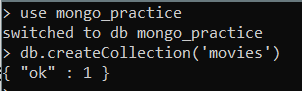
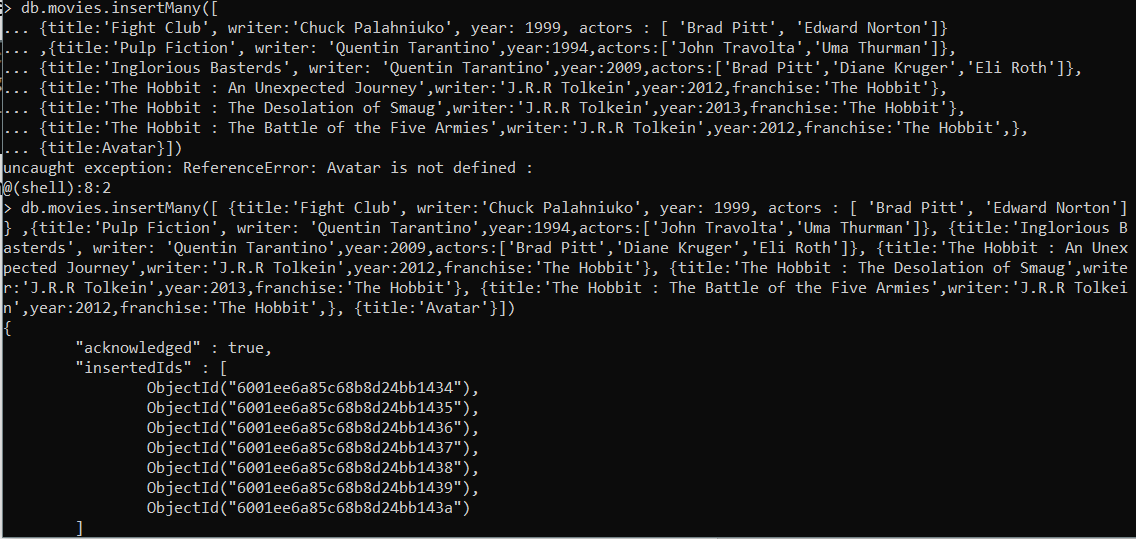
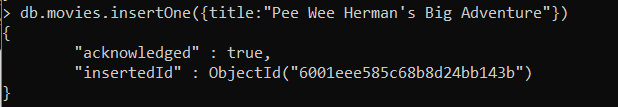
Creating database and  **Movies** collection

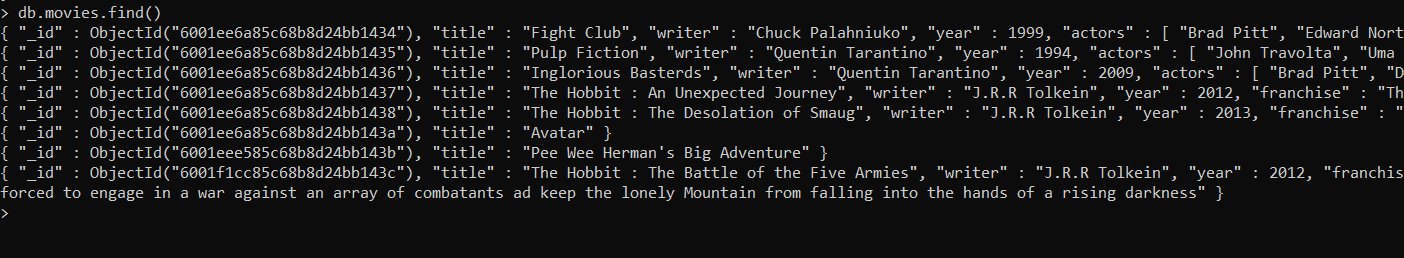


Inserting the documents:





