**MongoDb Assignment-1**

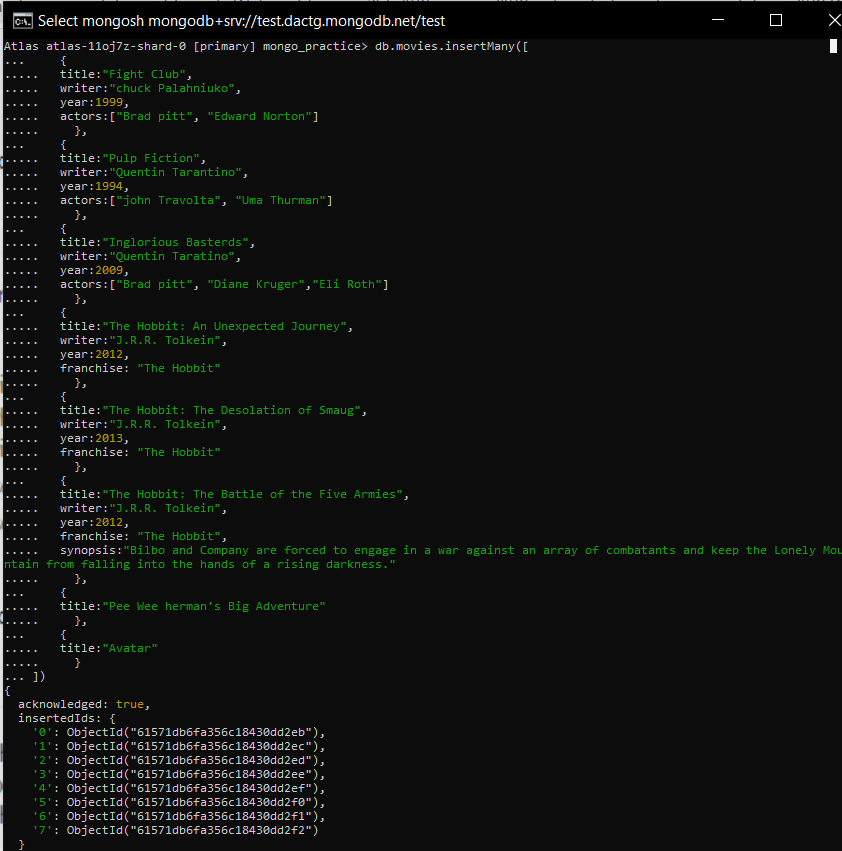
create database

Connect to a running mongo instance, use a database named **mongo-practice**.

-🡪 use mongo\_practice

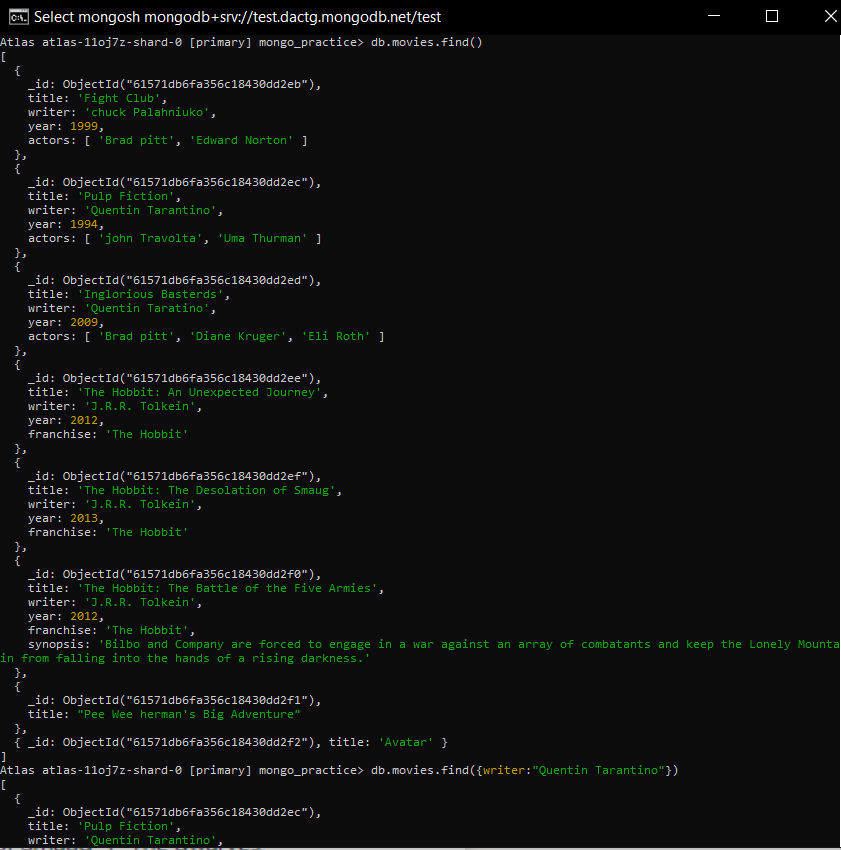
## Insert Documents

Insert the following documents into a movies collection.

