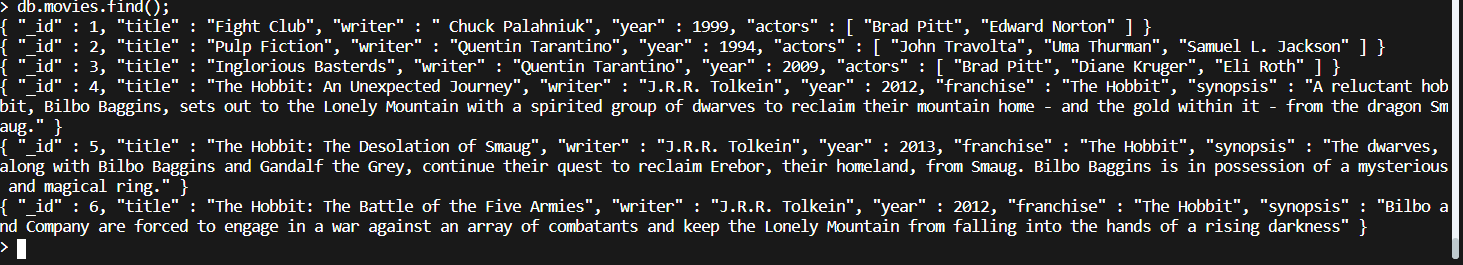
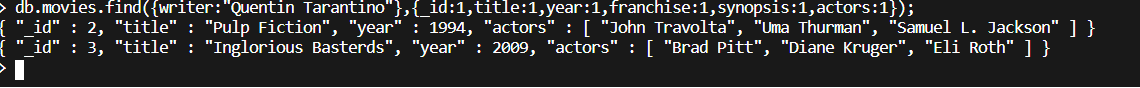
**Mongo DB:**

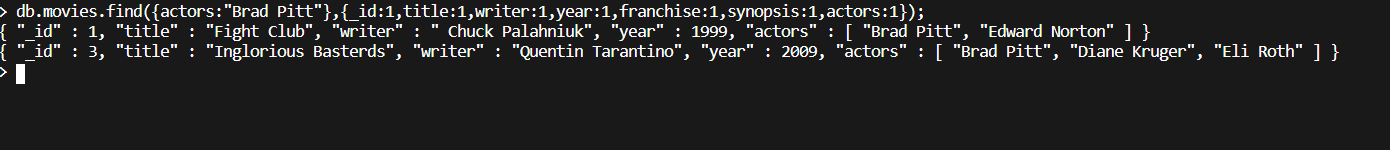
1. 1. get all documents : > **db.movies.find();**



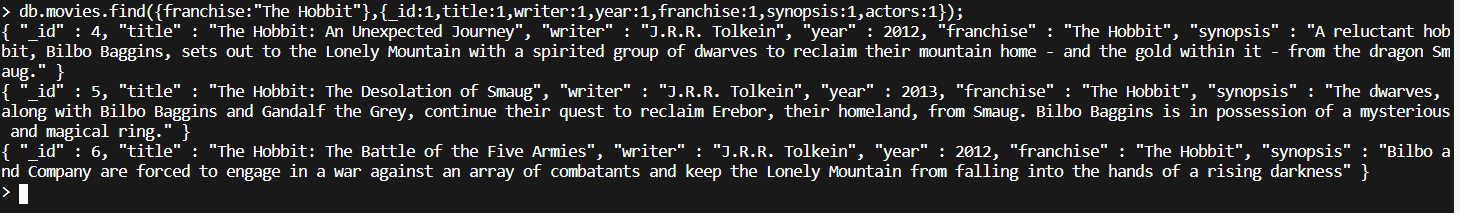
1. 2. get all documents with `writer` set to "Quentin Tarantino" : > **db.movies.find({writer:"Quentin Tarantino"},{\_id:1,title:1,year:1,franchise:1,synopsis:1,actors:1});**



