**Assignment 1 (MongoDB)**

**Queries to 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 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 te hands of a rising darkness."

},

{ title: "Pee Wee Herman's Big Adventure"},

{title: "Avatar"}

])

**Query/Find documents**

1. **Get all documents:**

db.getCollection('movies').find({})

1. **Get all documents with writer set to “Quentin Tarantino”:**

db.movies.aggregate( [

{$match: { writer: "Quentin Tarantino"}}

])

1. **Get all documents where actors include “Brad Pitt”**

db.movies.aggregate( [

{$match: { actors: "Brad Pitt"}}

])

1. **Get all documents with franchise set to “The Hobbit”**

db.movies.aggregate( [

{$match: { franchise: "The Hobbit"}}

])

1. **Get all the movies released in the 90s**

db.movies.aggregate( [

{$match:

{$and: [{ year: {$gt:1989}}, {year: {$lt:2000}}}

}

])

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

db.movies.aggregate( [

{$match:

{$and: [{ year: {$gt:2010}}, {year: {$lt:2000}}}

}

])

**Update Documents**

1. 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"} })

1. 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."} })

1. Add an actor name to the movie pulp fiction

db.movies.update({title: "Pulp Fiction"},

{$push: {actors: "Samuel L. Jackson"}})

**Query**

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

1. db.movies.find({$text: {$search: "Bilbo"}})
2. db.movies.find({$text: {$search: "Gandalf"}})
3. db.movies.find({$text: {$search: "Bilbo -Gandalf"}})
4. db.movies.find({$text: {$search: "dwarves hobbit"}})
5. db.movies.find({$text: {$search: "\"gold dragon\”"}})

**Delete Documents**

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

**Insert documents query**

**Users collection**

db.users.insertMany([

{

username: "GoodGuyGreg",

first\_name: "Good Guy",

last\_name: "Greg"

},

{

username: "ScumbagSteve",

full\_name:

{

first: "Scumbag",

last: "Steve"

}

}

])

**Posts collection**

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"

}

])

**Comments collection**

db.comments.insertMany([

{

username: "GoodGuyGreg",

comment: "Hope you got a good deal!",

post: "5f8db8b3620fceb37960306b"

},

{

username: "GoodGuyGreg",

comment: "What's mine is yours!",

post: "5f8db8b3620fceb37960306c"

},

{

username: "GoodGuyGreg",

comment: "Don't violate the licensing agreement!",

post: "5f8db8b3620fceb37960306d"

},

{

username: "ScumbagSteve",

comment: "It still isn't clean",

post: "5f8db8b3620fceb379603068"

},

{

username: "ScumbagSteve",

comment: "Denied your PR cause I found a hack",

post: "5f8db8b3620fceb37960306a"

}

])

**Querying related collections**

1. db.getCollection('users').find({})
2. db.getCollection('posts').find({})
3. db.posts.aggregate([{$match: {username: "GoodGuyGreg"}}])
4. db.posts.aggregate([{$match: {username: "ScumbagSteve"}}])
5. db.getCollection('comments').find({})
6. db.comments.aggregate([{$match: {username: "GoodGuyGreg"}}])
7. db.comments.aggregate([{$match: {username: " ScumbagSteve "}}])
8. db.comments.aggregate([{$match: {post: "5f8db8b3620fceb37960306a"}}])