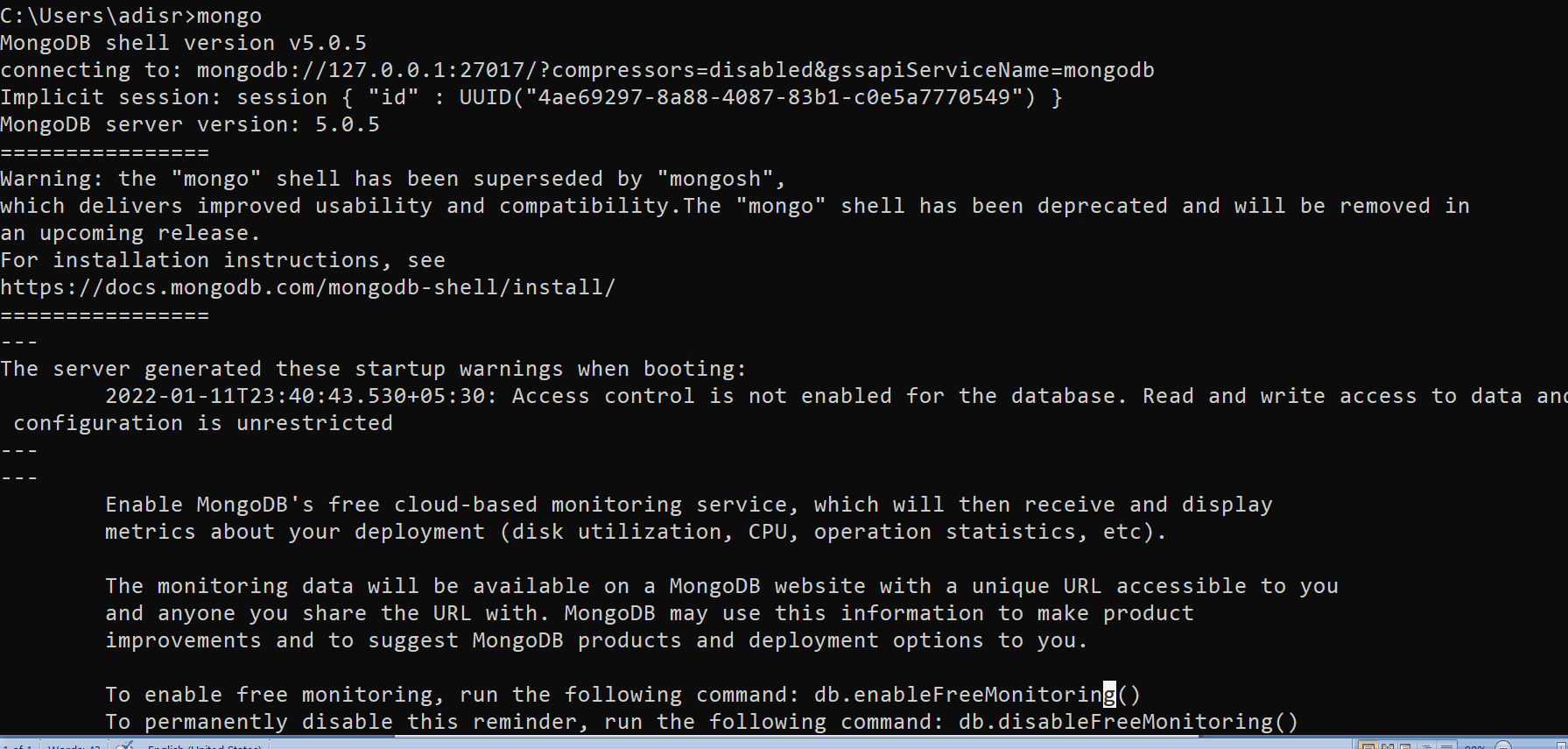
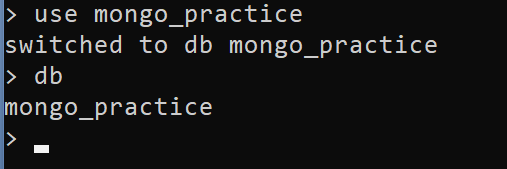
**MongoDB Lab Assignments -Day 1  
Adisri Sarode**

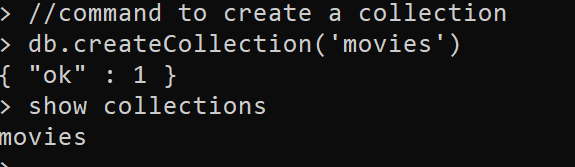
**SECTION 1 - MongoDB Exercise in mongo shell**

