**Basic Assignment**

**Q1 Insert the following documents into a movies collection.**

**ANS** use mongo\_practice

db.createCollection(“movies”)(optional)

db.movies.insertMany([{title: “Fight Club”, writer: “Chuck Palahniuk”, year: “1999”, actors:[“Brad Pitt”, “Edward Norton”]},

{title: “Pulp Fiction”, writer: “Quentin Tarantino”, year: “2009”, actors: [“John Travolta”, “Uma Thurman”]},

{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”},

{title: “The Hobbit: The Desolation of Smaug”, writer: “J.R.R Tolkien”, year:“2013”, franchise: “The Hobbit”},

{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.”},

{title: “Pee Wee Herman's Big Adventures”},

{title: “Avatar”}])

**get all documents**

1. db.movies.find()

**get all documents with writer set to "Quentin Tarantino"**

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

**get all documents where actors include "Brad Pitt"**

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

**get all documents with franchise set to "The Hobbit"**

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

**get all movies released in the 90s**

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

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

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

**UPDATE DOCUMENTS**

**Q1 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."**

**Ans** db.movies.update({\_id: ObjectId("61efda6e6342ad3c96fc0fcb")}, {$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."}})

**Q2 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."**

**Ans** db.movies.update({\_id: ObjectId("61efda6e6342ad3c96fc0fca")}, {$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."}})

**Q3 Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"**

**Ans** db.movies.update({\_id: ObjectId("61efda6e6342ad3c96fc0fc9")}, {$push:{actors: "Samuel L. Jackson"}})

**TEXT SEARCH**

1. db.movies.find({synopsis:/Bilbo./}).pretty()
2. db.movies.find({synopsis:/.Gandalf./}})
3. db.movies.find({$and:[{synopsis:/Bilbo/},{synopsis:{$not:/Gandalf/}}]}).pretty()
4. db.movies.find({$or:[{synopsis:/dwarves/},{synopsis:/habbit/}]}).pretty()
5. db.movies.find({$and:[{synopsis:/gold/},{synopsis:/dragon/}]}).pretty()

**DELETE**

**Q1 Delete the movie "Pee Wee Herman's Big Adventure"**

**Ans** db.movies.remove({\_id: ObjectId("61efda6e6342ad3c96fc0fc7")})

**Q2 Delete the movie "Avatar"**

db.movies.remove({\_id: ObjectId("61efda6e6342ad3c96fc0fc8")})

**RELATIONS**

**Q1 Insert the following documents into a users collection**

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

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

**POSTS Collection Insertion**

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

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

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

db.posts.insert({ username:"ScumbagSteve", title: "Borrows something", body:"Sells it"})

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

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

**Insertion in COMMENTS Collection**

**Ans** db.comments.insert({ username: "GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("61efda6e6342ad3c96fc0fc2")})

db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("61efda6e6342ad3c96fc0fc5")})

db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("61efda6e6342ad3c96fc0fc6")})

db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("61efda6e6342ad3c96fc0fc4")})

db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("61efda6e6342ad3c96fc0fc5")})

**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 belonging to the post "Reports a bug in your code"**

db.comments.find({post:db.posts.findOne({title:”Reports a bug in your code”}).\_id})