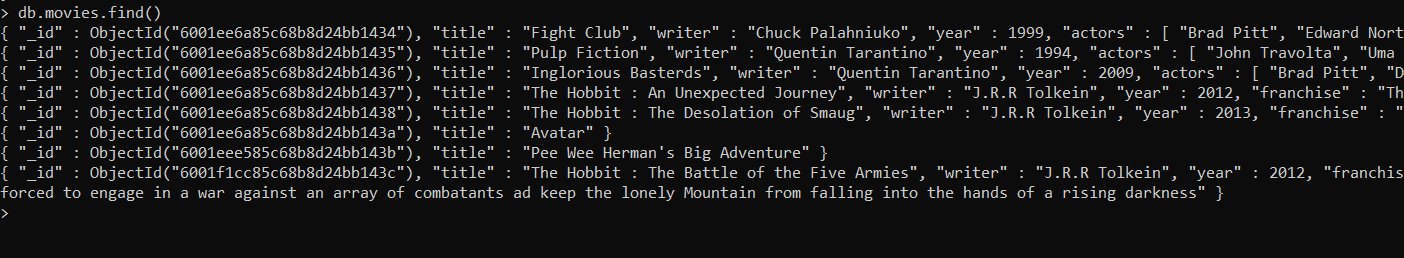
Collection movies after insertion :



Query on the documents:

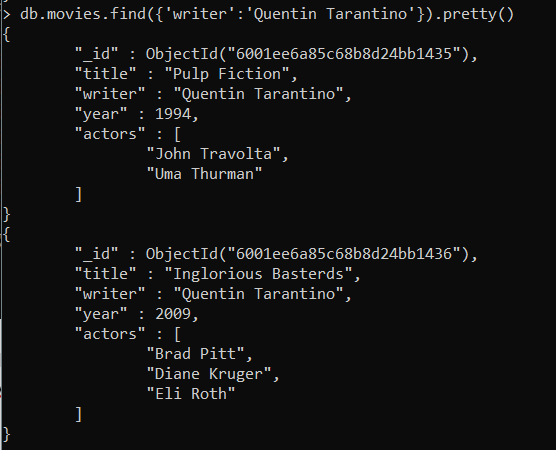
1. To get all the documents:

: db.movies.find() // db.movies.find().pretty() – for better representation



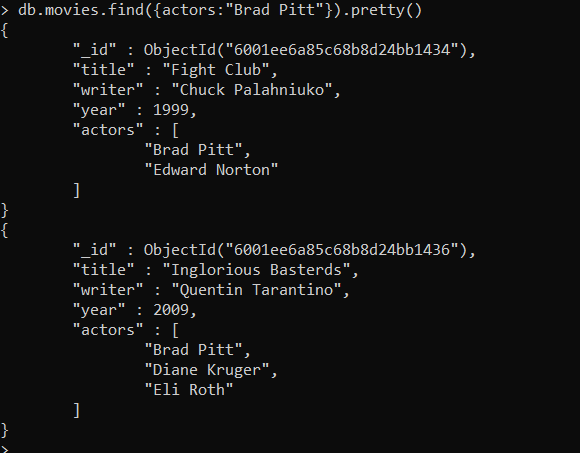
1. To get all the documents with writer set to "Quentin Tarantino"

: db.movies.find({‘writer’ : ’Quentin Tarantino’})