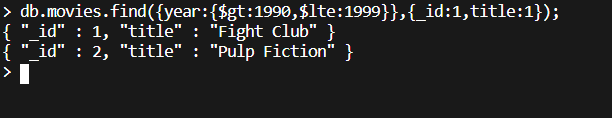
1. get all documents where `actors` include "Brad Pitt" ; = **> db.movies.find({actors:"Brad Pitt"},{\_id:1,title:1,writer:1,year:1,franchise:1,synopsis:1,actors:1});**



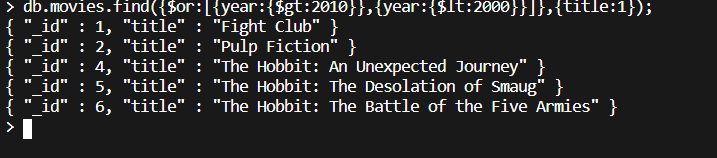
1. 4. get all documents with `franchise` set to "The Hobbit"= > **db.movies.find({franchise:"The Hobbit"},{\_id:1,title:1,writer:1,year:1,franchise:1,synopsis:1,actors:1});**



1. . get all movies released in the 90s : **db.movies.find({year:{$gt:1990,$lte:1999}},{\_id:1,title:1});**



1. . get all movies released before the year 2000 or after 2010 : **db.movies.find({$or:[{year:{$gt:2010}},{year:{$lt:2000}}]},{title:1});**



**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.updateMany({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."}});**

2) 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." :

**db.movies.updateMany({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."}});**

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

**> db.movies.update({title:”Pulp Fiction”},{$addToSet:{actors :”Samuel L. Jackson”}});**

**Delete Documents**

1. delete the movie "Pee Wee Herman's Big Adventure" : **> db.movies.remove({title : "Pee Wee Herman's Big Adventure"});**

2. delete the movie "Avatar" : - > **db.movies.remove({title : "Avatar"});**

**Relationships**

**db.users.insertMany([{\_id:1, username : "GoodGuyGreg", first\_name : "Good Guy",last\_name : "Greg"} , {\_id:2,username : "ScumbagSteve",first\_name : "Scumbag",last\_name : "Steve"}]);**

**db.posts.insertMany([{\_id:10 , username : "GoodGuyGreg", title : "Passes out at party", body : "Wakes up early and cleans house"}, {\_id:11 , username : "GoodGuyGreg", title : "Steals your identity", body : "Raises your credit score"},{\_id:12, username : "GoodGuyGreg", title : "Reports a bug in your code", body : "Sends you a Pull Request"},{\_id:13 , username : "ScumbagSteve", title : "Borrows something", body : "Sells it"},{\_id:14 , username : "ScumbagSteve" , title : "Borrows everything" , body : "The end"},{\_id:15 , username : "ScumbagSteve" , title : "Forks your repo on github", body : "Sets to private"}]);**

**db.comments.insertMany([{ \_id:20, username : "GoodGuyGreg", comment : "Hope you got a good deal!", post: db.posts.find()[3].\_id } , { \_id:21 , username : "GoodGuyGreg", comment : "What's mine is yours!",post : db.posts.find()[4].\_id } , { \_id:22 , username : "GoodGuyGreg", comment : "Don't violate the licensing agreement!", post : db.posts.find()[5].\_id } , { \_id:23 , username : "ScumbagSteve", comment : "It still isn't clean", post :db.posts.find()[0].\_id } , { \_id: 24 ,username : "ScumbagSteve", comment :" Denied your PR cause I found a hack" , post :db.posts.find()[2].\_id }]);**

**Querying related collections**

1. find all users : >> **db.users.find({},{first\_name:1, last\_name:1});**

2. find all posts : > > **db.posts.find();**

3. find all posts that was authored by "GoodGuyGreg" : > **db.posts.find({username: "GoodGuyGreg"},{});**

4. find all posts that was authored by "ScumbagSteve" : > **db.posts.find({username: "ScumbagSteve"},{});**

5. find all comments : **db.comments.aggregate([{$lookup:{from:"posts",localField:"post",foreignField:"\_id", as :"AllDetails"}}]).pretty();**

6. find all comments that was authored by "GoodGuyGreg" :> **db.comments.find({username:"GoodGuyGreg"});**

7. find all comments that was authored by "ScumbagSteve":> **db.comments.find({username:"ScumbagSteve"});**

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