**ASSIGNMENT 1**

1. > db.movies.find().pretty()

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

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

"writer" : "J.R.R Tolkien",

"year" : "2002”

1. Get all documents with writer set to "Quentin tarantino"

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

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

1. Get All the docs where actors include bradd pitt.

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

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

1. Get all the docs having fanchise as The Hobbit

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

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"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."

}

1. All movies released in 90s

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

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

1. All movies before 2000 and after 2010.

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

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

**Update Documents**

1. add synopsis

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

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

"writer" : "J.R.R Tolkien",tg

"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.update({\_id:ObjectId("5c9fa42ae5c2dfe9b3729c03")},

{$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."}})

WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })

db.movies.update

({\_id:ObjectId("5c9f983ce5c2dfe9b3729bfc")}, {$push:{actors:"Samuel L. Jackson"}})

WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })

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

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

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

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

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bibo and Company are forced to egage in a

war against an arrayof combatants and keep the Lonely Mountain from falling into

the hands of a rising darkness."

}

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

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

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"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."

}

1. 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/}}]}).pretty()

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"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."

}

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

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

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**

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

db.movies.remove({\_id:ObjectId("6106418d1506b1ec4a887c56")})

-------------------------------------------------------------------------------------------

**delete avatar**

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

WriteResult({ "nRemoved" : 1 })

-----------------------------------------------------------------------------------------

**POSTS**

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("5ca0b7e96435f98b5901f463")})

WriteResult({ "nInserted" : 1 })

db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("5ca0b9706435f98b5901f46a")})

WriteResult({ "nInserted" : 1 })

db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("5ca0b8766435f98b5901f467")})

WriteResult({ "nInserted" : 1 })

db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("5ca0b8546435f98b5901f466")})

WriteResult({ "nInserted" : 1 })

db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("5ca0b9256435f98b5901f469")})

WriteResult({ "nInserted" : 1 })

#USERS

db.users.insert({\_id:1,username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

WriteResult({ "nInserted" : 1 })

db.users.insert({\_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

WriteResult({ "nInserted" : 1 })

show collections

comments

posts

users

find all users

db.users.find().pretty()

{

"\_id" : 1,

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : 2,

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

find all posts

db.posts.find().pretty()

{

"\_id" : ObjectId("6106a9c9c53f6e517e223323"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9d1c53f6e517e223324"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9dac53f6e517e223325"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("6106a9e2c53f6e517e223326"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("6106a9e8c53f6e517e223327"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("6106a9f0c53f6e517e223328"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

Posts authored by greg

db.posts.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("6106a9c9c53f6e517e223323"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9d1c53f6e517e223324"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9dac53f6e517e223325"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

posts authored by scumnagsteve

db.posts.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("6106a9e2c53f6e517e223326"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("6106a9e8c53f6e517e223327"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("6106a9f0c53f6e517e223328"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

all comments

db.comments.find().pretty()

{

"\_id" : ObjectId("6106aa9ac53f6e517e223329"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b7e96435f98b5901f463")

}

{

"\_id" : ObjectId("6106aaabc53f6e517e22332a"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b9706435f98b5901f46a")

}

{

"\_id" : ObjectId("6106aab3c53f6e517e22332b"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b8766435f98b5901f467")

}

{

"\_id" : ObjectId("6106aabcc53f6e517e22332c"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ca0b8546435f98b5901f466")

}

{

"\_id" : ObjectId("6106aac5c53f6e517e22332d"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("5ca0b9256435f98b5901f469")

}

comments by good guy greg

db.comments.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("6106aa9ac53f6e517e223329"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b7e96435f98b5901f463")

}

{

"\_id" : ObjectId("6106aaabc53f6e517e22332a"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b9706435f98b5901f46a")

}

{

"\_id" : ObjectId("6106aab3c53f6e517e22332b"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ca0b8766435f98b5901f467")

}

comments by scumbag

db.comments.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("6106aabcc53f6e517e22332c"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ca0b8546435f98b5901f466")

}

{

"\_id" : ObjectId("6106aac5c53f6e517e22332d"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("5ca0b9256435f98b5901f469")

}

db.comments.find({\_id:ObjectId("6106a9dac53f6e517e223325")}).pretty()