



db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})

db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"2009", actors:["John Travolta", "Uma Thurman"]})

db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})

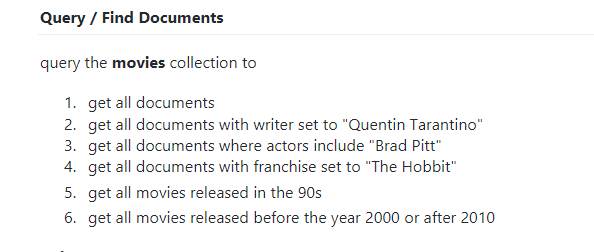
db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})

db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R Tolkien", year:"2013", franchise:"The Hobbit"})

db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."})

db.movies.insert({title:"Pee Wee Herman's Big Adventures"})

db.movies.insert({title:"Avatar"})



1.db.movies.find()

2.db.movies.find({"writer" : "Quentin Tarantino"})

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

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

5.db.movies.find({"year" : {"$gt":"1990","$lt":"2000"}})

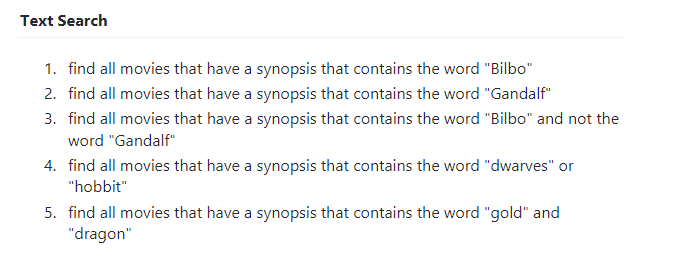
6.db.movies.find({$or:[{"year": {"$lt":"2000"}},{"year" : {"$gt":"2010"}}]})



db.movies.update({\_id:ObjectId("61054405e40120cbaaeafa63")}, {$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."}})

db.movies.update({\_id:ObjectId("610544a5e40120cbaaeafa64")}, {$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."}})

db.movies.update({\_id:ObjectId("610543cce40120cbaaeafa60")}, {$push:{actors:"Samuel L. Jackson"}})



1.db.movies.find({synopsis:{$regex:"Bilbo"}})

2.db.movies.find({synopsis:{$regex:"Gandalf"}})

3.db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})

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

5.db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})



1.db.movies.deleteOne({"title" : "Pee Wee Herman's Big Adventures"})

2.db.movies.deleteOne({"title" : "Avatar"})



Users

db.users.insert({username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

db.users.insert({username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

Posts

db.getCollection('posts').find({})

db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})

db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})

db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})

db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})

db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})

db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})

db.posts.find()

Comments

db.comments.find({})

db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("61055323e40120cbaaeafa6d")})

db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("61055330e40120cbaaeafa6e")})

db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("6105533be40120cbaaeafa6f")})

db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("610552e6e40120cbaaeafa6a")})

db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("61055301e40120cbaaeafa6c")})



1. db.users.find()
2. db.posts.find()
3. db.posts.find({"username" : "GoodGuyGreg"})
4. db.posts.find({"username" : "ScumbagSteve"})
5. db.comments.find()
6. db.comments.find({"username" : "GoodGuyGreg"})
7. db.comments.find({"username" : "ScumbagSteve"})
8. db.posts.aggregate([  
   {  
   $match: { title: 'Reports a bug in your code' }  
   },  
   {  
   $lookup: {  
   from: 'comments',  
   localField: '\_id',  
   foreignField: 'post',  
   as: 'comments'  
   }  
   }  
   ])