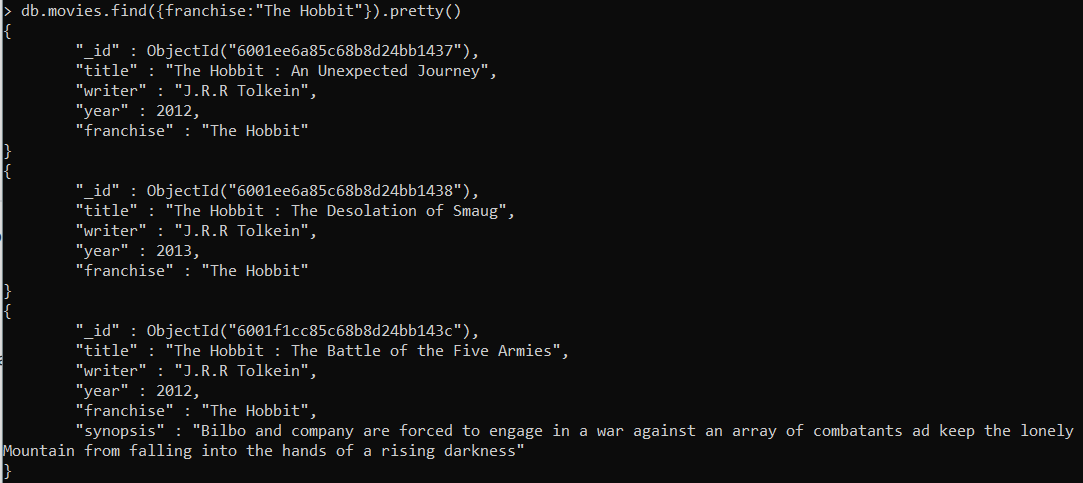
1. get all documents where actors include "Brad Pitt"

: db.movies.find({actors : ”Brad Pitt”})



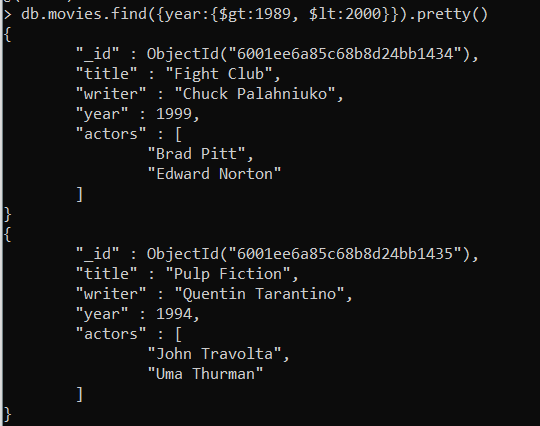
1. get all documents with franchise set to "The Hobbit"

: db.movies.find({franchise : ’The Hobbit’})



1. get all movies released in the 90s

: db.movies.find( { year:{$gt : 1989, $lt : 2000} } )



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

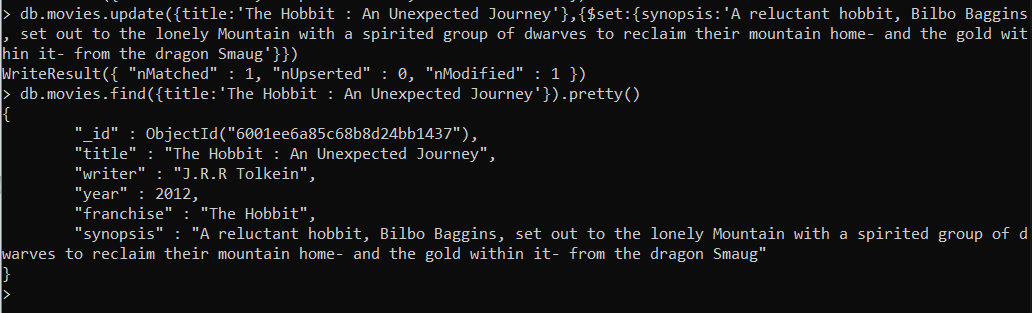
: db.movies.find( {$or : [*{ year:{$gt : 2010} }* ,*{year:{$lt : 2000} }*] } )



