Insert Documents

db.movies.insert([{

title : "Fight Club",

writer : "Chuck Palahniuko",

year : 1999,

actors : [

"Brad Pitt",

"Edward Norton"

]

},

{

title : "Pulp Fiction",

writer : "Quentin Tarantino",

year : 1994,

actors : [

"John Travolta",

"Uma Thurman"

]

},

{

title : "Inglorious Basterds",

writer : "Quentin Tarantino",

year : 2009,

actors : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

},

{

title : "The Hobbit: An Unexpected Journey",

writer : "J.R.R. Tolkein",

year : 2012,

franchise : "The Hobbit"

},

{

title : "The Hobbit: The Desolation of Smaug",

writer : "J.R.R. Tolkein",

year : 2013,

franchise : "The Hobbit"

},

{

title : "The Hobbit: The Battle of the Five Armies",

writer : "J.R.R. Tolkein",

year : 2012,

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."

},

{

title : "Pee Wee Herman's Big Adventure"

},

{

title : "Avatar"

}

]

)

Query / Find Documents

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({$and:[{"year":{$gte:1900}},{"year":{$lt:2000}}]})

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

Update Documents

1. db.movies.update({"title":"The Hobbit: The Desolation of Smaug"},{$set:{"synopsis":"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})

2. 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."}})

3. db.movies.update({"title":"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})

Text Search

db.movies.createIndex({synopsis:"text"})

1. db.movies.find({$text:{$search:"Bilbo"}})

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

3. db.movies.find({$text:{$search:"Bilbo -Gandalf"}})

4. db.movies.find({$text:{$search:"dwarves hobbit"}})

5. db.movies.find({$text:{$search:"\"gold\" \"hobbit\""}})

Delete Documents

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

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

Relationships

db.users.insertMany([{

username : "GoodGuyGreg",

first\_name : "Good Guy",

last\_name : "Greg"

},

{

username : "ScumbagSteve",

full\_name :{

first : "Scumbag",

last : "Steve"

}

}])

db.posts.insertMany([{

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"

}

])

db.comments.insertMany([{

username : "GoodGuyGreg",

comment : "Hope you got a good deal!",

post : ObjectId("5f9194a3ba87315ba9a13766")

},

{

username : "GoodGuyGreg",

comment : "What's mine is yours!",

post : ObjectId("5f9194a3ba87315ba9a13767")

},

{

username : "GoodGuyGreg",

comment : "Don't violate the licensing agreement!",

post : ObjectId("5f9194a3ba87315ba9a13768")

},

{

username : "ScumbagSteve",

comment : "It still isn't clean",

post : ObjectId("5f9194a3ba87315ba9a13763")

},

{

username : "ScumbagSteve",

comment : "Denied your PR cause I found a hack",

post : ObjectId("5f9194a3ba87315ba9a13765")

}

])

Querying related collections

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.find({title:"Reports a bug in your code"},{\_id :1})

db.comments.find({post:{$in:[ObjectId("5f9194a3ba87315ba9a13765")]}})