**Assignment 01**

**Create Database :**

> use mongo\_practice

switched to db mongo\_practice

**Create Collection :**

> db.createCollection("movies")

{ "ok" : 1 }

**Insert Documents :**

> db.movies.insertMany([

... {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"] } ,

... { title:"Inglorious Basterds" , writer:"Quentin Tarantion" , 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"} ,

... {title:"The Hobbit:The Desolation of Smaug",writer:"J.R.R.Tolkein",year:"2013",franchise:"The Hobbit"} ,

... {title:"The Hobbit:The Battle of 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"}, {title:"Avatar"}])

**Query/Find Documents :**

* To get all documents

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

* To get all documents with writer set to “Quentin Tarantino”

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

* To get all documents where actors include “Brad Pitt”

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

* To get all the documents with franchise set to “The Hobbit”

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

* To get all the movies released in the 90s

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

* To get all the movies released before the year 2000 or after 2010

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

**Update Documents :**

* Add synopsis to “The Hobbit:An unexpected Journey”

> 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."}})

* Add synopsis to “The Hobbit:The Desolation of 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."}})

* Add an actor named “Samuel L. Jackson” to movie “Pulp Fiction”

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

**Text Search :**

**Create index :**

> db.movies.createIndex({synopsis:"text"})

* Find all movies that have synopsis that contains the word “Bilbo”

>db.Movies.find({$text:{$search:"Bilbo"}}).pretty()

* Find all movies that have synopsis that contains the word “Gandalf”

> db.Movies.find({$text:{$search:"Gandalf"}}).pretty()

* Find all movies that have synopsis that contains the word “Bilbo” and not “Gandalf”

> db.Movies.find({$text:{$search:"Bilbo -Gandalf"}}).pretty()

* Find all movies that have synopsis that contains the word “dwarves” or “hobbit”

> db.Movies.find({$text:{$search:"dwarves hobbit"}}).pretty()

* Find all movies that have synopsis that contains the word “gold” and “dragon”

> db.Movies.find({$text:{$search:"\"gold\"\"dragon\""}}).pretty()

**Delete Documents :**

* Delete movie “Pee Wee Herman’s Big adventure”

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

* Delete movie “Avatar”

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

**Relationships :**

**Create collection – users :**

> db.createCollection("users")

**Insert documents – users :**

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

**Create Collection – posts:**

> db.createCollection("posts")

**Insert documents – posts :**

> db.posts.insertMany([

... {username:"GoodGuyGreg" , title:"Passes out at 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"} ])

**Create Collection – comments :**

> db.createCollection("comments")

**Insert documents – comments :**

> db.comments.insertMany([

... {username:"GoodGuyGreg" , comment:"Hope you got a good deal!" , post:"5f8de89ed5f79d5a976511d0"},

... {username:"GoodGuyGreg" , comment:"What's mine is yours!" , post:"5f8de89ed5f79d5a976511d1"},

... {username:"GoodGuyGreg" , comment:"Don't violate the licensing agreement!" , post:"5f8de89ed5f79d5a976511d2"},

... {username:"ScumbagSteve" , comment:"It still isn't clean" , post:"5f8de89ed5f79d5a976511cd"},

... {username:"ScumbagSteve" , comment:"Denied your PR cause I found a hack" , post:"5f8de89ed5f79d5a976511cf"}])

* Find all users

> db.users.find().pretty()

* Find all posts

> db.posts.find().pretty()

* Find all posts that was authored by “GoodGuyGreg”

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

* Find all posts that was authored by “ScumbagSteve”

> db.posts.find({username:“ScumbagSteve”}).pretty()

* Find all comments

> db.comments.find().pretty()

* Find all comments that was authored by “GoodGuyGreg”

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

* Find all comments that was authored by “ScumbagSteve”

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

* Find all comments that was authored by “Reports a bug in your code”

> db.comments.aggregate([{$match:{post:"5f8de89ed5f79d5a976511cf"}}]).pretty()