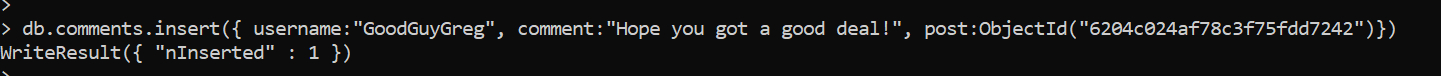


**Insert the following documents into a comments collection**

username : GoodGuyGreg

comment : Hope you got a good deal!

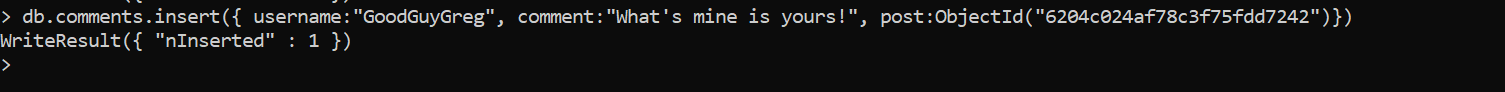
post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows something"



username : GoodGuyGreg

comment : What's mine is yours!

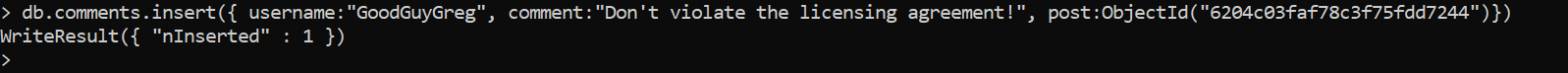
post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows everything"



username : GoodGuyGreg

comment : Don't violate the licensing agreement!

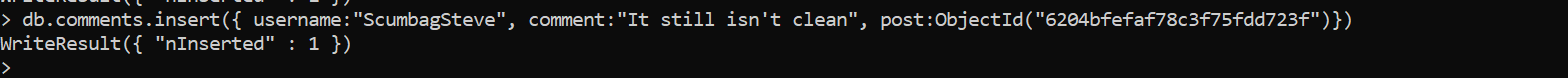
post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Forks your repo on github



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"

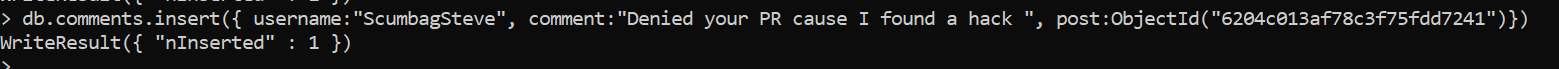


username : ScumbagSteve

comment : Denied your PR cause I found a hack

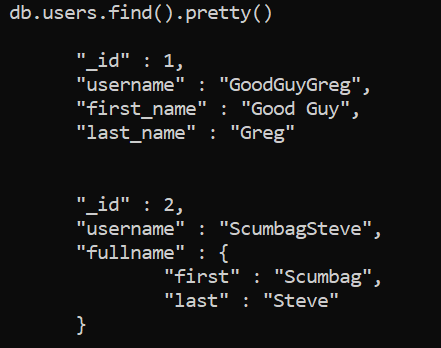
post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts

document: "Reports a bug in your code"

