**Name: Lutika Kolhe**

# MongoDB Lab Assignments -Day 1

**MongoDB Exercise in mongo shell**

Connect to a running mongo instance, use a database named **mongo\_practice**. Document all your queries in a javascript file to use as a reference.

Ans- Microsoft Windows [Version 10.0.19042.1415]

(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Downloads\mongosh-1.1.7-win32-x64\bin>mongosh "mongodb+srv://cluster0.k9kgm.mongodb.net/myFirstDatabase" --username Lutika

Enter password: \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Current Mongosh Log ID: 61d83989bcc4c9d324b47d81

Connecting to: mongodb+srv://cluster0.k9kgm.mongodb.net/myFirstDatabase

Using MongoDB: 4.4.10

Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-ojy0k0-shard-0 [primary] first-test> use mongodb\_practice

switched to db mongodb\_practice

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> use mongodb\_practice

already on db mongodb\_practice

# Insert Documents

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] > mongodb\_practice> db.createCollection('movies')

{ ok: 1 }

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> show collections

movies

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 darkness.

title : Pee Wee Herman's Big Adventure title : Avatar

Reference https[://ww](http://www.tutorialspoint.com/mongodb/mongodb_insert_document.htm)w.t[utorialspoint.com/mongodb/mongodb\_insert\_document.htm](http://www.tutorialspoint.com/mongodb/mongodb_insert_document.htm)

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.insertMany([{title:'Fight Club', writer: 'Chuck PalahniAtlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.insertMany([{title:'Fight Club', writer: 'Chuck PalahniAtlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.insertMany([{title:'Fight Club', writer: 'Chuck Palahniuko', year:1999, actors:['Brad Pitt','Edwrd 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 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'}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("61daaa6efe12c3f0cacd55ec"),

'1': ObjectId("61daaa6efe12c3f0cacd55ed"),

'2': ObjectId("61daaa6efe12c3f0cacd55ee"),

'3': ObjectId("61daaa6efe12c3f0cacd55ef"),

'4': ObjectId("61daaa6efe12c3f0cacd55f0"),

'5': ObjectId("61daaa6efe12c3f0cacd55f1"),

'6': ObjectId("61daaa6efe12c3f0cacd55f2"),

'7': ObjectId("61daaa6efe12c3f0cacd55f3")

}

}

# Query / Find Documents

query the **movies** collection to

1. get all documents

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find()

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ec"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: 1999,

actors: [ 'Brad Pitt', 'Edwrd Norton' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ed"),

title: 'Pulp Fiction',

writer: 'Quentin Tarantino',

year: 1994,

actors: [ 'John Travolta', 'Uma Thurman' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ee"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

title: 'The Hobbit: An Unexpected Journey',

writer: 'J.R.R Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R Tolkein',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f1"),

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 array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f2"),

title: "Pee Wee Herman's Big Adventure"

},

{ \_id: ObjectId("61daaa6efe12c3f0cacd55f3"), title: 'Avatar' }

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({writer:"Quentin Tarantino"})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ed"),

title: 'Pulp Fiction',

writer: 'Quentin Tarantino',

year: 1994,

actors: [ 'John Travolta', 'Uma Thurman' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ee"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

}

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({actors:"Brad Pitt"})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ec"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: 1999,

actors: [ 'Brad Pitt', 'Edwrd Norton' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ee"),

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"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({franchise:"The Hobbit"})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

title: 'The Hobbit: An Unexpected Journey',

writer: 'J.R.R Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R Tolkein',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f1"),

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 array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness'

}

]

1. get all movies released in the 90s

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({year:{$gt:1990, $lt:2000}})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ec"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: 1999,

actors: [ 'Brad Pitt', 'Edwrd Norton' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ed"),

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({$or:[{year:{$gt:2010}},{year: {$lt:2000}}]})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ec"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: 1999,

actors: [ 'Brad Pitt', 'Edwrd Norton' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ed"),

title: 'Pulp Fiction',

writer: 'Quentin Tarantino',

year: 1994,

actors: [ 'John Travolta', 'Uma Thurman' ]

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

title: 'The Hobbit: An Unexpected Journey',

writer: 'J.R.R Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R Tolkein',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f1"),

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 array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness'

}

]

Reference: https[://ww](http://www.tutorialspoint.com/mongodb/mongodb_query_document.htm)w.t[utorialspoint.com/mongodb/mongodb\_query\_document.htm](http://www.tutorialspoint.com/mongodb/mongodb_query_document.htm)

# 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."

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> 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."}})

DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

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."

