**Mongo DB Assignment – 1**

**Query/find Documents:**

1. To get all the documents:

db.movies.find()

1. To get all the documents with writer set to “Quentin Tarantino”:

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

1. To get all the documents where actors include “Brad Pitt”:

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

1. To get all the documents with franchise set to “The Hobbit”:

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

1. To get all movies released in the 90s:

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

1. To get movies released before the year 2000 or after 2010:

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

**Text Search:**

1. Find all movies that have a synopsis that contains the word “Bilbo”.

* db.movies.getIndexes()
* db.movies.createIndex({synopsis:"text"})
* db.movies.getIndexes()
* db.movies.find({$text:{$search:"Bilbo"}})

or

* db.movies.find({synopsis:/Bilbo/g})

1. Find all movies that have a synopsis that contains the word “Gandalf”.

db.movies.find({$text:{$search:"Gandalf"}})

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

db.movies.find({$text:{$search:"bilbo -gandlf"}})

1. Find all movies that have a synopsis that contains the word “dwarves” or “hobbit”.

db.movies.find({$text:{$search:"dwarves hobbit"}})

1. Find all movies that have a synopsis that contains the word “gold” and “dragon”.

db.movies.find({$text:{$search:"gold +dragon"}})

**Delete Documents:**

1. Delete the movie “Pee Wee Herman’s big Adventure”

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

1. Delete the movie “Avatar”

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

**Relationships:**

Inserting documents into the users collection:

* db.users.insertOne({username:"GoodGuyGrey",first\_name:"GoodGuy",last\_name:"Grey"})
* db.users.insertOne({username:"ScumbagSteve",full\_name:[{first:"Scumbag",last:"Steve"}]})
* db.users.find()

[

{

\_id: ObjectId("617e5426c4057dd5ea3716d3"),

username: 'GoodGuyGrey',

first\_name: 'GoodGuy',

last\_name: 'Grey'

},

{

\_id: ObjectId("617e54a8c4057dd5ea3716d4"),

username: 'ScumbagSteve',

full\_name: [ { first: 'Scumbag', last: 'Steve' } ]

}

]

Inserting documents into the posts collection:

* db.posts.insertMany([{username:"GoodGuyGrey",titletitle:"Passes out at party",body:"Wakes up early and cleans house"},{username:"GoodGuyGrey",title:"Steals your identity",body:"Raises your credit score"},{username:"GoodGuyGrey",title:"Reports a bug in your code",body:"Sends you a Pull Request"}])
* db.posts.insertMany([{username:"ScumbagSteve",titletitle:"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.posts.find()

[

{

\_id: ObjectId("617e58c5c4057dd5ea3716de"),

username: 'GoodGuyGrey',

titletitle: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("617e58c5c4057dd5ea3716df"),

username: 'GoodGuyGrey',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("617e58c5c4057dd5ea3716e0"),

username: 'GoodGuyGrey',

title: 'Reports a bug in your code',

body: 'Sends you a Pull Request'

},

{

\_id: ObjectId("617e58e0c4057dd5ea3716e1"),

username: 'ScumbagSteve',

titletitle: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("617e58e0c4057dd5ea3716e2"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("617e58e0c4057dd5ea3716e3"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

Inserting documents into the comments collection:

* db.comments.insertOne({username:"GoodGuyGrey",comment:"Hope you got a good deal!",post: ObjectId("617e58e0c4057dd5ea3716e1")})
* db.comments.insertOne({username:"GoodGuyGrey",comment:"What mine is yours",post: ObjectId("617e58e0c4057dd5ea3716e2")})
* db.comments.insertOne({username:"GoodGuyGrey",comment:"Don't voilate the licensing agreement!",post: ObjectId("617e58e0c4057dd5ea3716e3")})
* db.comments.insertOne({username:"ScumbagSteve",comment:"It still isn't clean",post: ObjectId("617e58c5c4057dd5ea3716de")})
* db.comments.insertOne({username:"ScumbagSteve",comment:"Denied your PR cause I found a hack",post: ObjectId("617e58c5c4057dd5ea3716e0")})
* db.comments.find()

[

{

\_id: ObjectId("617e59c0c4057dd5ea3716e5"),

username: 'GoodGuyGrey',

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

post: ObjectId("617e58e0c4057dd5ea3716e1")

},

{

\_id: ObjectId("617e5bcbc4057dd5ea3716e6"),

username: 'GoodGuyGrey',

comment: 'What mine is yours',

post: ObjectId("617e58e0c4057dd5ea3716e2")

},

{

\_id: ObjectId("617e5e18c4057dd5ea3716e7"),

username: 'GoodGuyGrey',

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

post: ObjectId("617e58e0c4057dd5ea3716e3")

},

{

\_id: ObjectId("617e5e68c4057dd5ea3716e8"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("617e58c5c4057dd5ea3716de")

},

{

\_id: ObjectId("617e5eb0c4057dd5ea3716e9"),

username: 'ScumbagSteve',

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

post: ObjectId("617e58c5c4057dd5ea3716e0")

}

]

**Querying related collections:**

1. Find all users

db.users.find()

1. Find all posts

db.posts.find()

1. Find all posts that was authored by “GoodGuyGreg”

db.posts.find({username:"GoodGuyGrey"})

[

{

\_id: ObjectId("617e58c5c4057dd5ea3716de"),

username: 'GoodGuyGrey',

titletitle: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("617e58c5c4057dd5ea3716df"),

username: 'GoodGuyGrey',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("617e58c5c4057dd5ea3716e0"),

username: 'GoodGuyGrey',

title: 'Reports a bug in your code',

body: 'Sends you a Pull Request'

}

]

1. Find all posts that was authored by “ScumbagSteve”

db.posts.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("617e58e0c4057dd5ea3716e1"),

username: 'ScumbagSteve',

titletitle: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("617e58e0c4057dd5ea3716e2"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("617e58e0c4057dd5ea3716e3"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

1. Find all comments

db.comments.find()

1. Find all comments that was authored by “GoodGuyGreg”

db.comments.find({username:"GoodGuyGrey"})

[

{

\_id: ObjectId("617e59c0c4057dd5ea3716e5"),

username: 'GoodGuyGrey',

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

post: ObjectId("617e58e0c4057dd5ea3716e1")

},

{

\_id: ObjectId("617e5bcbc4057dd5ea3716e6"),

username: 'GoodGuyGrey',

comment: 'What mine is yours',

post: ObjectId("617e58e0c4057dd5ea3716e2")

},

{

\_id: ObjectId("617e5e18c4057dd5ea3716e7"),

username: 'GoodGuyGrey',

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

post: ObjectId("617e58e0c4057dd5ea3716e3")

}

]

1. Find all comments that was authored by “ScumbagSteve”

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

[

{

\_id: ObjectId("617e5e68c4057dd5ea3716e8"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("617e58c5c4057dd5ea3716de")

},

{

\_id: ObjectId("617e5eb0c4057dd5ea3716e9"),

username: 'ScumbagSteve',

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

post: ObjectId("617e58c5c4057dd5ea3716e0")

}

]

1. Find all comments belonging to the post “Reports a bug in your code”

db.comments.find({post:ObjectId( '617d85eaeec2e86abd6a8056')})