**MONGO DB Basic Assignment** by K Jyothi Gangarsha Id: 259921

**MongoDB Exercise in mongo shell**

**Connect to a running mongo instance, use a database named mongo\_practice.**

>mongo

>use mongo\_practice

>show dbs;

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

users 0.000GB

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

**> db.createCollection("movies")**

{ "ok" : 1 }

> 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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness."

... },

... {

... title:"Pee Wee Herman's Big Adventure"

...

... },

... {

... title:"Avatar"

... }

... ]

... )

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60011c2e09dd7764467eb226"),

ObjectId("60011c2e09dd7764467eb227"),

ObjectId("60011c2e09dd7764467eb228"),

ObjectId("60011c2e09dd7764467eb229"),

ObjectId("60011c2e09dd7764467eb22a"),

ObjectId("60011c2e09dd7764467eb22b"),

ObjectId("60011c2e09dd7764467eb22c"),

ObjectId("60011c2e09dd7764467eb22d")

]

}

**Query / Find Documents**

**query the movies collection to**

1. **get all documents**

**> db.movies.find()**

{ "\_id" : ObjectId("60011c2e09dd7764467eb226"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb227"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb228"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22b"), "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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22c"), "title" : "Pee Wee Herman's Big Adventure" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22d"), "title" : "Avatar" }

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

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

{ "\_id" : ObjectId("60011c2e09dd7764467eb227"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb228"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

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

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

{ "\_id" : ObjectId("60011c2e09dd7764467eb226"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb228"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

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

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

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22b"), "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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

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

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

{ "\_id" : ObjectId("60011c2e09dd7764467eb226"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb227"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "John Travolta", "Uma Thurman" ] }

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

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

{ "\_id" : ObjectId("60011c2e09dd7764467eb226"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb227"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22b"), "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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

**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 ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

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."}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

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

> db.movies.update({title:"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**Text Search**

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

**> db.movies.find({synopsis:{$regex:"Bilbo"}})**

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit,Bilbo Baggins, sets out to the Lonely Mountain with a spirited group ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit", "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." }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22b"), "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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

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

**> db.movies.find({synopsis:{$regex:"Gandalf"}})**

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit", "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. **find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"**

**> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}},{synopsis:{$not:/Gandalf/}}]})**

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit,Bilbo Baggins, sets out to the Lonely Mountain with a spirited group ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22b"), "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 ofcombatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

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

**>db.movies.find({$or:[{synopsis:{$regex:"hobbit"}},{synopsis:{$regex:"dwarves"}}]})**

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit,Bilbo Baggins, sets out to the Lonely Mountain with a spirited group ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60011c2e09dd7764467eb22a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : "2013", "franchise" : "The Hobbit", "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. **find all movies that have a synopsis that contains the word "gold" and "dragon"**

**>db.movies.find({$and:[{synopsis:{$regex:"gold"}},{synopsis:{$regex:"dragon"}}]})**

{ "\_id" : ObjectId("60011c2e09dd7764467eb229"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit,Bilbo Baggins, sets out to the Lonely Mountain with a spirited group ofdwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

**Delete Documents**

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

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

**WriteResult({ "nRemoved" : 1 })**

1. **delete the movie "Avatar"**

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

**WriteResult({ "nRemoved" : 1 })**

**Relationships**

**Insert the following documents into users collection**

> db.createCollection("users")

{ "ok" : 1 }

> db.users.insertMany([{username:"GoodGuyGreg",

... first\_name:"Good Guy",

... lastname:"Greg"},

... {username:"ScumbagSteve",

... fullname:{first:"Scumbag",last:"Steve"}}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60013b5309dd7764467eb22e"),

ObjectId("60013b5309dd7764467eb22f")

]

}

**Insert the following documents into a posts collection**

> db.createCollection("posts")

{ "ok" : 1 }

> db.posts.insertMany([{username:"GoodGuyGreg",

... title:"Passes out at a 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"

... }

...

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60013d7409dd7764467eb230"),

ObjectId("60013d7409dd7764467eb231"),

ObjectId("60013d7409dd7764467eb232"),

ObjectId("60013d7409dd7764467eb233"),

ObjectId("60013d7409dd7764467eb234"),

ObjectId("60013d7409dd7764467eb235")

]

}

**Insert the following documents into a comments collection**

> db.createCollection("comments")

{ "ok" : 1 }

> db.comments.insertMany([{username:"GoodGuyGreg",

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

... post:ObjectId("5ffd71b2abdee923fc6e63dc")

... },

... {

... username:"GoodGuyGreg",

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

... post:ObjectId("5ffd71b2abdee923fc6e63dd")

... },

... {

... username:"GoodGuyGreg",

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

... post:ObjectId("5ffd71b2abdee923fc6e63de")

... },

... {

... username:"ScumbagSteve",

... comment:"It still isn't clean",

... post:ObjectId("5ffd71b2abdee923fc6e63d9")

...

... },

... {

... username:"ScumbagSteve",

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

... post:ObjectId("5ffd71b2abdee923fc6e63db")

... }

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60013e0509dd7764467eb236"),

ObjectId("60013e0509dd7764467eb237"),

ObjectId("60013e0509dd7764467eb238"),

ObjectId("60013e0509dd7764467eb239"),

ObjectId("60013e0509dd7764467eb23a")

]

}

**Querying related collections**

1. **find all users**

**> db.users.find().pretty()**

{

"\_id" : ObjectId("60013b5309dd7764467eb22e"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"lastname" : "Greg"

}

{

"\_id" : ObjectId("60013b5309dd7764467eb22f"),

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

1. **find all posts**

**> db.posts.find().pretty()**

{

"\_id" : ObjectId("60013d7409dd7764467eb230"),

"username" : "GoodGuyGreg",

"title" : "Passes out at a party",

"body" : "Wakes up early and cleans house"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb231"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb232"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb233"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb234"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb235"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

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

**> db.posts.find({username:"GoodGuyGreg"}).pretty()**

{

"\_id" : ObjectId("60013d7409dd7764467eb230"),

"username" : "GoodGuyGreg",

"title" : "Passes out at a party",

"body" : "Wakes up early and cleans house"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb231"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb232"),

"username" : "GoodGuyGreg",

"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"}).pretty()**

{

"\_id" : ObjectId("60013d7409dd7764467eb233"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb234"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("60013d7409dd7764467eb235"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

1. **find all comments**

**> db.comments.find().pretty()**

{

"\_id" : ObjectId("60013e0509dd7764467eb236"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63dc")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb237"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63dd")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb238"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63de")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb239"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ffd71b2abdee923fc6e63d9")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb23a"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63db")

}

1. **find all comments that was authored by "GoodGuyGreg"**

**> db.comments.find({username:"GoodGuyGreg"}).pretty()**

{

"\_id" : ObjectId("60013e0509dd7764467eb236"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63dc")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb237"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63dd")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb238"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63de")

}

1. **find all comments that was authored by "ScumbagSteve"**

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

{

"\_id" : ObjectId("60013e0509dd7764467eb239"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ffd71b2abdee923fc6e63d9")

}

{

"\_id" : ObjectId("60013e0509dd7764467eb23a"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("5ffd71b2abdee923fc6e63db")

}

1. **find all comments belonging to the post "Reports a bug in your code"**

> db.posts.aggregate([{$match:{title:"Reports a bug in your code"}},

... {$lookup:{from:"comments",

... localField:"\_id",

... foreignField:"post",

... as:"comments"}

...

... }]).pretty()

{

"\_id" : ObjectId("60013d7409dd7764467eb232"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request",

"comments" : [ ]

}

**---THANKYOU---**