**MongoDB Lab Assignments -Day 1**

**1.Insert Documents - Insert the following documents into a movies collection**.

db.movies.insert({title:"Pulp Fiction",writer:"Quentin Tarantino",year:"2009",actors:["John Travolta","Uma Thurman"]})

db.movies.insert({title:"Inglorious Basterds",writer:"Quentin Tarantino",year:"2009",actors:["Brad Pitt","Diane Kruger","Eli Roth"]})

db.movies.insert({title:"The Hobbit: An unexpected Journey",writer:"J.R.R Tolkein",year:"2012",franchise:"The Hobbit"})

db.movies.insert({title:"The Hobbit: The Desolation of Smaug",writer:"J.R.R Tolkien",year:"2013",franchise:"The Hobbit"})

db.movies.insert({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."})

db.movies.insert({title:"Pee Wee Herman's Big Adventures"})

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

**2.Query / Find Documents query the movies collection to**

1. get all documents

db.movies.find()

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

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

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

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

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

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

1. get all movies released in the 90s

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

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

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

**3. Update Documents**

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."}})

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."}})

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

**4.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"}}]})

**5.Delete Documents**

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

db.movies.remove({id:ObjectId("61eed1dd92d1e244c1fc3634")})

2. delete the movie "Avatar"

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

**6.Relationships**

I added the relationship.txt in MongoDB Directory at github

These commands we need to insert.

**7.Querying related collections**

1. find all users

db.users.find().pretty()

1. find all posts

db.posts.find().pretty()

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

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

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

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

1. find all comments

db.comments.find().pretty()

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

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

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

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