**Basic Assignment**

Name: Ch Rajesh

Gmail-ID: chagantipati.rajesh09@gmail.com

use mongo\_practice

db.createCollection('movies')

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","EliRouth"]}, {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. Tokein",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

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:{$lt:2000}})

1. 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."}})

1. 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. Jackson"}})

# Text Search

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

db.movies.find({$text:{$search:'Bilbo'}},{'title':1})

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

db.movies.find({$text:{$search:'Gandalf'}},{'title':1})

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

db.movies.find({$text:{$search:'Bilbo -Gandalf'}},{'title':1})

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

db.movies.find({$text:{$search:'dwarves hobbit'}},{'title':1})

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

db.movies.find({$text:{$search:"\"gold\"\"dragon\""}},{'title':1})

# 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:"Avatar"})

db.users.insertMany([{username: 'GoodGuyGreg', 'first name':'Good Guy','last name':'Greg'},{username:'ScumbagSteve',fullname:{first:'Scumbag',last:'Steve'}}])

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:'GoodGuyGerg',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'}])

db.comments.insert([{username:'GoodGuyGreg',comment:'Hope you got a good deal!',post:ObjectId("61efaf80e9bc21d2e3949190")},{username:'GoodGuyGerg',comment:'Whats mine is yours',post:ObjectId("61efaf80e9bc21d2e3949191")},{username:'GoodGuyGreg',comment:'Dont violate the licensing agreement!',post:ObjectId("61efaf80e9bc21d2e3949192")},{username:'ScumbagSteve',comment:'It still isnt clean',post:ObjectId("61efaf80e9bc21d2e394918d")},{username:'ScumbagSteve',comment:'Denied your PR cause I found a hack',post:ObjectId("61efaf80e9bc21d2e394918f")}])

# Querying related collections

1. find all users

db.users.find()

1. find all posts

db.users.find()

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

mongo\_practice> 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:ObjectId("61efaf80e9bc21d2e394918f")})