**MongoDB Exercise in mongo shell**

**Create database- use mongo\_practice.**

**Collection Name- movies.**

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 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. 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 darkenss."**

**}])**

1. **Query / Find Documents**
2. get all documents-

=>db.movies.find()

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

=> db.movies.find({writer: “Quentin Tarantino”}).pretty()

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

=>db.movies.find({actor:”brad pitt”}).pretty()

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

=>db.movies.find({franchise:”The hobbit”}).pretty()

1. get all movies released in the 90s -

=>db.movies.find({$and:[{year:{$gt:1990}},{year:{$lt:2000}}]}).pretty()

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

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

1. **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"},{$set:{actor:["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/i}}).pretty()

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

=>db.movies.find({synopsis:{$regex:/Gandalf/i}}).pretty()

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

=>

4. find all movies that have a synopsis that contains the word "dwarves" or

"hobbit"

=>db.movies.find({$or:[{synopsis:{$regex:/dwarver/i},synopsis:{$regex:/hobbit/i}}]}).pretty()

5. find all movies that have a synopsis that contains the word "gold" and

"dragon" -

=>db.movies.find({$or:[{synopsis:{$regex:/gold/i},synopsis:{$regex:/dragon/i}}]}).pretty()

**Delete Documents**

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

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

1. delete the movie "Avatar"-

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

**Relationships**

1. Insert the following documents into a **users** collection -

=> db.users.insertOne({username:"GoodGuyGreg",first\_name:"Good Guy",last\_name:"Greg",username:"Scumbagsteve",fullname:{first:"Scumbag",last:"steve"}})

**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()

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”})

1. 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}).pretty()