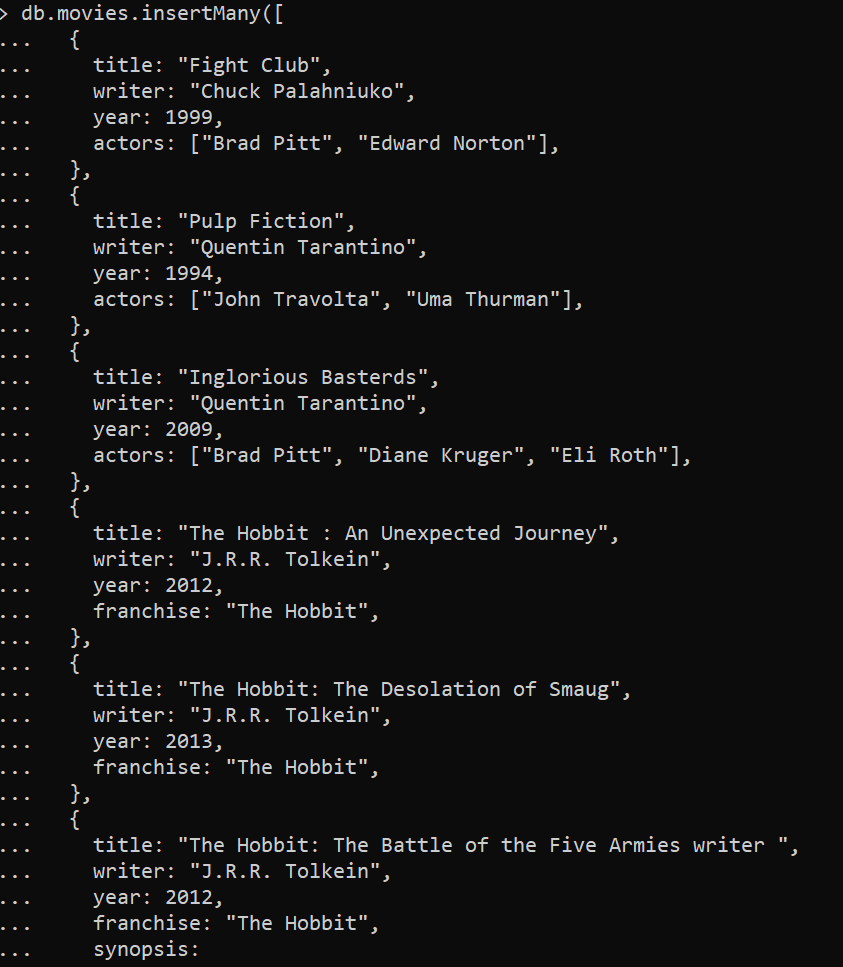
Connect to a running mongo instance, use a database named **mongo\_practice**.

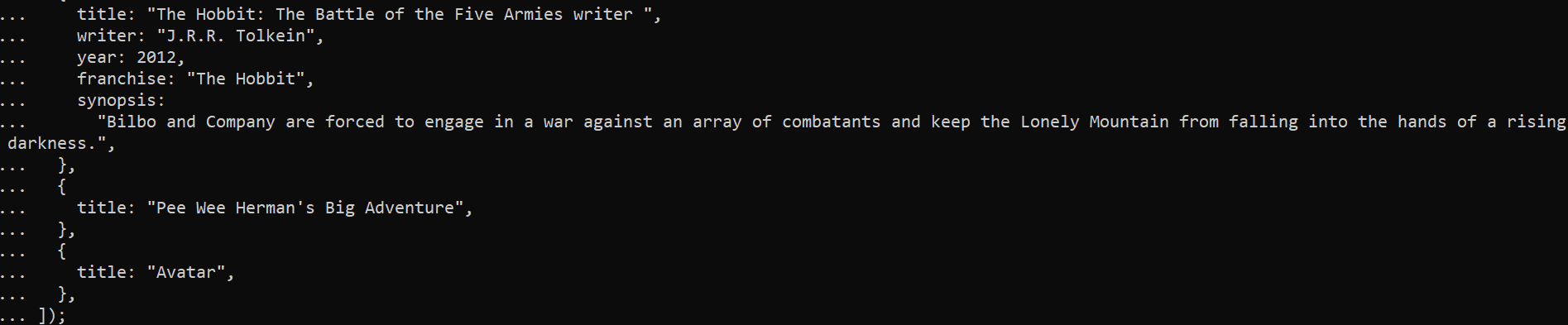
Document all your queries in a JavaScript file to use as a reference.  
  
Command to open mongo shell in command prompt: **mongo**  
  
Command to create and switch to database named **mongo\_practice**



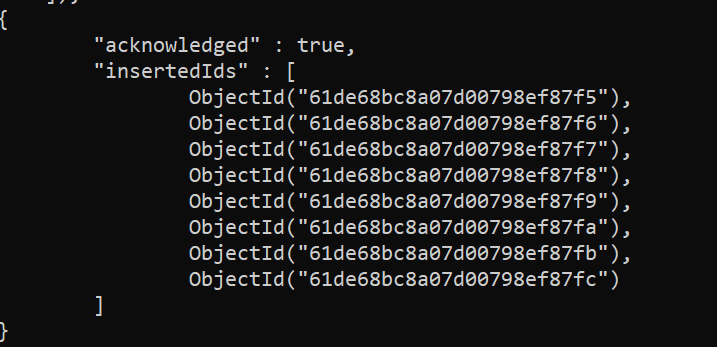
**SECTION 2- Insert Documents**

Insert the following documents into a **movies** collection  
  
****  
  
To insert the provided data in the collection ‘movies’:

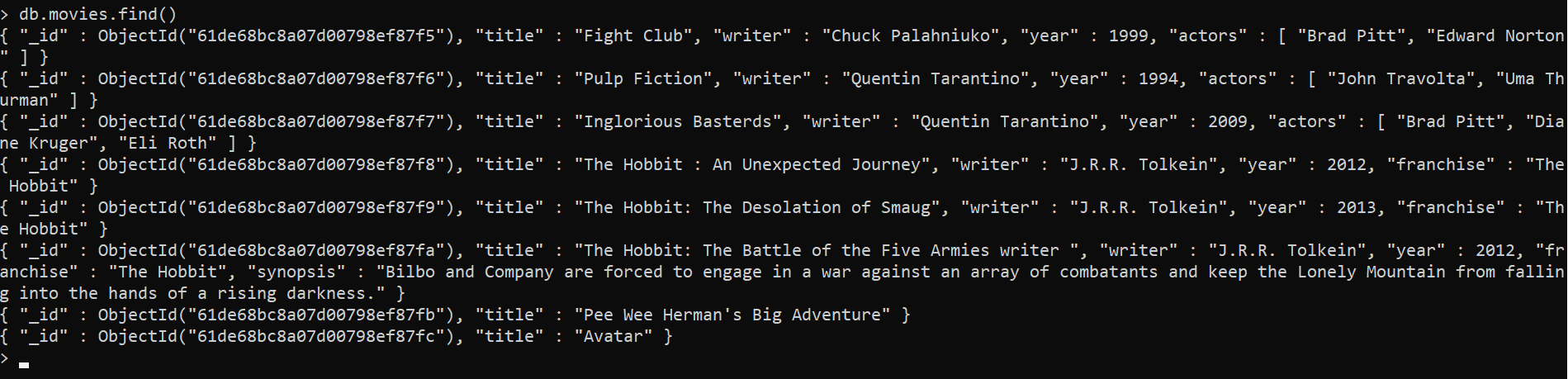
****

****

All rows have been inserted with unique object id.

