

1Ans:

db.movies.find()

2Ans:

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

3Ans:

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

4Ans

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

5Ans

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

6Ans

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

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."

Ans:

db.movies.update({title: " The Hobbit: An Unexpected Journey "},{ $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."}})

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."

Ans:

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

Ans:

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

Text Search

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

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

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

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

3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf" db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}},{synopsis:{$not:/Gandalf/}}]})

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

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

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

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

Delete Documents

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

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

1. delete the movie "Avatar"

Ans: db.movies.remove({movie:"Avatar" })

db.createCollection("users")

db.users.insertMany([{username:"GoodGuyGreg",

first\_name:"Good Guy",

lastname:"Greg"},

{username:"ScumbagSteve",

fullname:{first:"Scumbag",last:"Steve"}}])

db.createCollection("posts")

db.posts.insertMany([{username:"GoodGuyGreg",

title:"Passes out at a 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.createCollection("comments")

db.comments.insertMany([{username:"GoodGuyGreg",

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

post:ObjectId("5ffd71b2abdff923fc6f63dc")

},

{

username:"GoodGuyGreg",

comment:"What's mine is yours",

post:ObjectId("5ffd71b2abdff923fc6f63dd")

},

{

username:"GoodGuyGreg",

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

post:ObjectId("5ffd71b2abdff923fc6f63de")

},

{

username:"ScumbagSteve",

comment:"It still isn't clean",

post:ObjectId("5ffd71b2abdff923fc6f63d9")

},

{

username:"ScumbagSteve",

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

post:ObjectId("5ffd71b2abdff923fc6f63db")

}

])

1. find all users

db.users.find().pretty()

2. find all posts

db.posts.find().pretty()

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

4. find all posts that was authored by "ScumbagStevedb.posts.find({username:"ScumbagSteve"})

5. find all comments

db.comments.find().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"

db.posts.aggregate([{$match:{title:"Reports a bug in your code"}},

{$lookup:{from:"comments",

localField:"\_id",

foreignField:"post",

as:"comments"}

}]).pretty()