Update Documents

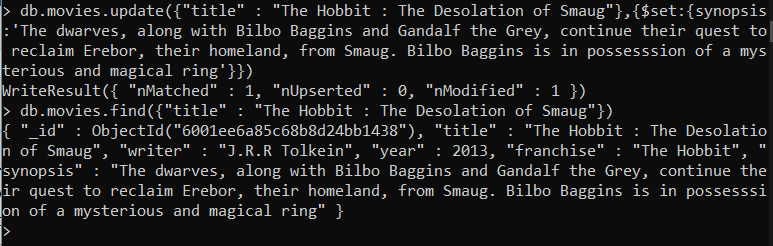
1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

: 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."}}



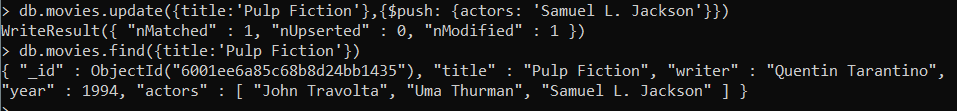
1. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

: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"

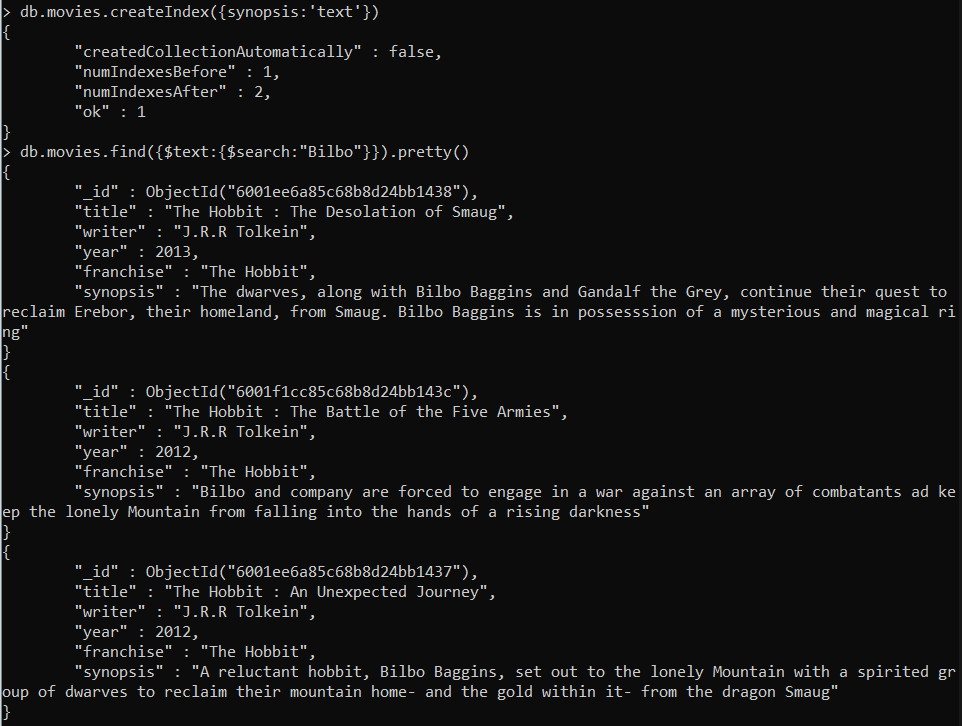
:db.movies.find({title : ’Pulp Fiction’},{$push: {actors : ‘Samuel L. Jackson’}})



Text Search

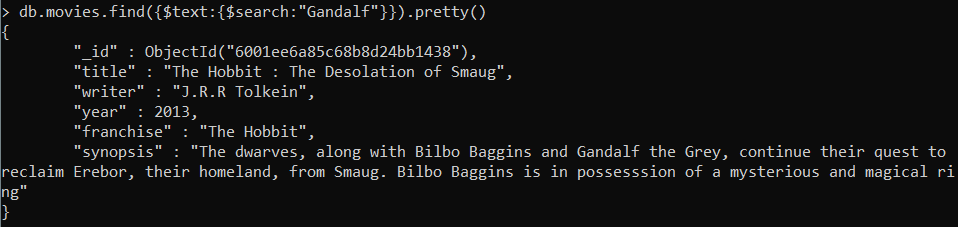
1. . find all movies that have a synopsis that contains the word "Bilbo"