****

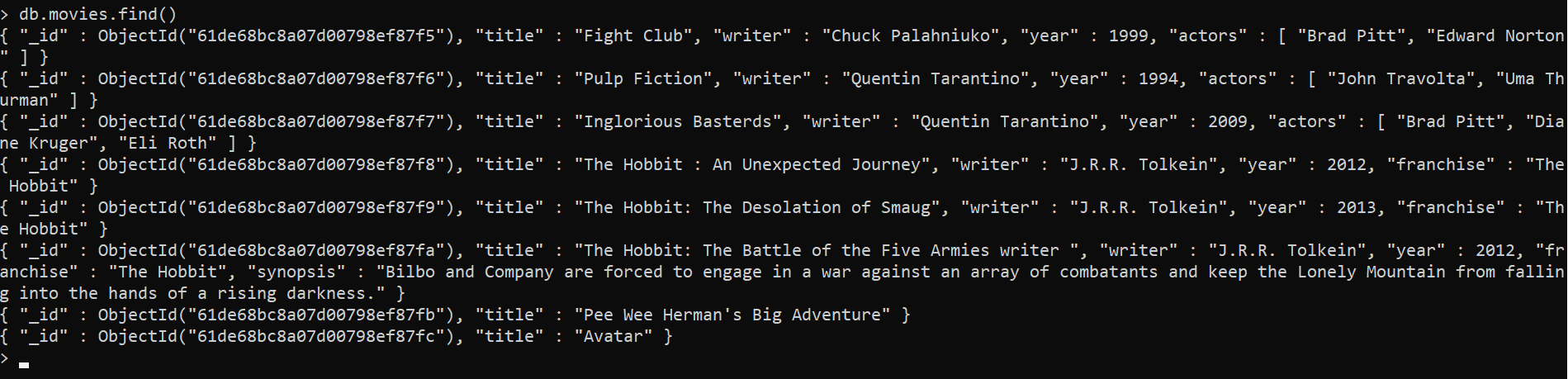
To display all the inserted fields in the collection.

****

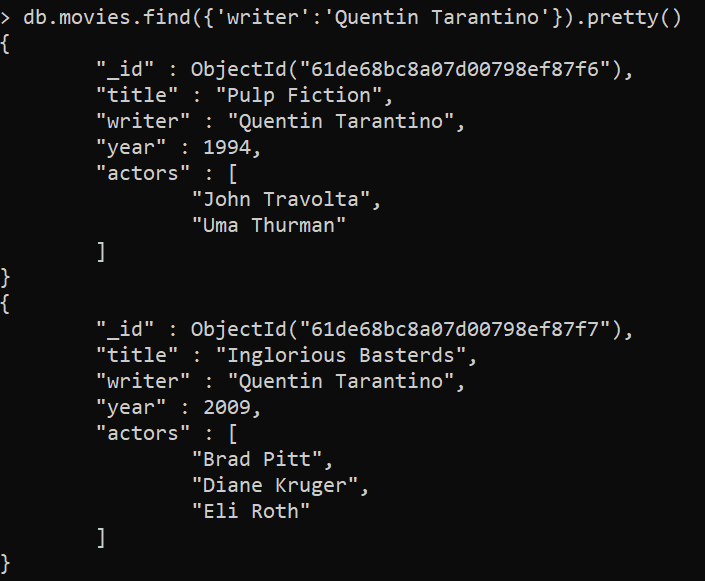
**MongoDB Compass interface:  
**

**  
  
Query / Find Documents**query the **movies** collection to

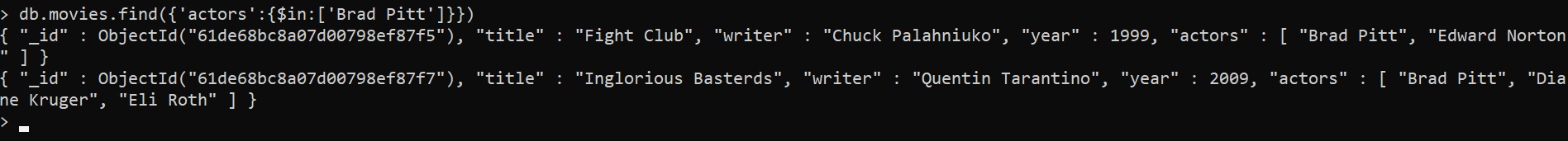
1. get all documents

**Command:** db.movies.find()  


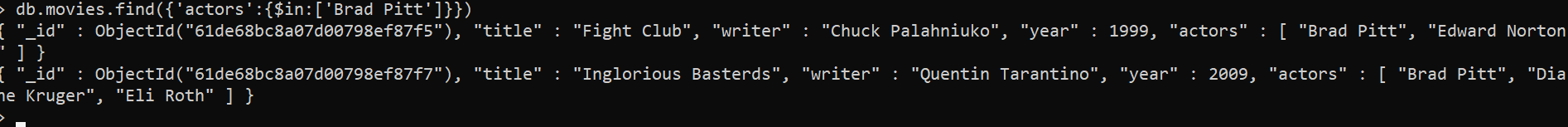
2. get all documents with writer set to "Quentin Tarantino"



