**MongoDB Assignment 1**

**Insert Documents**

* db.movies.insertMany([ { title : "Fight Club", writer : "Chuck Palahniuk", year : 1999, actors : [ "Brad Pitt", "Edward Norton" ] }, { title : "Pulp Fiction", writer : "Quentin Tarantino", year : 1994, actors : [ "John Travolta", "Uma Thurman" ] } ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60e5d9cc881ceffce93d42c5"),

ObjectId("60e5d9cc881ceffce93d42c6")

]

}

* db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})

WriteResult({ "nInserted" : 1 })

* db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})

WriteResult({ "nInserted" : 1 })

* db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R Tolkien", year:"2013", franchise:"The Hobbit"})

WriteResult({ "nInserted" : 1 })

* db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", 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."})

WriteResult({ "nInserted" : 1 })

* db.movies.insert({title:"Pee Wee Herman's Big Adventures"})

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

**Query / Find Documents**

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 Pit”**

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

1. **get all documents with franchise set to “The hobbit”**

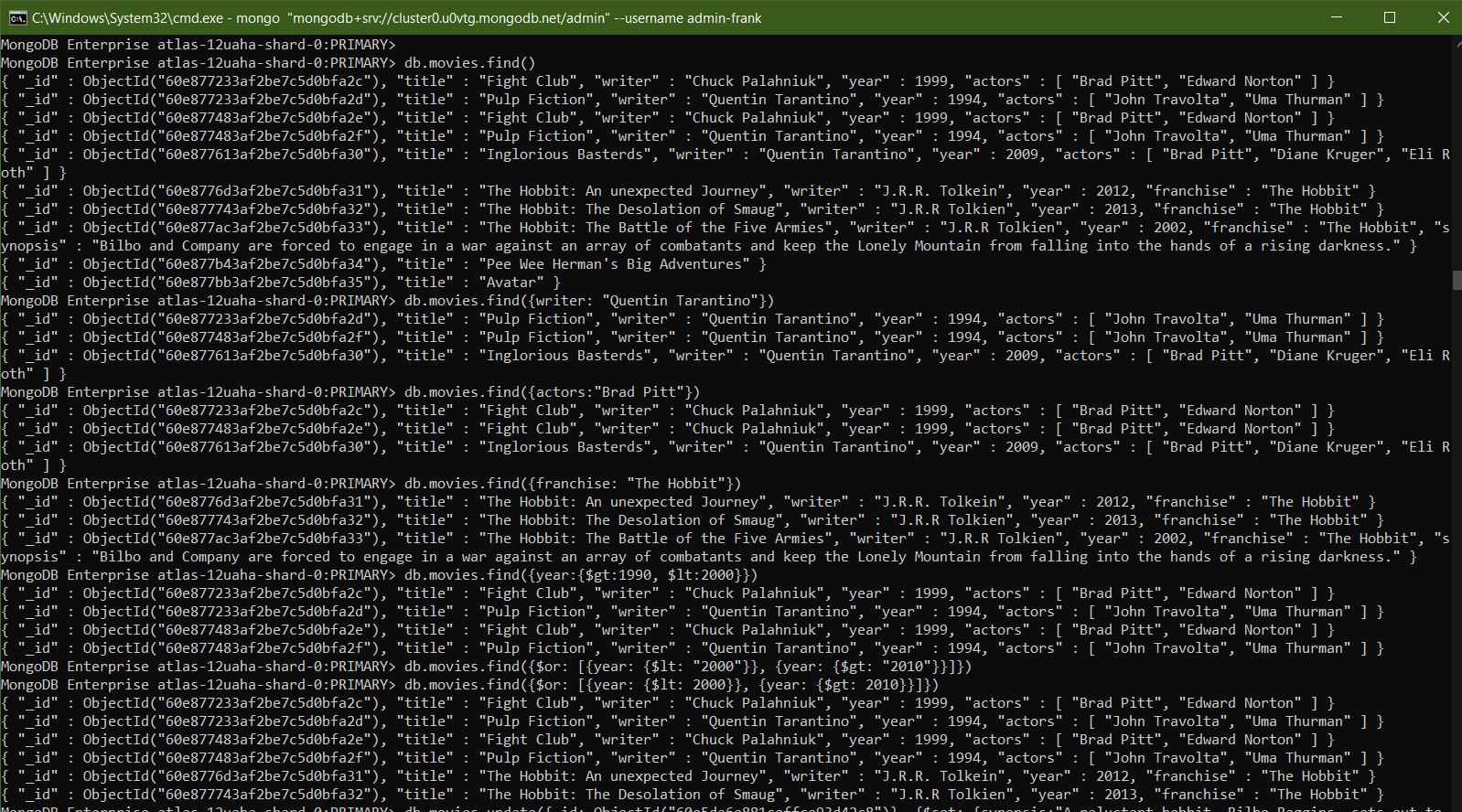
* db.movies.find({franchise: "The Hobbit"})