: db.movies.find({$text:{$search:”Bilbo”}}) \\ after creating text index of synopsis



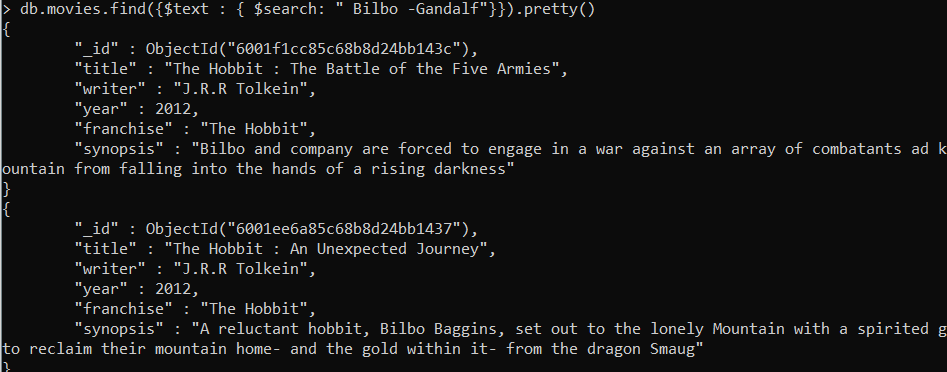
1. find all movies that have a synopsis that contains the word "Gandalf"

: db.movies.find({$text:{$search:”Gandalf”}})

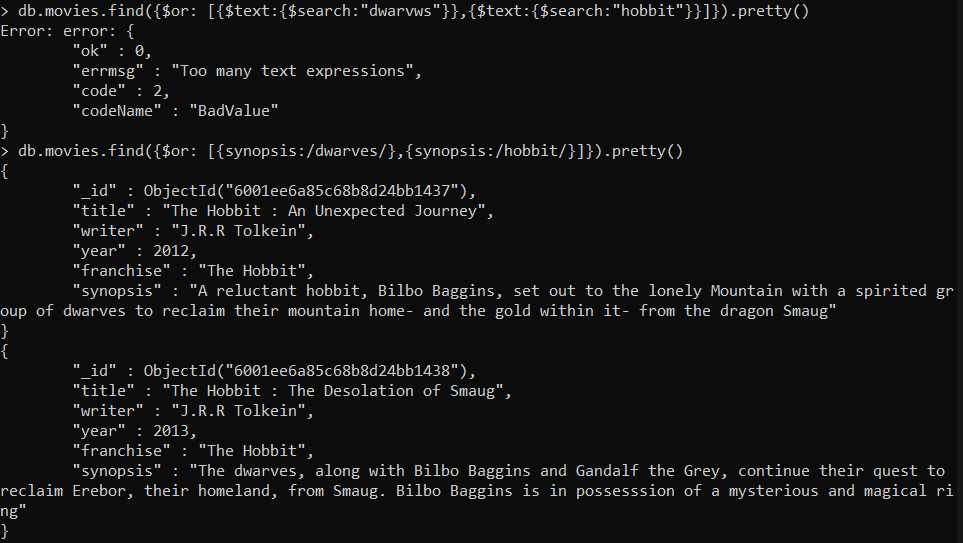


1. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

:db.movies.find({$text : { $search : “Bilbo -Ganalf”}})