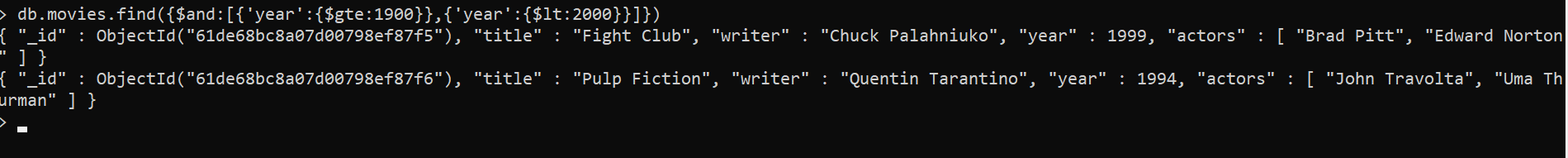
3. get all documents where actors include "Brad Pitt"



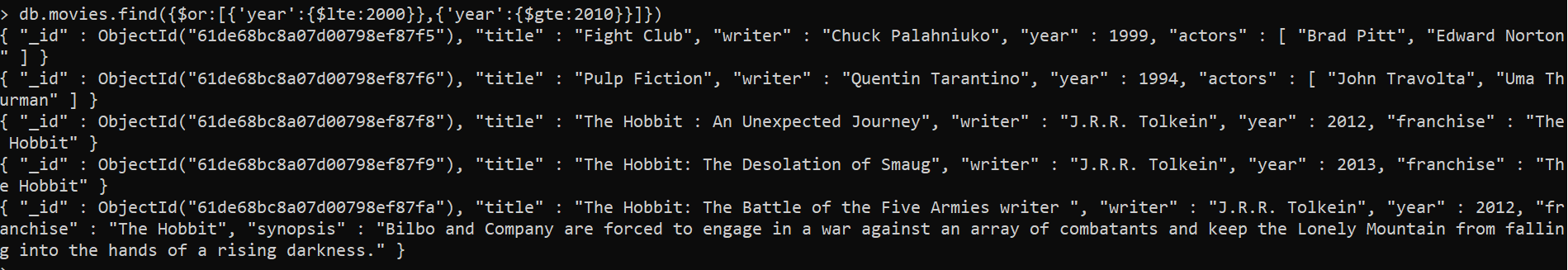
4. get all documents with franchise set to "The Hobbit"



5. get all movies released in the 90s

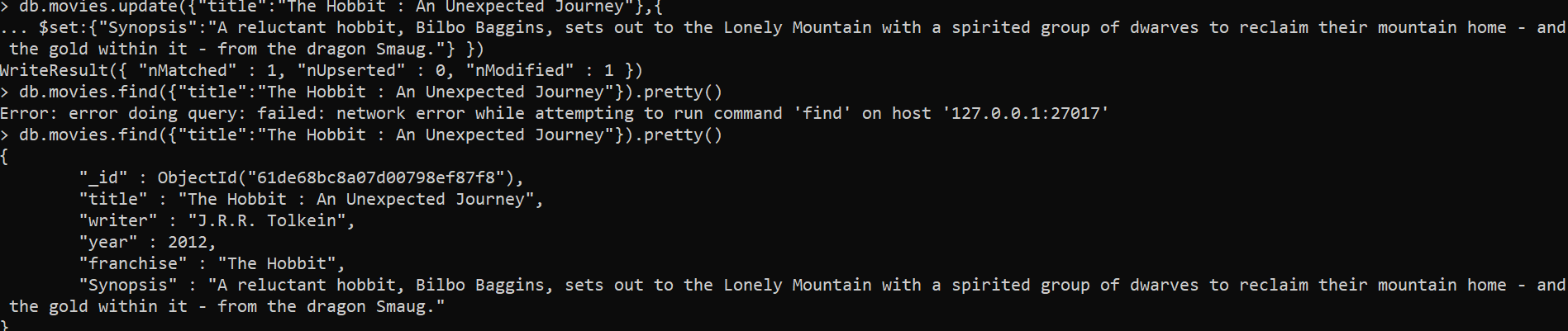


6. get all movies released before the year 2000 or after 2010

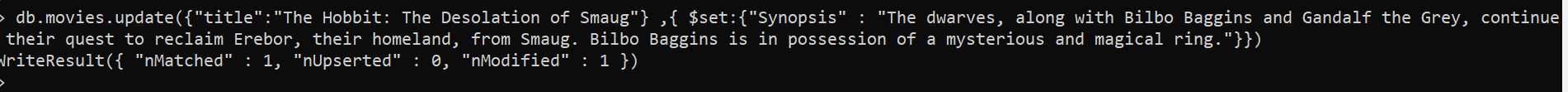


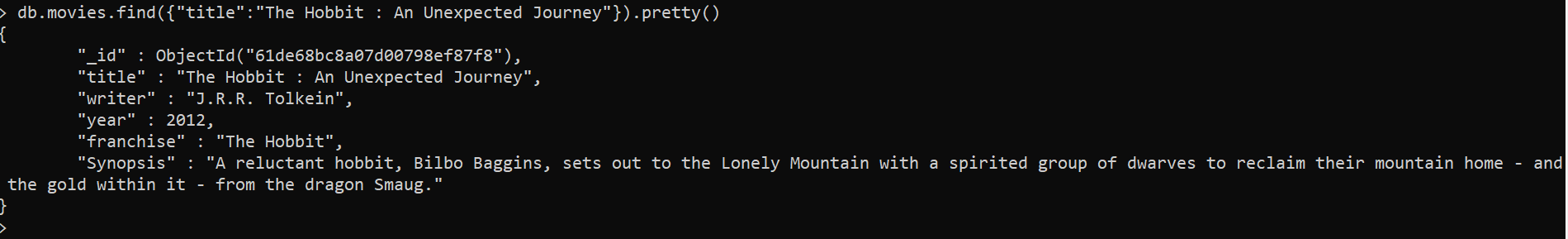
**Section 3- Update Documents**

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."