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> 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."}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.update({title:'Pulp Fiction'}, {$push:{actors:"Samuel L. Jackson"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

Reference: https[://ww](http://www.tutorialspoint.com/mongodb/mongodb_update_document.htm)w.t[utorialspoint.com/mongodb/mongodb\_update\_document.htm](http://www.tutorialspoint.com/mongodb/mongodb_update_document.htm)

# Text Search

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({synopsis:{$regex:"Bilbo"}})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

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 of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

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("61daaa6efe12c3f0cacd55f1"),

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 array of combatants 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"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({synopsis:{$regex:"Gandalf"}})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

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"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$no$not:/Gandalf/}}]})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

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 of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f1"),

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 array of combatants 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"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

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 of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'

},

{

\_id: ObjectId("61daaa6efe12c3f0cacd55f0"),

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"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})

[

{

\_id: ObjectId("61daaa6efe12c3f0cacd55ef"),

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 of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'

}

]

Reference: https[://w](http://www.tutorialspoint.com/mongodb/mongodb_text_search.htm)ww[.t](http://www.tutorialspoint.com/mongodb/mongodb_text_search.htm)u[torialspoint.com/mongodb/mongodb\_text\_search.htm](http://www.tutorialspoint.com/mongodb/mongodb_text_search.htm)

# Delete Documents

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.remove({title:"Pee Wee Herman's Big Adventure"})

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.

{ acknowledged: true, deletedCount: 1 }

1. delete the movie "Avatar"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.movies.remove({title:"Avatar"})

{ acknowledged: true, deletedCount: 1 }

Reference: https[://ww](http://www.tutorialspoint.com/mongodb/mongodb_delete_document.htm)w.t[utorialspoint.com/mongodb/mongodb\_delete\_document.htm](http://www.tutorialspoint.com/mongodb/mongodb_delete_document.htm)

# Relationships

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

username : GoodGuyGreg first\_name : "Good Guy" last\_name : "Greg" username : ScumbagSteve full\_name :

first : "Scumbag" last : "Steve"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.createCollection('users')

{ ok: 1 }

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.users.insertMany([{ username: 'GoodGuyGreg', first\_name: 'GoodGuy', last\_name: 'Greg' },{ username: 'ScumbagSteve', fullname: { first: 'Scumbag', last: 'Steve' }}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("61dab4b3fe12c3f0cacd55f4"),

'1': ObjectId("61dab4b3fe12c3f0cacd55f5")

}

}

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

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

username : GoodGuyGreg title : Passes out at party

body : Wakes up early and cleans house

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.createCollection('posts')

{ ok: 1 }

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> 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 in github', body: 'Sets to private'}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("61dab5e9fe12c3f0cacd55f6"),

'1': ObjectId("61dab5e9fe12c3f0cacd55f7"),

'2': ObjectId("61dab5e9fe12c3f0cacd55f8"),

'3': ObjectId("61dab5e9fe12c3f0cacd55f9"),

'4': ObjectId("61dab5e9fe12c3f0cacd55fa"),

'5': ObjectId("61dab5e9fe12c3f0cacd55fb")

}

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

username : GoodGuyGreg

comment : Hope you got a good deal! post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the posts document: "Borrows something"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.createCollection('comments')

{ ok: 1 }

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.insert({username:'GoodGuyGreg', comment: 'Hope you got a good deal!', post:ObjectId("61da896cf1e258629fdd6f5f")})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

acknowledged: true,

insertedIds: { '0': ObjectId("61dab6a5fe12c3f0cacd55fc") }

}

username : GoodGuyGreg comment : What's mine is yours! post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the posts document: "Borrows everything"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.insert({username:'GoodGuyGreg', comment: 'Hope you got a good deal!', post:ObjectId("61da896cf1e258629fdd6f5f")})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

acknowledged: true,

insertedIds: { '0': ObjectId("61dab6a5fe12c3f0cacd55fc") }

}

username : GoodGuyGreg

comment : Don't violate the licensing agreement! post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the posts document: "Forks your repo on github

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.insert(

... {username: 'GoodGuyGreg', comment: "What's mine is yours!", post:ObjectId("61da896cf1e258629fdd6f60")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("61dab6defe12c3f0cacd55fd") }

}

username : ScumbagSteve comment : It still isn't clean post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the posts document: "Passes out at party"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.insert({username: 'ScumbagSteve', comment: "It still isn't clean", post:ObjectId("61da896cf1e258629fdd6f5c")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("61dab70bfe12c3f0cacd55fe") }

}

username : ScumbagSteve

comment : Denied your PR cause I found a hack post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the posts document: "Reports a bug in your

code"

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.insert({username: 'ScumbagSteve', comment: "Denied your PR cause I found a hack",post:ObjectId("61da896cf1e258629fdd6f5e")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("61dab731fe12c3f0cacd55ff") }

}

# Querying related collections

1. find all users

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.users.find().pretty()

[

{

\_id: ObjectId("61dab4b3fe12c3f0cacd55f4"),

username: 'GoodGuyGreg',

first\_name: 'GoodGuy',

last\_name: 'Greg'

},

{

\_id: ObjectId("61dab4b3fe12c3f0cacd55f5"),

username: 'ScumbagSteve',

fullname: { first: 'Scumbag', last: 'Steve' }

}

]

1. find all posts

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.posts.find().pretty()

[

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f6"),

username: 'GoodGuyGreg',

title: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f7"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f8"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f9"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55fa"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55fb"),

username: 'ScumbagSteve',

title: 'Forks your repo in github',

body: 'Sets to private'

}

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.posts.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f6"),

username: 'GoodGuyGreg',

title: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f7"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f8"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

}

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.posts.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55f9"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55fa"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("61dab5e9fe12c3f0cacd55fb"),

username: 'ScumbagSteve',

title: 'Forks your repo in github',

body: 'Sets to private'

}

]

1. find all comments

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.find().pretty()

[

{

\_id: ObjectId("61dab6a5fe12c3f0cacd55fc"),

username: 'GoodGuyGreg',

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

post: ObjectId("61da896cf1e258629fdd6f5f")

},

{

\_id: ObjectId("61dab6defe12c3f0cacd55fd"),

username: 'GoodGuyGreg',

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

post: ObjectId("61da896cf1e258629fdd6f60")

},

{

\_id: ObjectId("61dab70bfe12c3f0cacd55fe"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("61da896cf1e258629fdd6f5c")

},

{

\_id: ObjectId("61dab731fe12c3f0cacd55ff"),

username: 'ScumbagSteve',

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

post: ObjectId("61da896cf1e258629fdd6f5e")

}

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("61dab6a5fe12c3f0cacd55fc"),

username: 'GoodGuyGreg',

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

post: ObjectId("61da896cf1e258629fdd6f5f")

},

{

\_id: ObjectId("61dab6defe12c3f0cacd55fd"),

username: 'GoodGuyGreg',

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

post: ObjectId("61da896cf1e258629fdd6f60")

}

]

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

Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.find({username:'ScumbagSteve'})

[

{

\_id: ObjectId("61dab70bfe12c3f0cacd55fe"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("61da896cf1e258629fdd6f5c")

},

{

\_id: ObjectId("61dab731fe12c3f0cacd55ff"),

username: 'ScumbagSteve',

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

post: ObjectId("61da896cf1e258629fdd6f5e")

}

]

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

Ans- Atlas atlas-ojy0k0-shard-0 [primary] mongodb\_practice> db.comments.find({post:ObjectId("61da896cf1e258629fdd6f5e")})

[

{

\_id: ObjectId("61dab731fe12c3f0cacd55ff"),

username: 'ScumbagSteve',

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

post: ObjectId("61da896cf1e258629fdd6f5e")

}

References: <https://docs.mongodb.com/manual/reference/method/db.collection.find/>

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@