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

* db.movies.find({year:{$gt:1990, $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**

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({\_id: ObjectId("60e8776d3af2be7c5d0bfa31")}, {$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."}})

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

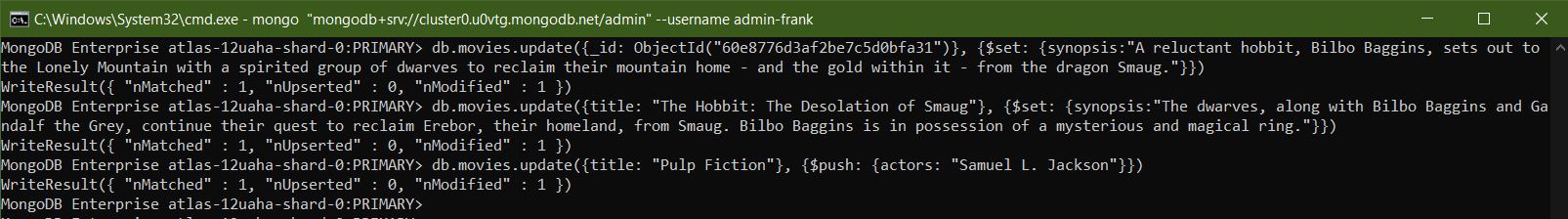
1. **add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."**

* 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"}})

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

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

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

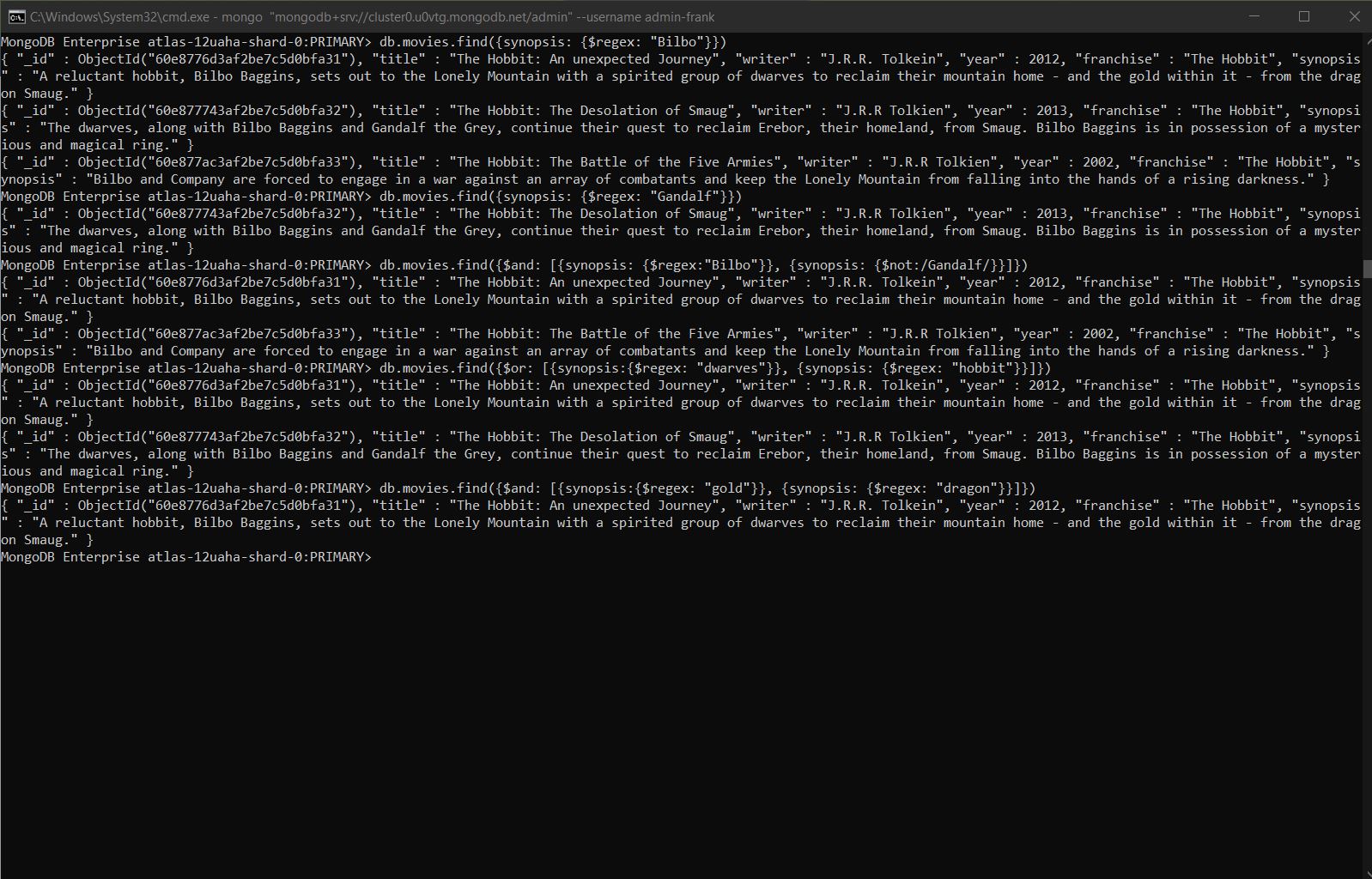
* db.movies.find({$and: [{synopsis: {$regex:"Bilbo"}}, {synopsis: {$not:/Gandalf/}}]})