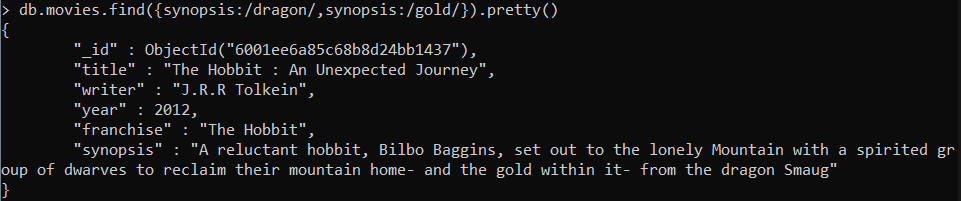


1. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

: db.movies.find( {$or : [{synopsis:/dwarves/},{synopsis:/hobbit/}]} )

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

: db.movies.find({{synopsis:/gold/,synopsis:/dragon/})



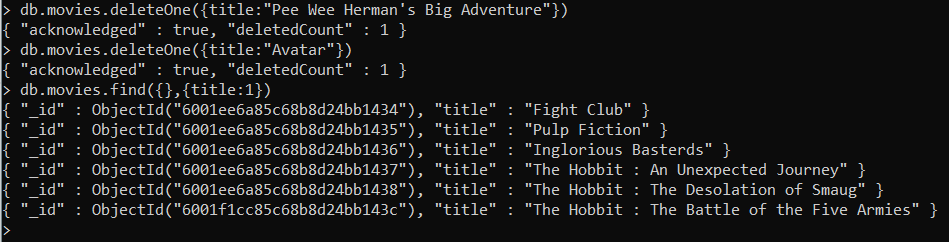
Delete Documents

1. delete the movie “Pee Wee Herman’s Big Adventure”

: db.movies.deleteOne({title: “Pee Wee Herman’s Big Adventure”})

1. delete the movie "Avatar"

: db.movies.deleteOne({title: “Avatar”})



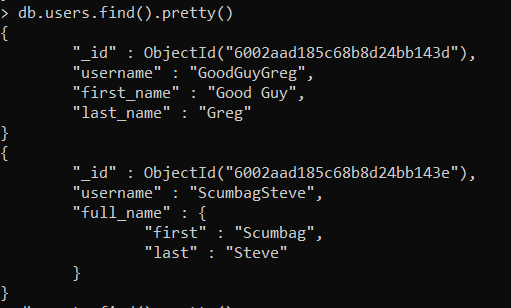
Relationships

Inserted all the elements

Querying related collection

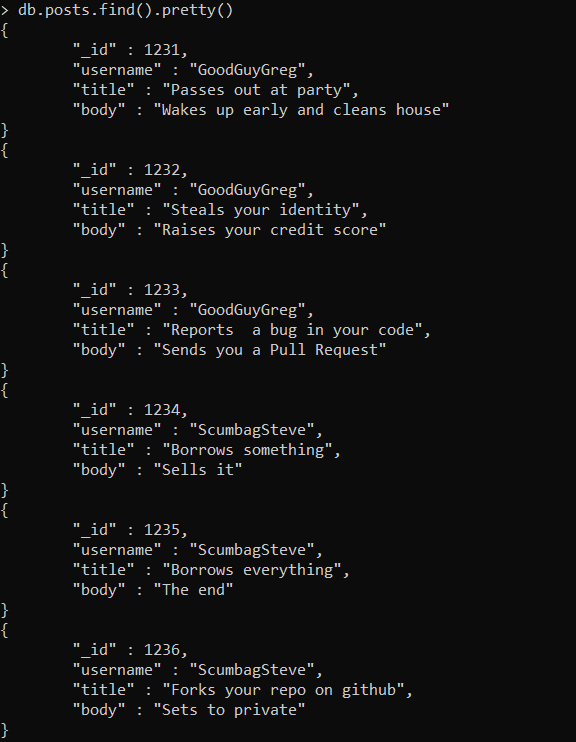
1. find all users

: db.users.find().pretty()