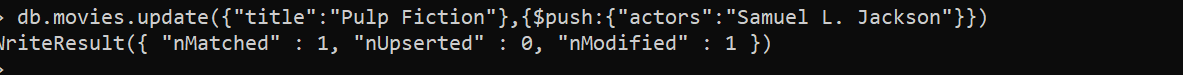
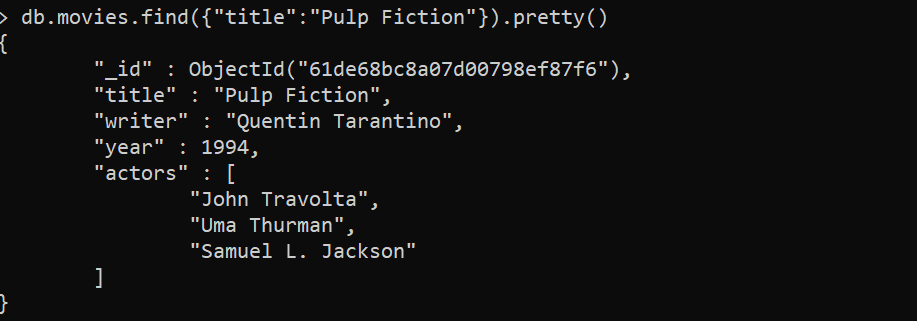


2. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."





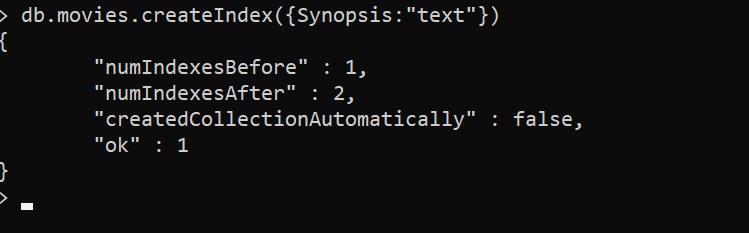
3. Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

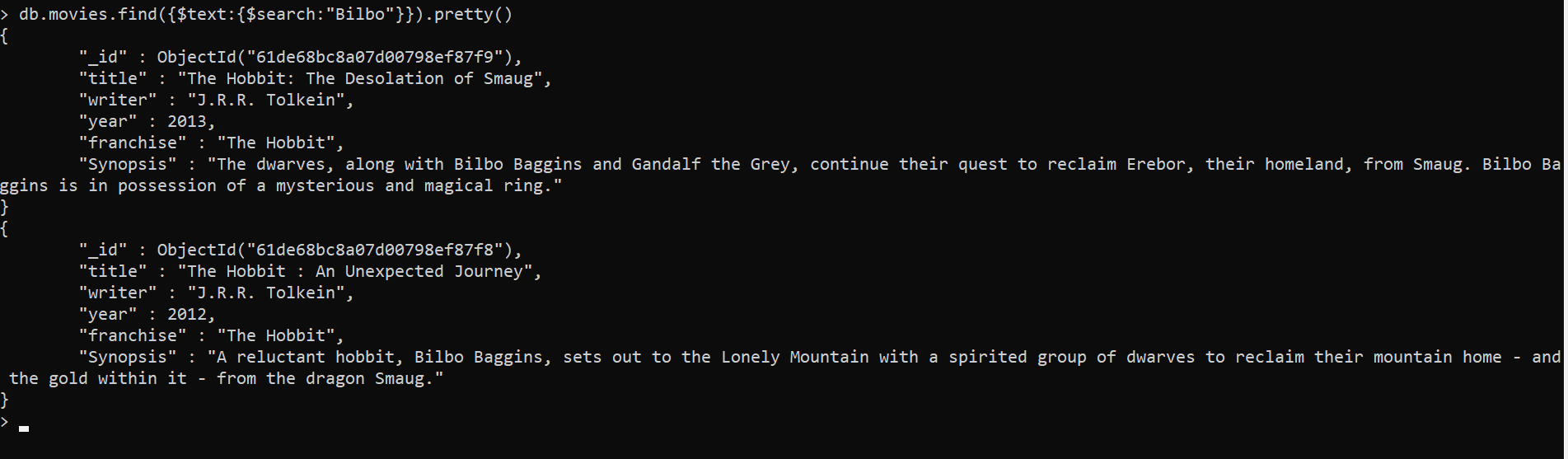
  
  


**Section 4 - Text Search**

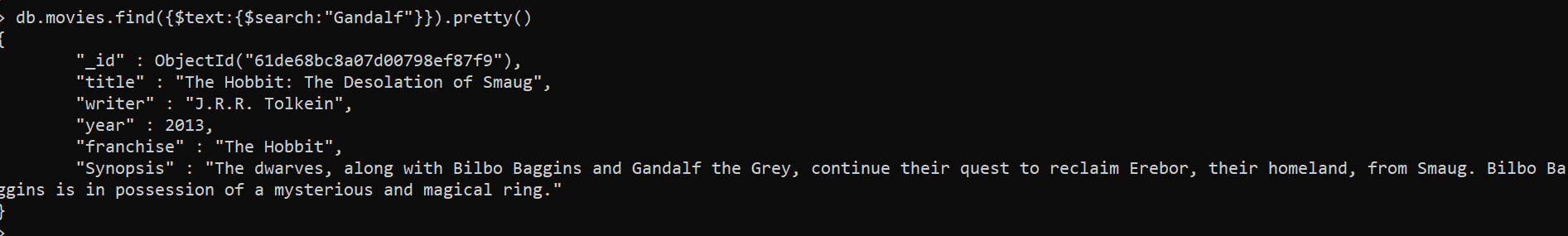
1. find all movies that have a synopsis that contains the word "Bilbo"

Creating text index:

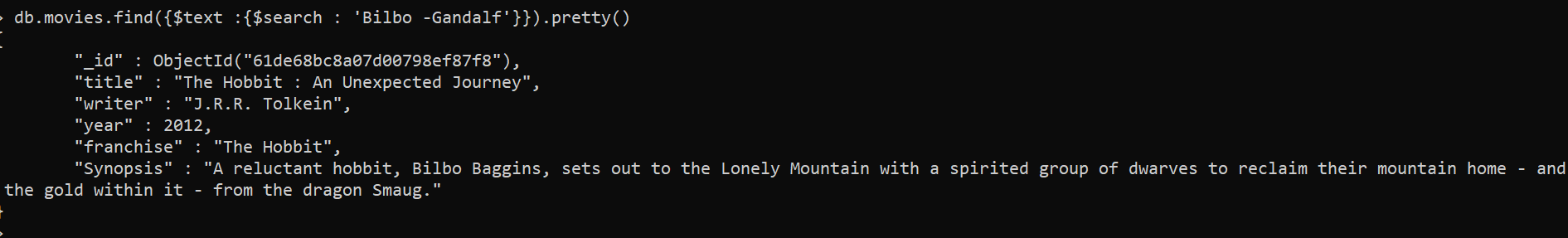




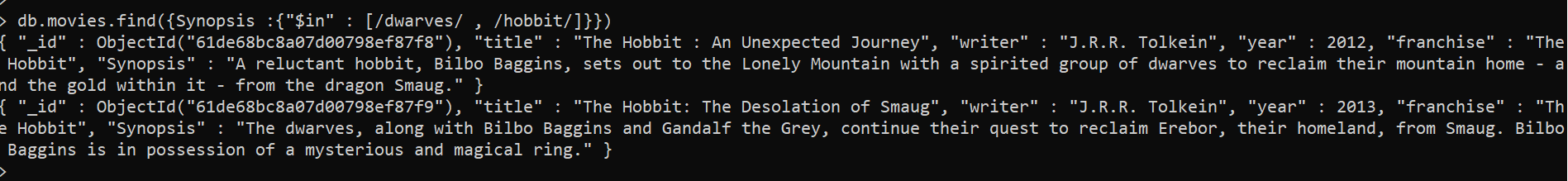
2. find all movies that have a synopsis that contains the word "Gandalf"



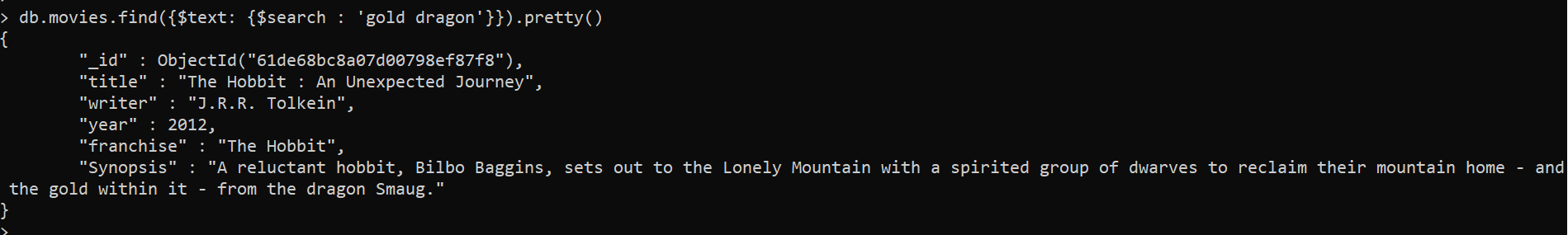
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"



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

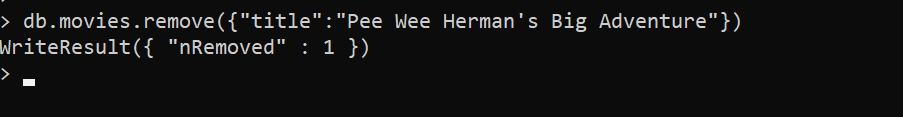


5. find all movies that have a synopsis that contains the word "gold" and "dragon"

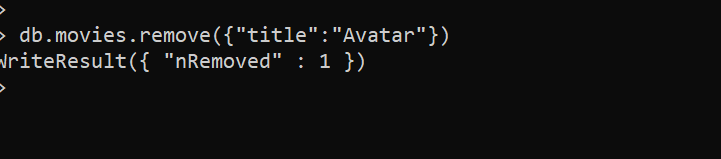


**Section 5-Delete Documents**

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



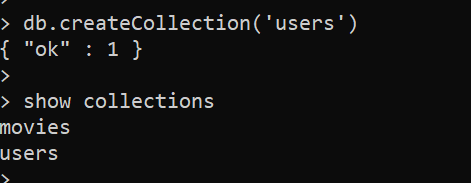
2. delete the movie "Avatar"