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

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

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"}}]})



**Delete Documents**

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

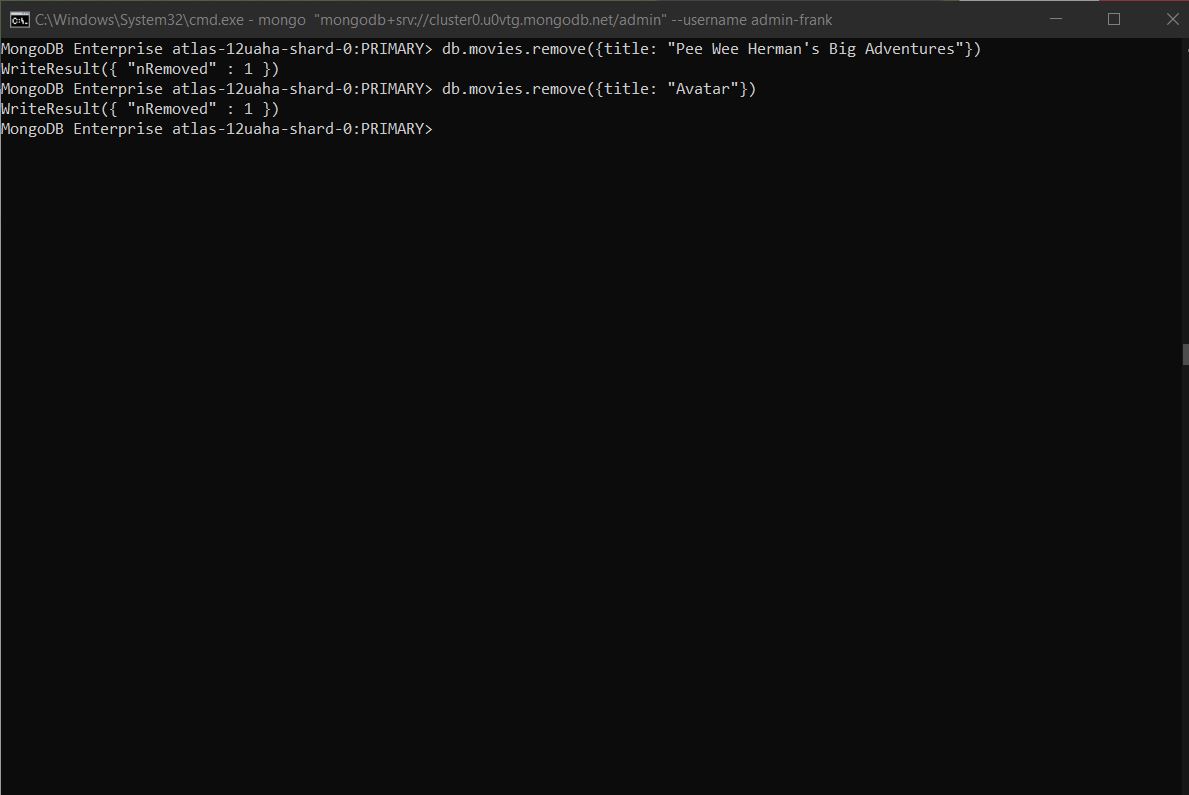
* db.movies.remove({title: "Pee Wee Herman's Big Adventures"})

WriteResult({ "nRemoved" : 1 })

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

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

WriteResult({ "nRemoved" : 1 })