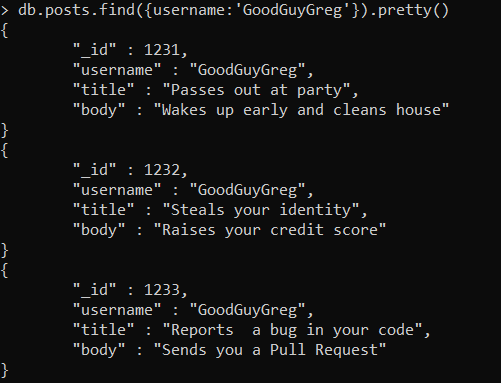
1. find all posts

: db.posts.find().pretty()



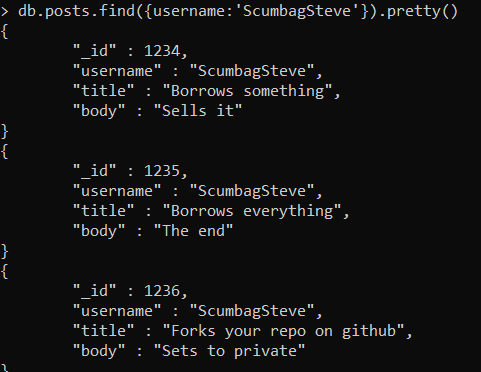
1. find all posts that was authored by "GoodGuyGreg"

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



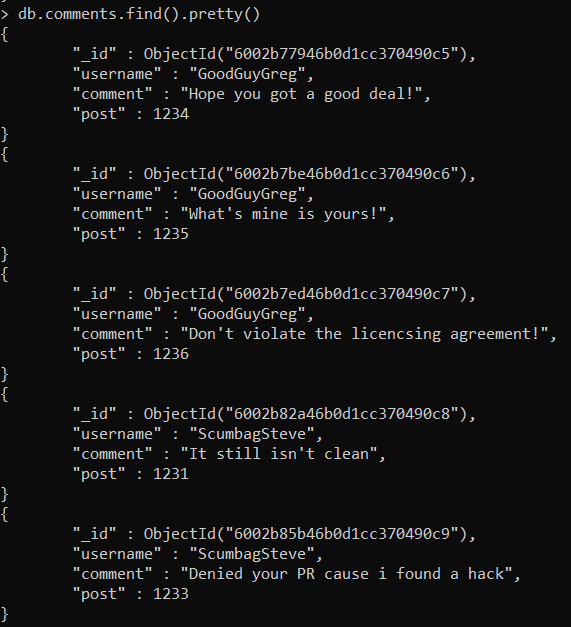
1. find all posts that was authored by "ScumbagSteve"

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



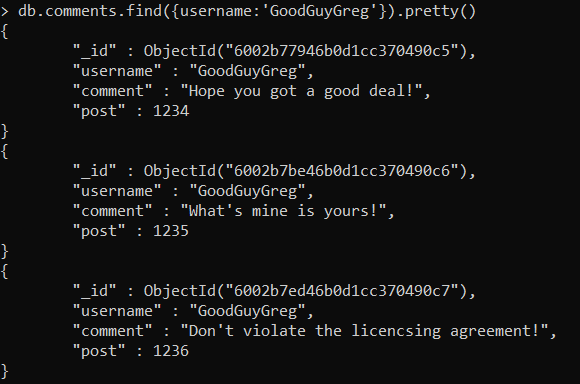
1. find all comments

: db.comments.find().pretty()



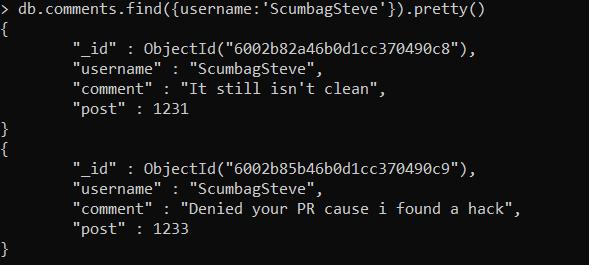
1. find all comments that was authored by "GoodGuyGreg"

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



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

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



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