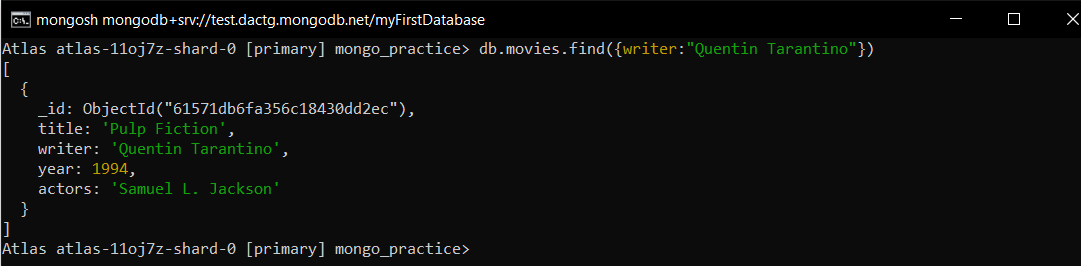


## Query / Find Documents

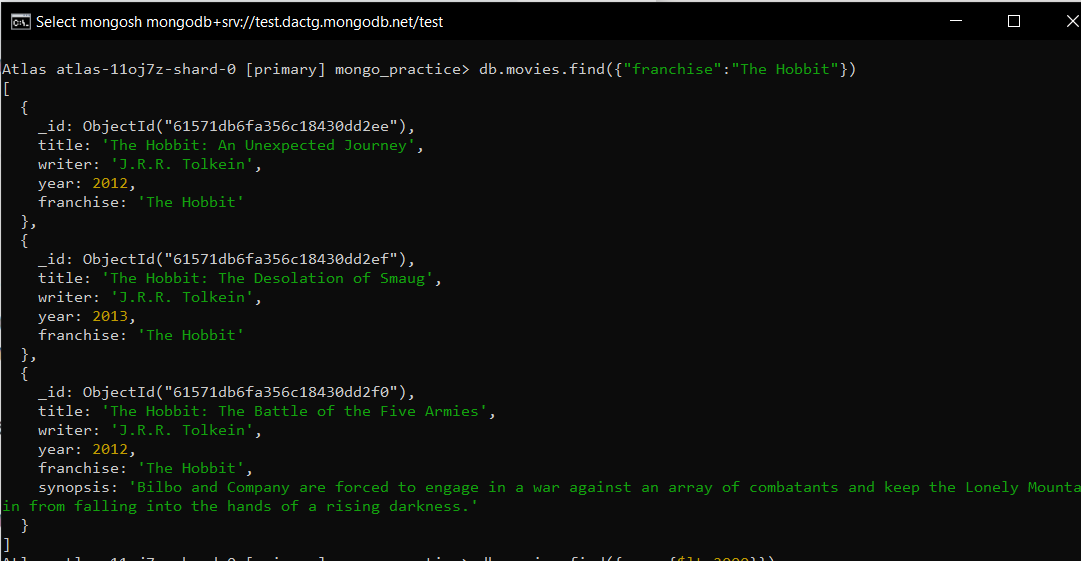
1. get all documents



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



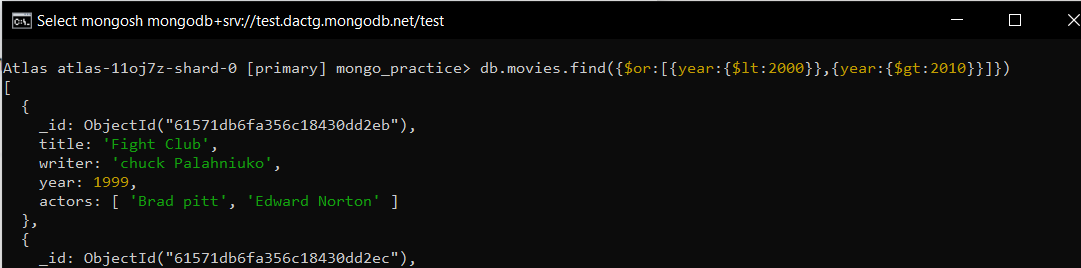
1. get all documents where actors include "Brad Pitt"
   * db.movies.find({actors:"Brad Pitt"})
2. get all documents with franchise set to "The Hobbit"



1. get all movies released in the 90s

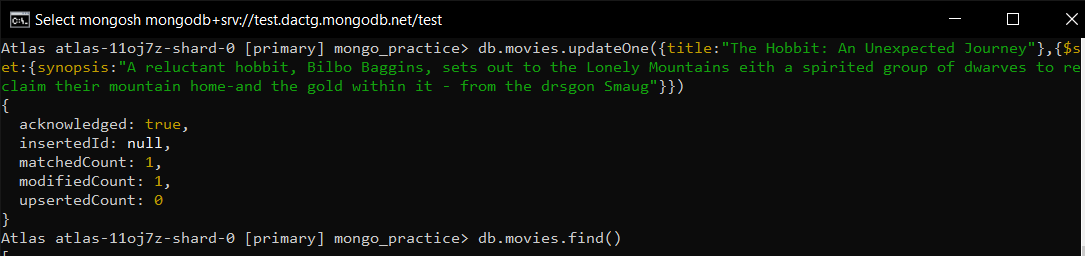


1. get all movies released before the year 2000 or after 2010

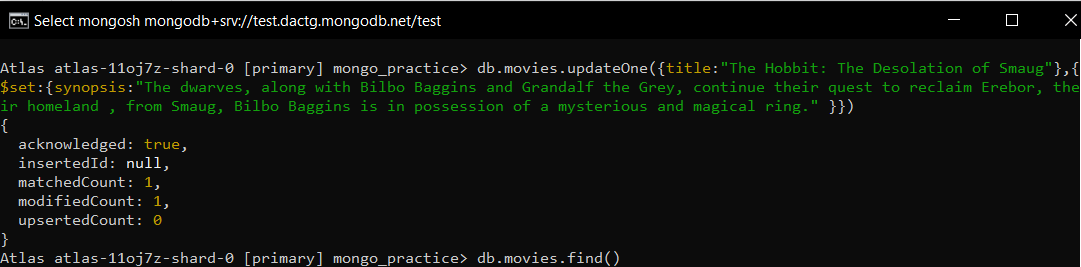


Update Documents

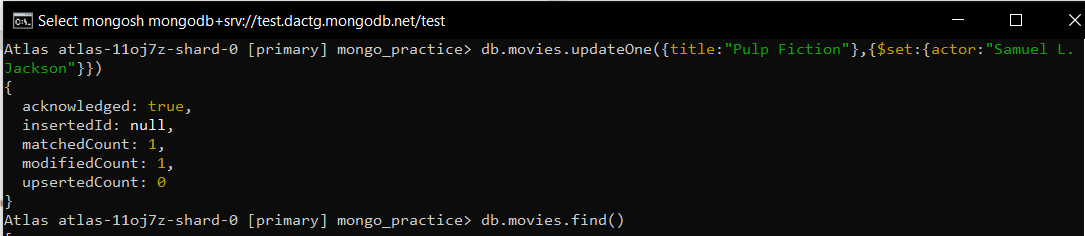
1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."



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."

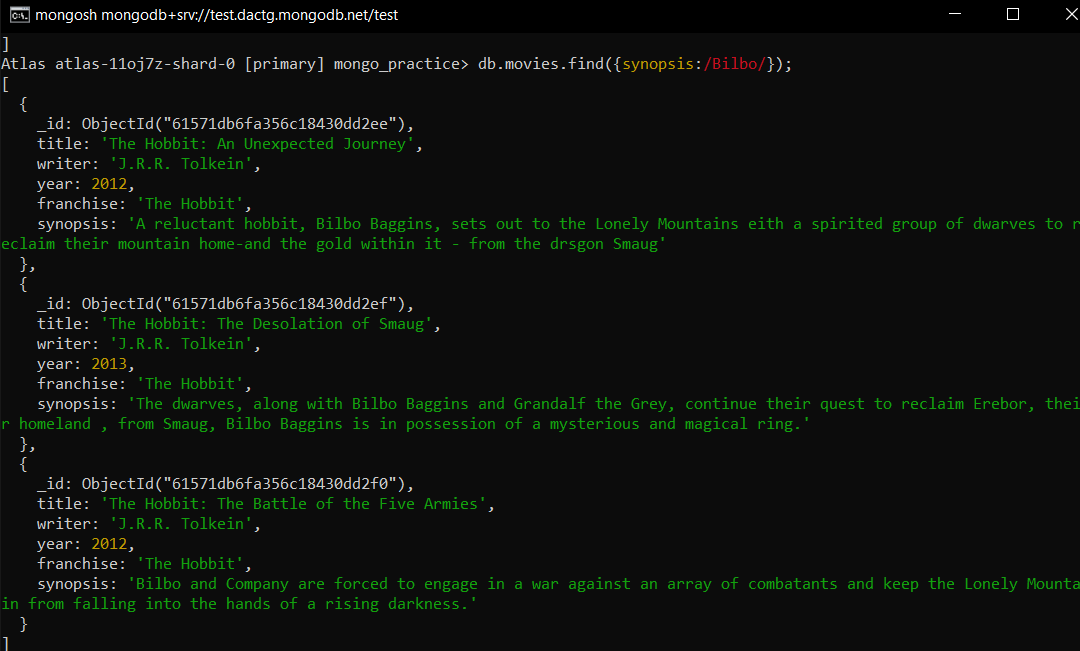


1. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

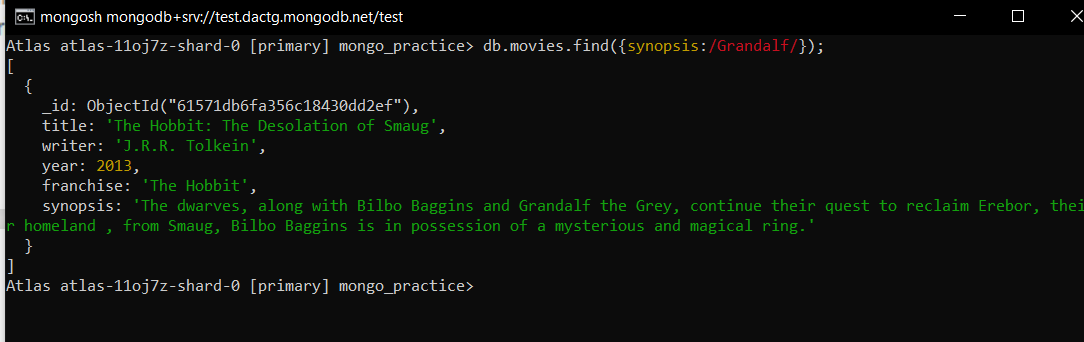


Text Search

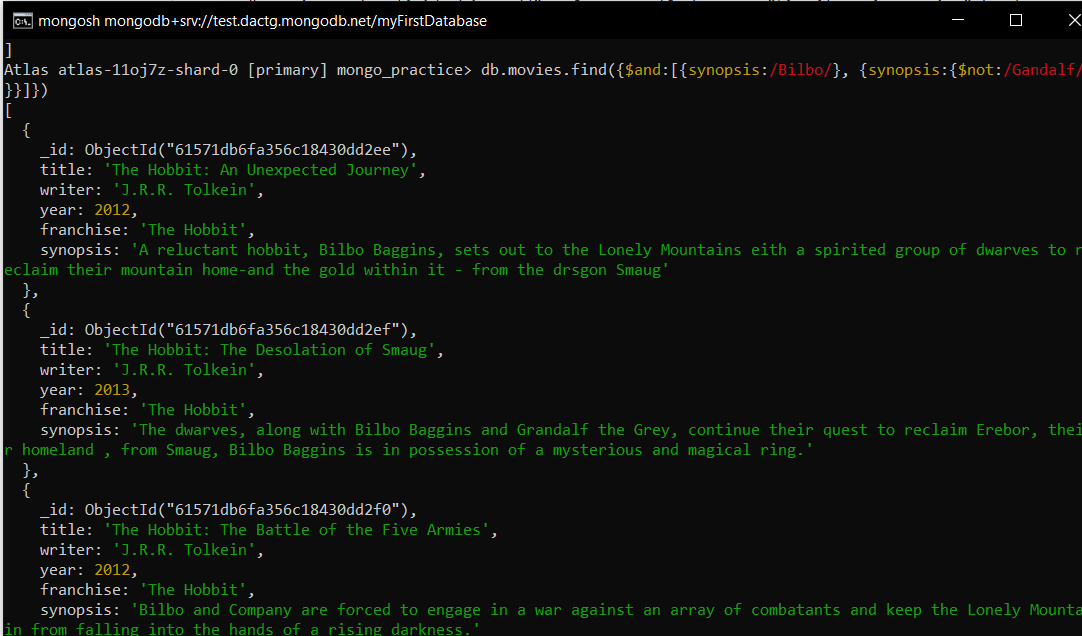
1. find all movies that have a synopsis that contains the word "Bilbo"



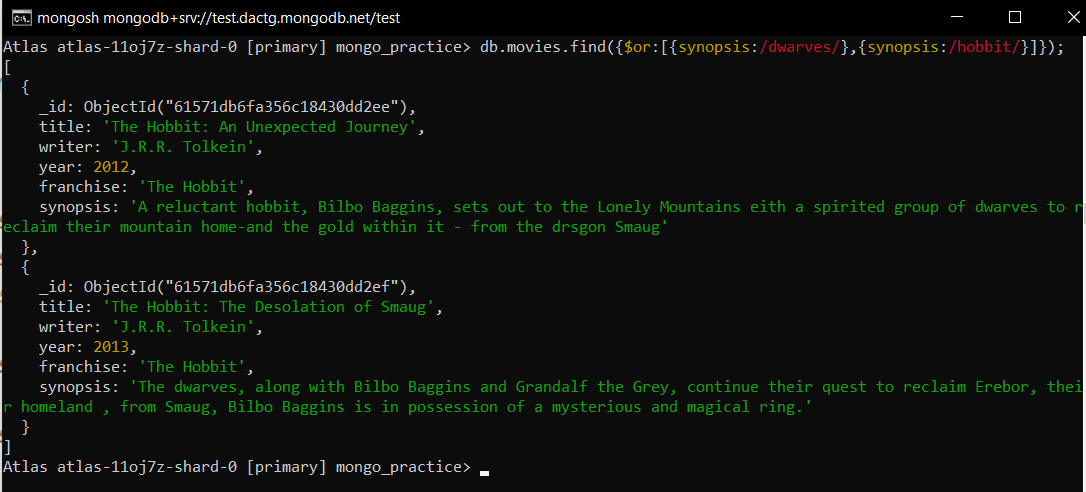
1. find all movies that have a synopsis that contains the word "Gandalf"



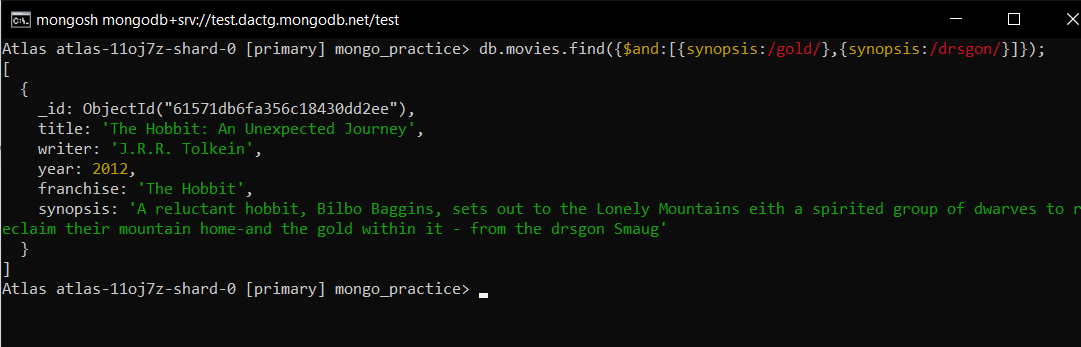
1. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"



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

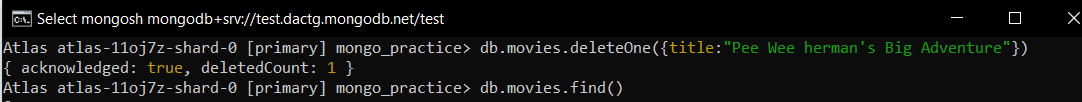


1. find all movies that have a synopsis that contains the word "gold" and "dragon"

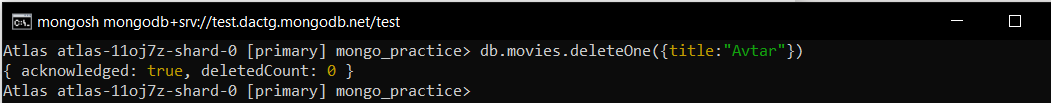


Delete Documents

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



1. delete the movie "Avatar"



## Relationships

### Insert the following documents into a **user** collection.

username : GoodGuyGreg

first\_name : "Good Guy"