**RELATIONSHIPS**

Insert the following documents into **users** collection

**username : GoodGuyGreg**

**first\_name : "Good Guy”**

**last\_name : "Greg"**

* db.users.insert({username: "GoodGuyGreg", first\_name:" Good Guy", last\_name: "Greg"})

WriteResult({ "nInserted" : 1 })

**username : ScumbagSteve**

**full\_name :**

**first : "Scumbag"**

**last : "Steve"**

* db.users.insert({username: "ScumbagSteve", fullname: {first: "Scumbag", last: "Steve"}}) WriteResult({ "nInserted" : 1 })

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

**username : GoodGuyGreg**

**title : Passes out at party**

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

* db.posts.insert({username: "GoodGuyGreg", title: "Passes out at Party", body: "Raises your credit score"})

WriteResult({ "nInserted" : 1 })

**username : GoodGuyGreg**

**title : Steals your identity**

**body : Raises your credit score**

* db.posts.insert({username: "GoodGuyGreg", title: "Steals your identity", body: "Raises your credit score"})

WriteResult({ "nInserted" : 1 })

**username : GoodGuyGreg**

**title : Reports a bug in your code**

**body : Sends you a Pull Request**

* db.posts.insert({username: "GoodGuyGreg", title: "Reports a bug in your code", body: "Sends you a pull request"})

WriteResult({ "nInserted" : 1 })

**username : ScumbagSteve**

**title : Borrows something**

**body : Sells it**

* db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})

WriteResult({ "nInserted" : 1 })

**username : ScumbagSteve**

**title : Borrows everything**

**body : The end**

* db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})

WriteResult({ "nInserted" : 1 })

**username : ScumbagSteve**

**title : Forks your repo on github**

**body : Sets to private**

* db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})

WriteResult({ "nInserted" : 1 })

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"**

* db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("60e5f934881ceffce93d42d2")})

WriteResult({ "nInserted" : 1 })

**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"**

* db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("60e5f93f881ceffce93d42d3")})

WriteResult({ "nInserted" : 1 })

**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**

* db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("60e5f949881ceffce93d42d4")})

WriteResult({ "nInserted" : 1 })

**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"**

* db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("60e5f888881ceffce93d42cf")})

WriteResult({ "nInserted" : 1 })

**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"**

* db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("60e5f92a881ceffce93d42d1")})

WriteResult({ "nInserted" : 1 })

**Querying related collections**

1. find all users

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

{

"\_id" : ObjectId("60e5f7ed881ceffce93d42cd"),

"username" : "GoodGuyGreg",

"first\_name" : " Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : ObjectId("60e5f867881ceffce93d42ce"),

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

1. find all posts

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

{

"\_id" : ObjectId("60e5f888881ceffce93d42cf"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60e5f90a881ceffce93d42d0"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60e5f92a881ceffce93d42d1"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("60e5f934881ceffce93d42d2"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60e5f93f881ceffce93d42d3"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("60e5f949881ceffce93d42d4"),

"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"})

{ "\_id" : ObjectId("60e5f888881ceffce93d42cf"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5f90a881ceffce93d42d0"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5f92a881ceffce93d42d1"), "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"})

{ "\_id" : ObjectId("60e5f934881ceffce93d42d2"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : ObjectId("60e5f93f881ceffce93d42d3"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : ObjectId("60e5f949881ceffce93d42d4"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

1. find all comments

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

{

"\_id" : ObjectId("60e601e0881ceffce93d42d5"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f934881ceffce93d42d2")

}

{

"\_id" : ObjectId("60e60252881ceffce93d42d6"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f93f881ceffce93d42d3")

}

{

"\_id" : ObjectId("60e603bc881ceffce93d42d7"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f949881ceffce93d42d4")

}

{

"\_id" : ObjectId("60e60460881ceffce93d42d8"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60e5f888881ceffce93d42cf")

}

{

"\_id" : ObjectId("60e6048f881ceffce93d42d9"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60e5f92a881ceffce93d42d1")

}

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

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

{

"\_id" : ObjectId("60e601e0881ceffce93d42d5"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f934881ceffce93d42d2")

}

{

"\_id" : ObjectId("60e60252881ceffce93d42d6"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f93f881ceffce93d42d3")

}

{

"\_id" : ObjectId("60e603bc881ceffce93d42d7"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60e5f949881ceffce93d42d4")

}

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

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

{

"\_id" : ObjectId("60e60460881ceffce93d42d8"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60e5f888881ceffce93d42cf")

}

{

"\_id" : ObjectId("60e6048f881ceffce93d42d9"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60e5f92a881ceffce93d42d1")

}

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

* { "\_id" : ObjectId("60e5f92a881ceffce93d42d1"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }