MongoDB Lab Assignments -Day 1

MongoDB Exercise in mongo shell

Connect to a running mongo instance, use a database named mongo\_practice. Document all your queries in a javascript file to use as a reference.

Insert Documents

Insert the following documents into a movies collection.

**Show dbs**

**Use mongo\_practice**

**db.createCollection(“movies”)**

title : Fight Club writer : Chuck Palahniuko year : 1999 actors : [ Brad Pitt Edward Norton ]

**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 ]

**db.movies.insert({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 ]

**db.movies.insert({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

**db.movies.insert({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

**db.movies.insert({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.

**db.movies.insert({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

**db.movies.insert({title : “Pee Wee Herman's Big Adventure”})**

title : Avatar

**db.movies.insert({title:”Avatar”})**

Query / Find Documents

query the movies collection to

1.get all documents

**db.movies.find({})**

2. get all documents with writer set to "Quentin Tarantino"

**db.movies.find({writer:”quentin tarantino”})**

3. get all documents where actors include "Brad Pitt"

**db.movies.find({actors:”Brad pitt”})**

4. get all documents with franchise set to "The Hobbit"

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

5. get all movies released in the 90s

**db.movies.find({year:{$gte:(“1990”),$lte(“1999”)}})**

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

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

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({title:” The Hobbit: An Unexpected Journey"},{$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."}})**

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({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"

**db.movies.update({title:”pulp fiction”},{$push:{actors:”Samuel l.jaclson”}})**

Text Search

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

**db.movies.find({synopsis:/.bilbo/})**

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

**db.movies.find({synopsis:/.gandalf/})**

3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

**db.movies.find({$and:{synopsis:/bilbo/},{synopsis:{$not:/gandlf/}}]})**

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

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

**db.movies.find({$or:[{synopsis:/gold/]},{synopsis:/dragon/}]})**

Delete Documents

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

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

2. delete the movie "Avatar"

**db.movies.remove({title:”Avatar”})**

Relationships

Insert the following documents into a users collection

**db.createCollection(“user”)**

username : GoodGuyGreg first\_name : "Good Guy" last\_name : "Greg"

**db.user.insert({username: "goodguygreg", firstname:"good guy", lastname:"greg"})**

username : ScumbagSteve full\_name : first : "Scumbag" last : "Steve"

**db.user.insert((username: "scumbagsteve", firstname:"scumbag", lastname:"steve"})**

Insert the following documents into a posts collection

username : GoodGuyGreg title : Passes out at party body : Wakes up early and cleans house

**db.posts.insert({id:"1",username: "goodguygreg", title: "passes out at party", body: wakes up early and clean house"})**

username : GoodGuyGreg title : Steals your identity body : Raises your credit score

**db.posts.insert({id:"2",username:"goodguygreg", title: "steals your identity", body:"raise your credit score"})**

username : GoodGuyGreg title : Reports a bug in your code body : Sends you a Pull Request

**db.posts.insert({id:"3",username:"goodguygreg", title: "reports a bug in your code", body:"sends you a pull request"})**

username : ScumbagSteve title : Borrows something body : Sells it

**db.posts.insert({id:"4",username: "scumbagsteve", title: "borrows something",body: "sells it"})**

username : ScumbagSteve title : Borrows everything body : The end

**db.posts.insert({id:"5",username: "scumbagsteve", title: "borrows everything",body:"the end"))**

username : ScumbagSteve title : Forks your repo on github body : Sets to private

**db.posts.insert({id:"6",username: "scumbagsteve", title: "forks your repo on github",body:"sets to private"})**

Insert the following documents into a comments collection

**db.createCollection("comments")**

username : GoodGuyGreg comment : Hope you got a good deal! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows something"

**db.comments.insert({username:"goodguygreg",comment: " Hope you got a good deal!",post:"1"})**

username : GoodGuyGreg comment : What's mine is yours! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Borrows everything"

**db.comments.insert({username:"goodguygreg",comment: "whats mine is yours!",post:"2"})**

username : GoodGuyGreg comment : Don't violate the licensing agreement! post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Forks your repo on github

**db.comments.insert((username: "goodguygreg",comment:"dont violate the licensing agreement!",post:"3"])**

username : ScumbagSteve comment : It still isn't clean post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Passes out at party"

**db.comments.insert((username:"scumbagsteve",comment:"it still isnt clean",post:"4"})**

username : ScumbagSteve comment : Denied your PR cause I found a hack post : [post\_obj\_id] where [post\_obj\_id] is the ObjectId of the posts document: "Reports a bug in your code"

**db.comments.insert((username:"scumbagsteve",comment:"denied your pr cause i found a hock",post:"5"})**

Querying related collections

1.find all users

**db.user.find({})**

2. find all posts

**db.posts.find({})**

3. find all posts that was authored by "GoodGuyGreg"

**db.posts.find({username: "goodguygreg"})**

4. find all posts that was authored by "ScumbagSteve"

**db.posts.find({username: "scumbagsteve"})**

5. find all comments

**db.comments.find({})**

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.comments.find({username: "scumbagsteve"})**