Assignment: 1

. use a database named mongo\_practice: use mongo\_practice

. Insert the following documents into a movies collection.: > db.movies.insert([

... {\_id:1,title:"Fight Club",writer:"Chuck Palahniuko",year:1999,actors:["Brad Pitt","Edward Norton"]},{\_id:2,title:"Pulp Fiction",writer:"Quentin Tarantino",year:1994,actors:["John Travolta","Uma Thurman"]},{\_id:3,title:"Inglorious Basterds",writer:"Quentin Tarantino",year:2009,actors:["Brad Pitt","Diane Kruger","Eli Roth"]},{\_id:4,title:"The Hobbit:An Unexpected Journey",writer:"J.R.R. Tolkein",year:2013,franchise:"The Hobbit"},{\_id:5,title:"The Hobbit: The Desolation of Smaug",writer:"J.R.R. Tolkien",year:2013,franchise:"The Hobbit"}])

db.movies.insert([

... {\_id:6,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."},{\_id:7,title:"Pee Wee Hermen's Big Adventure"},{\_id:8,title:"Avatar"}])

query the movies collection

1. Get all documents

db.movies.find()

{ "\_id" : 1, "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : 3, "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 6, "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." }

{ "\_id" : 7, "title" : "Pee Wee Hermen's Big Adventure" }

{ "\_id" : 8, "title" : "Avatar" }

2. Get all documents with writer set to "Quentin Tarantino":

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

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : 3, "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

3. Get all documents where actors include "Brad Pitt":

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

{ "\_id" : 1, "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : 3, "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

4. Get all documents with franchise set to "The Hobbit":

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

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 6, "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." }

5. Get all movies released in the 90s:

db.movies.find({"year": {"$gte": 1900, "$lt": 2000}})

{ "\_id" : 1, "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

6. Get all movies released before the year 2000 or after 2010:

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

{ "\_id" : 1, "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : 6, "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." }

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.update({"\_id":4},{$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."}})

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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. 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.update({"\_id":5},{$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."}})

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit", "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"

db.movies.update({"\_id":2}, {$push:{actors:"Samuel L. Jackson"}})

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman", "Samuel L. Jackson" ] }

Text Search

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

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

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 6, "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." }

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

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

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit", "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." }

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/}}]})

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 6, "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." }

1. Find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

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

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit", "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." }

1. Find all movies that have a synopsis that contains the word "gold" and "dragon"

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

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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." }

Delete Documents

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

db.movies.remove({"\_id":7})

1. Delete the movie "Avatar"

db.movies.remove({"\_id":8})

{ "\_id" : 1, "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : 2, "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman", "Samuel L. Jackson" ] }

{ "\_id" : 3, "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : 4, "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 5, "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : 2013, "franchise" : "The Hobbit", "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." }

{ "\_id" : 6, "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." }

Relationships

1. Insert the following documents into a users collection

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

1. Insert the following documents into a posts collection

db.posts.insert([{"\_id":1,username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"},{"\_id":2, username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"},{"\_id":3,username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"},{"\_id":4,username:"ScumbagSteve", title:"Borrows something", body:"Sells it"},{"\_id":5,username:"ScumbagSteve", title:"Borrows everything", body:"The end"},{"\_id":6,username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"}])

1. Insert the following documents into a comments collection

db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:{"\_id":4}})

db.comments.insert([{ username:"GoodGuyGreg", comment:"What's mine is yours!!", post:{"\_id":4}},{username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:{"\_id":6}},{username:"ScumbagSteve", comment:"It still isn't clean", post:{"\_id":1}},{username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:{"\_id":3}}])

Querying related collections

1. Find all users

db.users.find()

{ "\_id" : 1, "username" : "GoodGuyGreg", "first\_name" : "Good Guy", "last\_name" : "Greg" }

{ "\_id" : 2, "username" : "ScumbagSteve", "full\_name" : { "first" : "Scumbag", "last" : "Steve" } }

> db.posts.find()

{ "\_id" : 1, "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : 2, "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : 3, "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

{ "\_id" : 4, "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : 5, "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : 6, "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

1. Find all posts

db.posts.find()

{ "\_id" : 1, "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : 2, "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : 3, "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

{ "\_id" : 4, "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : 5, "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : 6, "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

1. Find all posts that was authored by "GoodGuyGreg"

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

{ "\_id" : 1, "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : 2, "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : 3, "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

1. Find all posts that was authored by "ScumbagSteve"

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

{ "\_id" : 4, "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : 5, "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : 6, "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

1. Find all comments

db.comments.find()

{ "\_id" : ObjectId("5ffffc23ec21d798b7d41ffa"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : { "\_id" : 4 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffb"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!!", "post" : { "\_id" : 4 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffc"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : { "\_id" : 6 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffd"), "username" : "ScumbagSteve", "comment" : "It still isn't clean", "post" : { "\_id" : 1 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffe"), "username" : "ScumbagSteve", "comment" : "Denied your PR cause I found a hack", "post" : { "\_id" : 3 } }

1. Find all comments that was authored by "GoodGuyGreg"

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

{ "\_id" : ObjectId("5ffffc23ec21d798b7d41ffa"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : { "\_id" : 4 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffb"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!!", "post" : { "\_id" : 4 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffc"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : { "\_id" : 6 } }

1. Find all comments that was authored by "ScumbagSteve"

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

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffd"), "username" : "ScumbagSteve", "comment" : "It still isn't clean", "post" : { "\_id" : 1 } }

{ "\_id" : ObjectId("5ffffe12ec21d798b7d41ffe"), "username" : "ScumbagSteve", "comment" : "Denied your PR cause I found a hack", "post" : { "\_id" : 3 } }