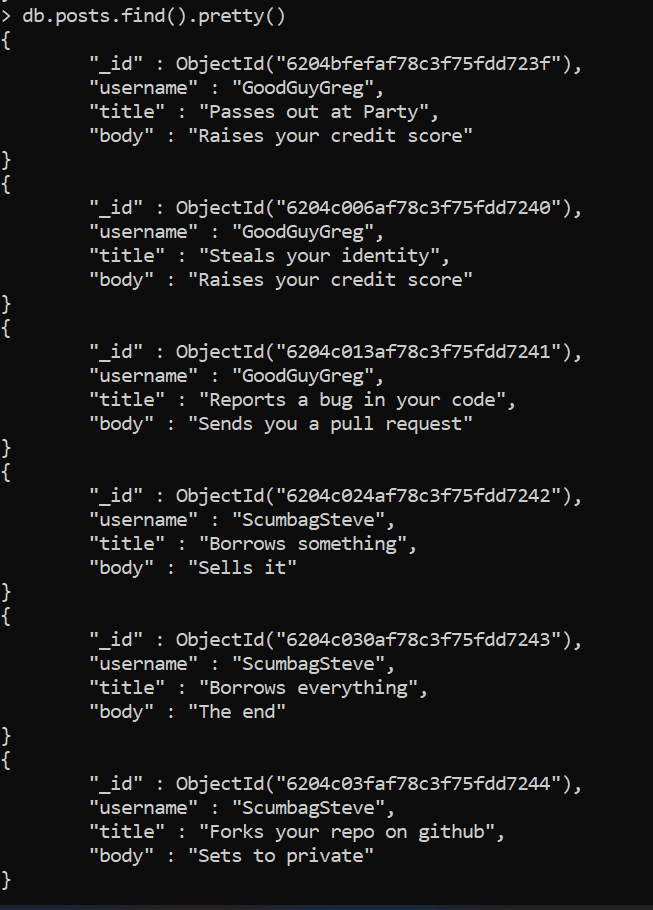


Querying related collections

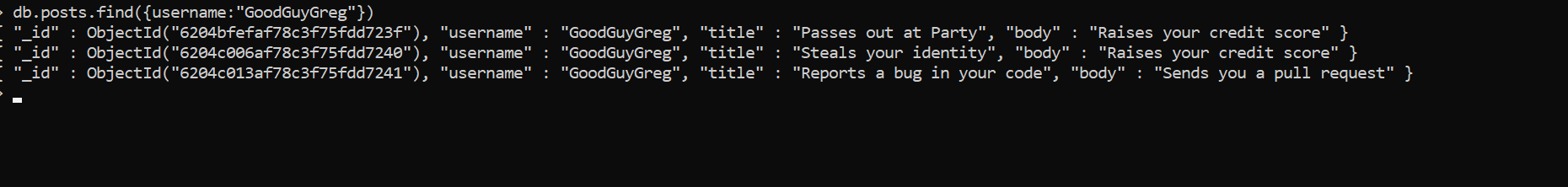
1. find all users



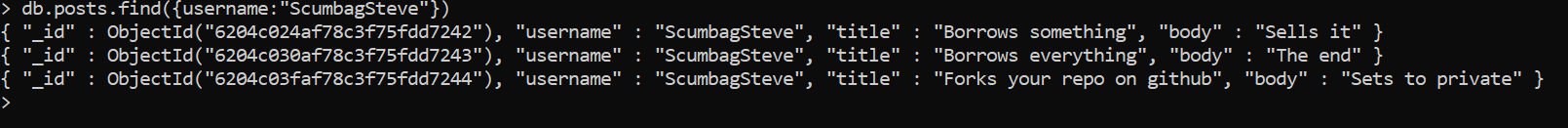
1. find all posts



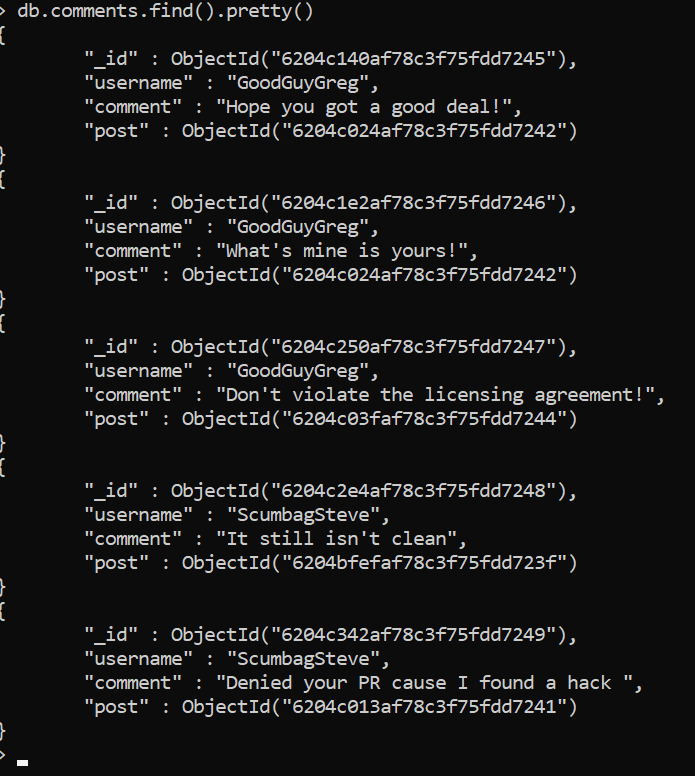
1. find all posts that was authored by "GoodGuyGreg"



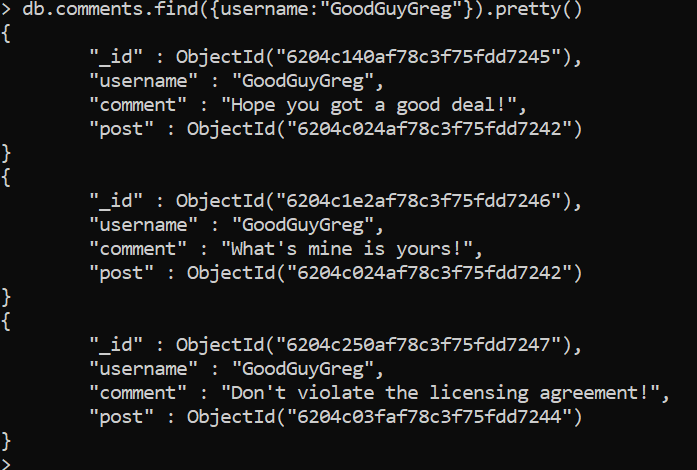
1. find all posts that was authored by "ScumbagSteve"



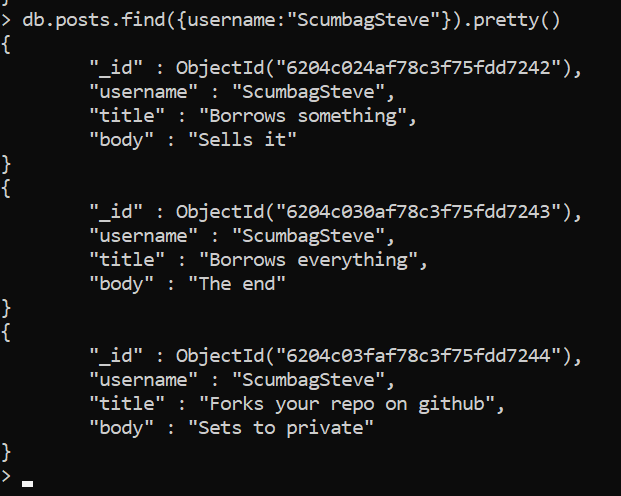
1. find all comments



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



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



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