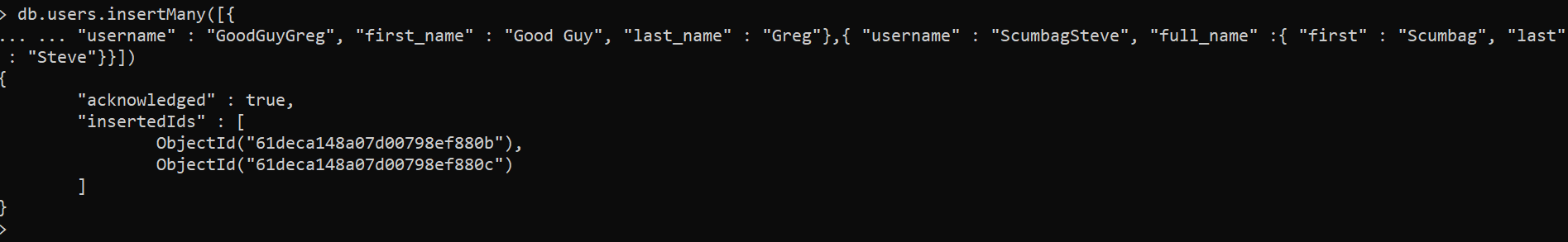
**Section 6 – Relationships**

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

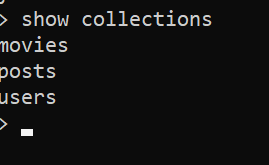
Create a new collection named ‘users’ using the command **db.createCollection (‘users’)**

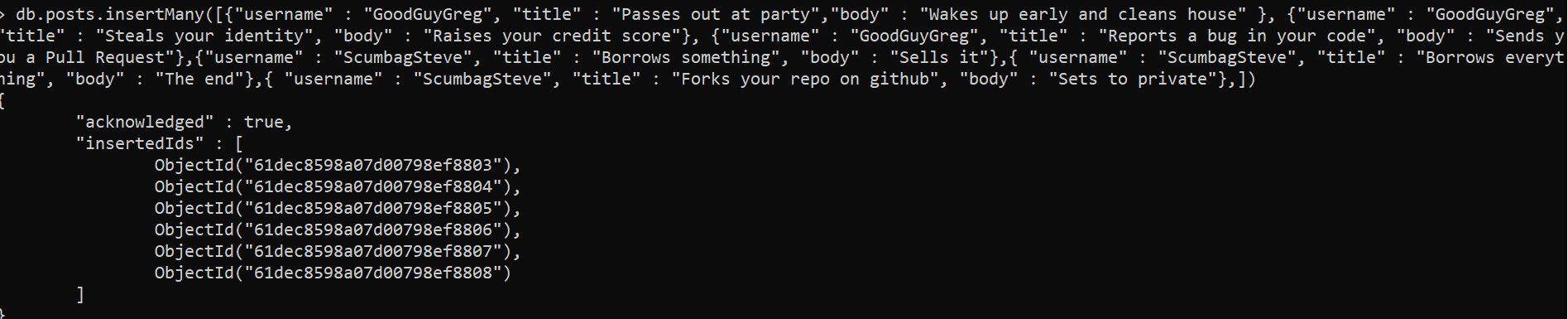


Insert the following documents into a **users** collection

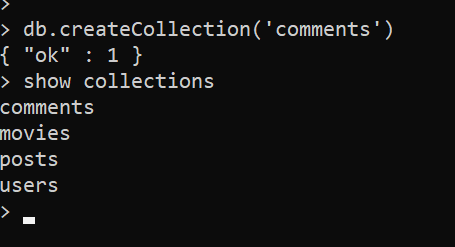


Insert the following documents into a **posts** collection

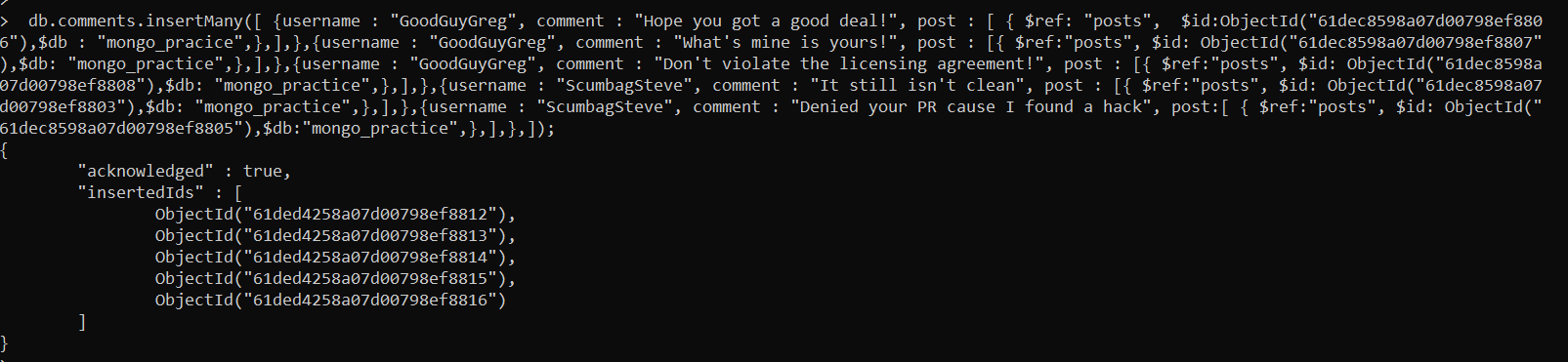
command : **db.createCollection(‘posts’)  
  
**Insert rows inside the posts collection



Insert the following documents into a **comments** collection  
command: **db.createCollection(‘comments’)**