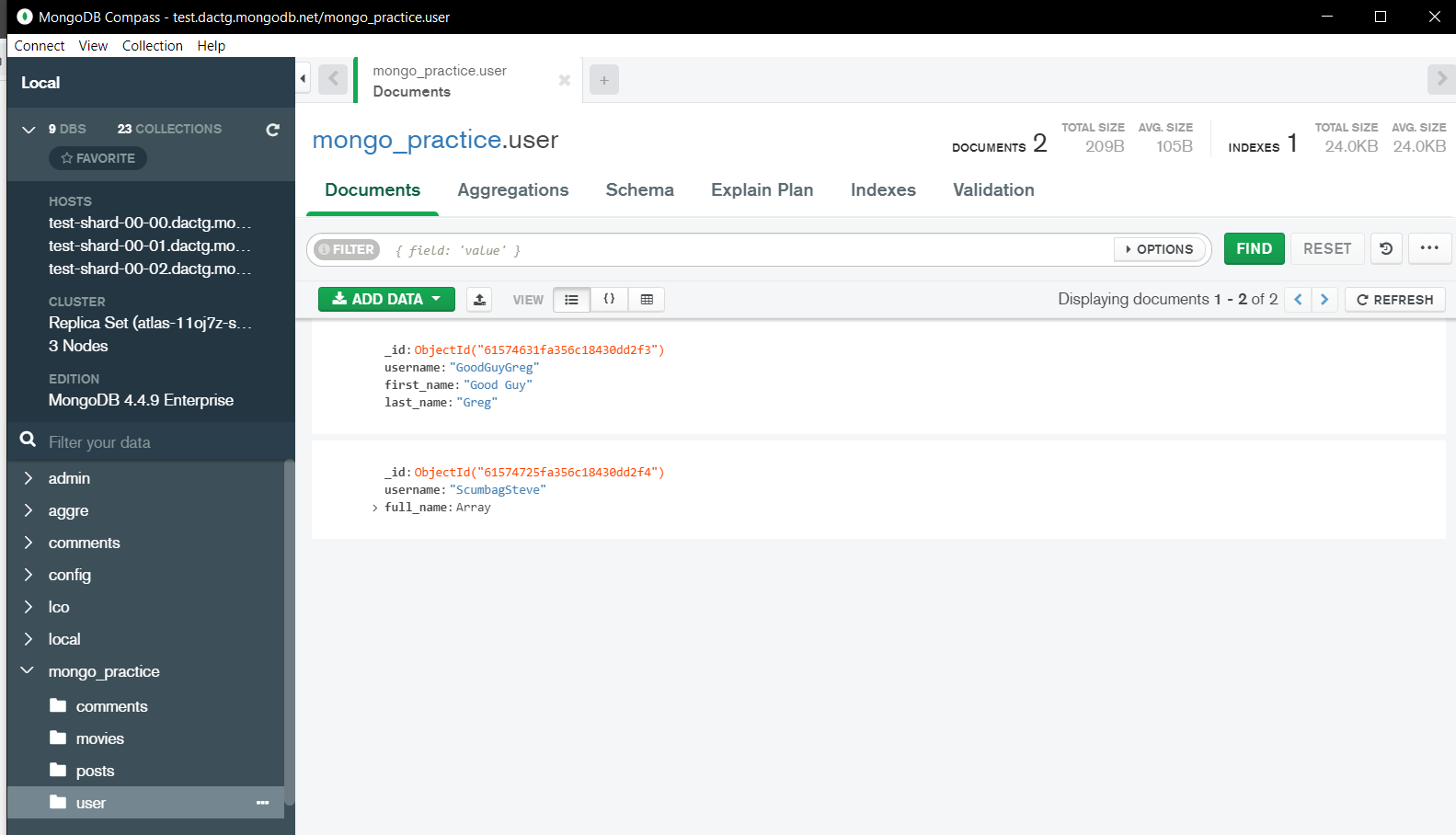
last\_name : "Greg"

username : ScumbagSteve

full\_name :

first : "Scumbag"

last : "Steve"



### 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

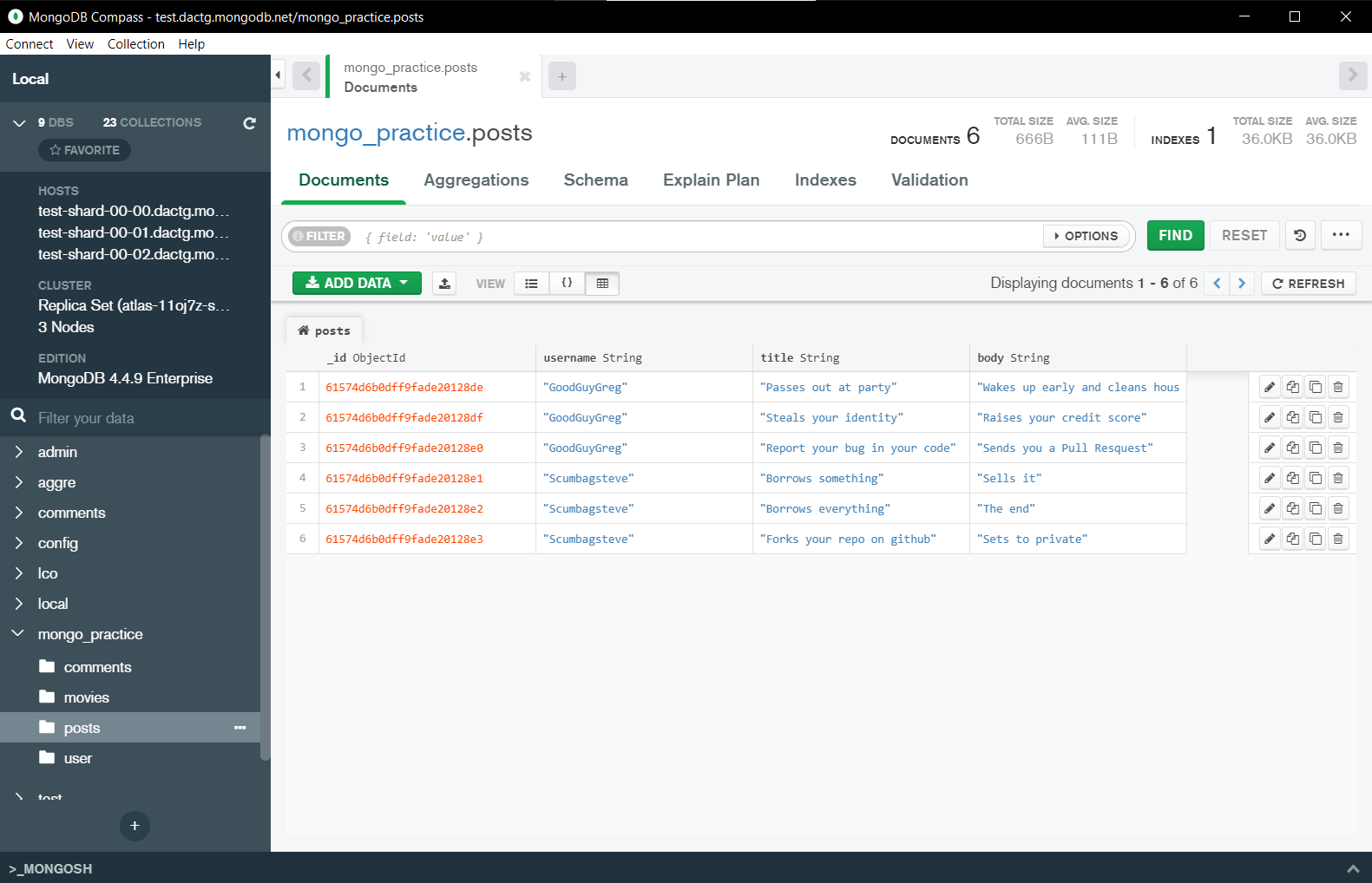
title : Borrows everything

body : The end

username : ScumbagSteve

title : Forks your repo on github

body : Sets to private



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

username : GoodGuyGreg

comment : Hope you got a good deal!

post : [post\_obj\_id]

username : GoodGuyGreg

comment : What's mine is yours!

post : [post\_obj\_id]

username : GoodGuyGreg

comment : Don't violate the licensing agreement!

post : [post\_obj\_id]

username : ScumbagSteve

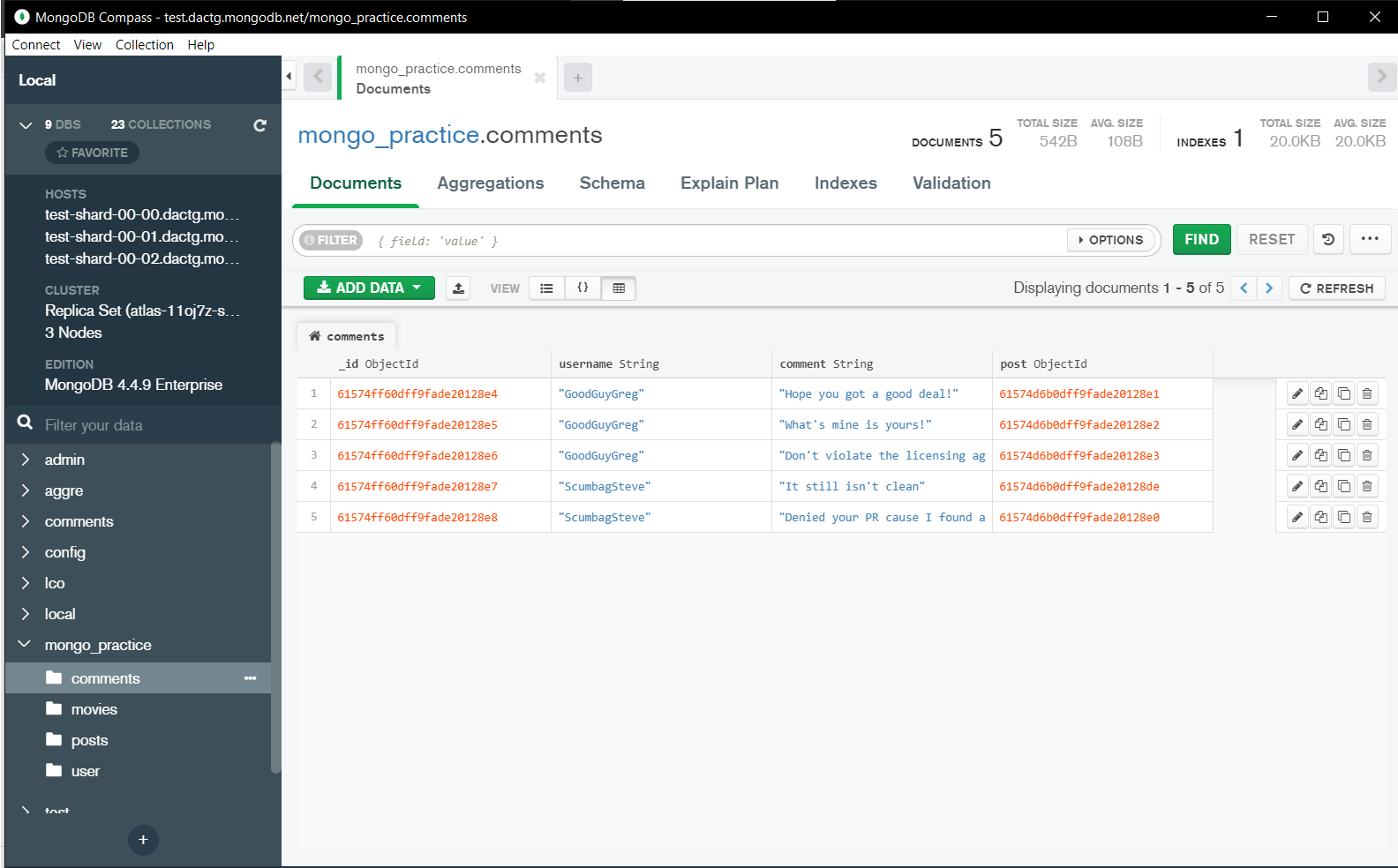
comment : It still isn't clean

post : [post\_obj\_id]

username : ScumbagSteve

comment : Denied your PR cause I found a hack

post : [post\_obj\_id]

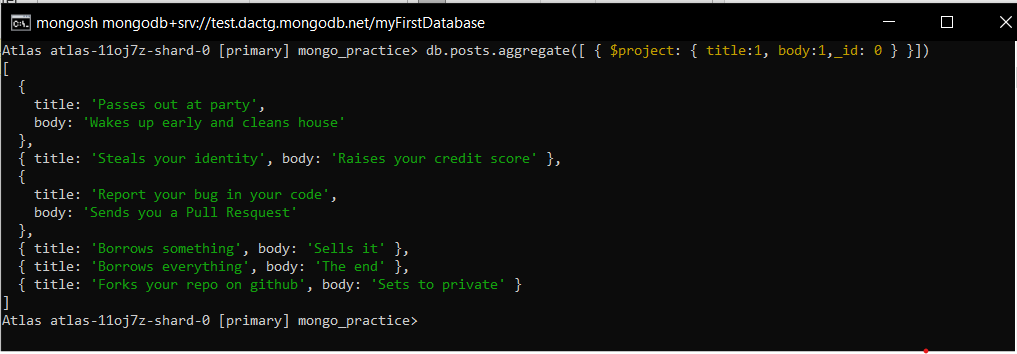


## Querying related collections

1. find all users

## 

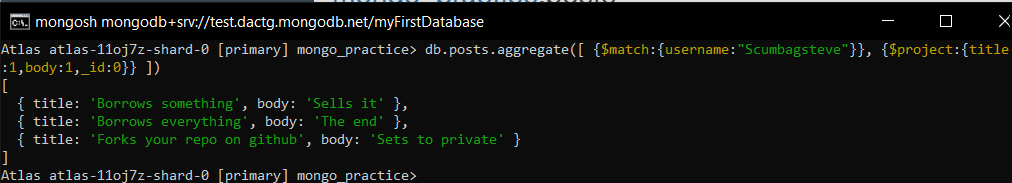
1. find all posts



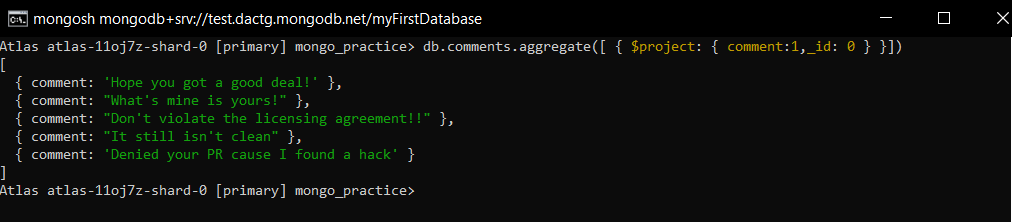
1. find all posts that was authored by "GoodGuyGreg"

## 

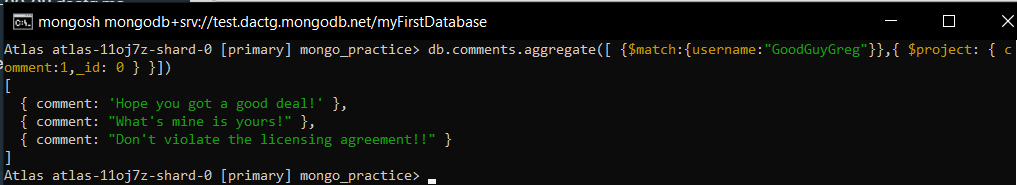
1. find all posts that was authored by "ScumbagSteve"



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"