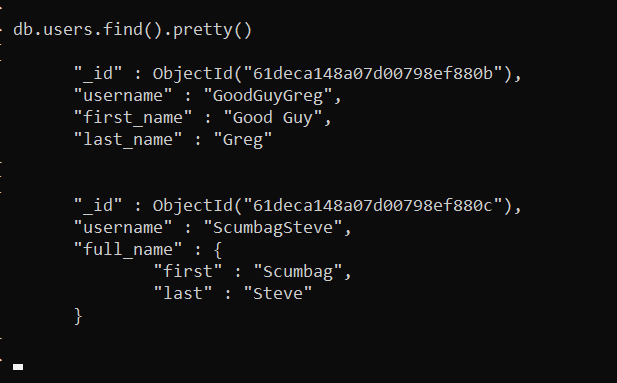
****

Insert rows inside the comments collection

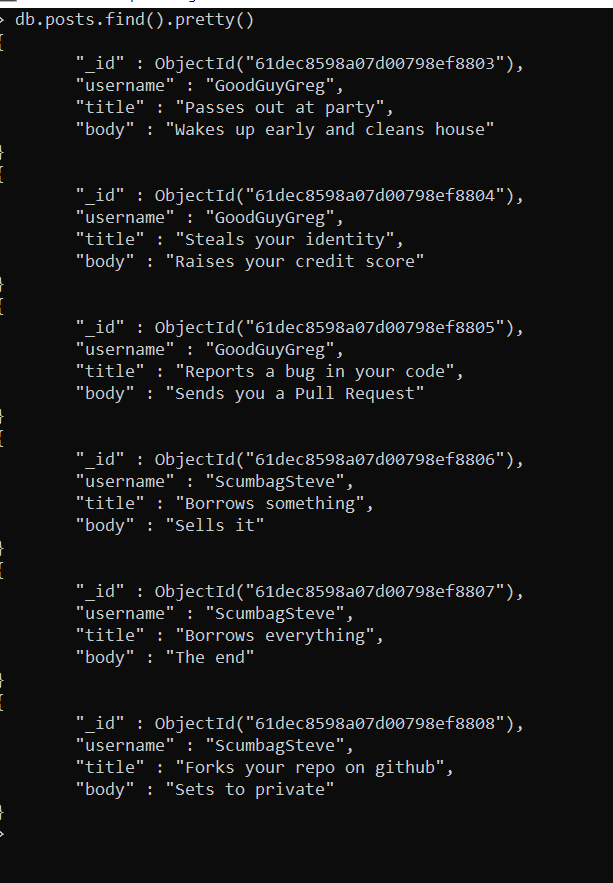


**Querying related collections**

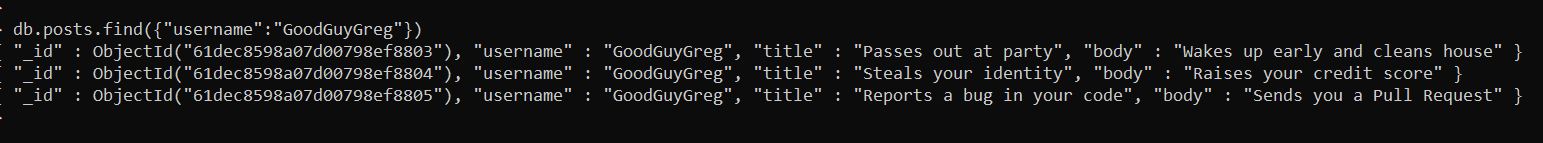
1. find all users



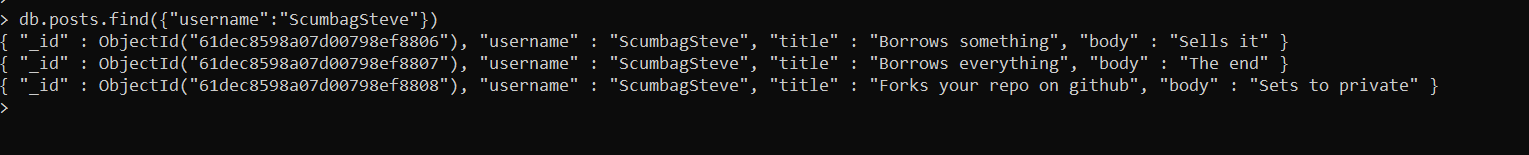
2. find all posts



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



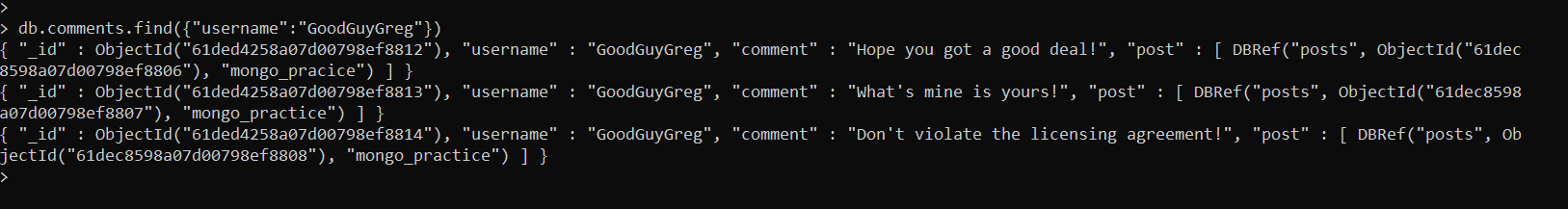
4. find all posts that was authored by "ScumbagSteve"



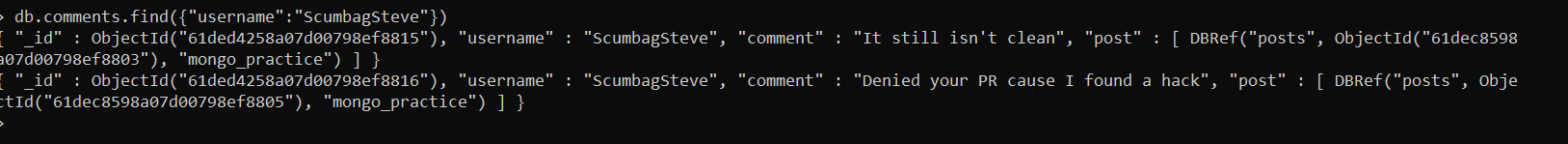
5. find all comments



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



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



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

