**MongoDB Assignment 1**

**Insert Document**

db.movies.insertMany([

... { title: 'Fight Club',

... writer: 'Chuck Palahniuko',

... year: 1999,

... actors: [ 'Brad Pitt', 'Ekward Norton']},

... {title: 'Pulp Fiction',

... writer: 'Quentin Tarantino',

... year: 1999,

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

... { 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 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'}

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("6106361cce0235e2758de117"),

ObjectId("6106361cce0235e2758de118"),

ObjectId("6106361cce0235e2758de119"),

ObjectId("6106361cce0235e2758de11a"),

ObjectId("6106361cce0235e2758de11b"),

ObjectId("6106361cce0235e2758de11c"),

ObjectId("6106361cce0235e2758de11d")

]

}

**Query / Find Documents**

**1.**

db.movies.find()

{ "\_id" : ObjectId("6106361cce0235e2758de117"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Ekward Norton" ] }

{ "\_id" : ObjectId("6106361cce0235e2758de118"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1999, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("6106361cce0235e2758de119"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("6106361cce0235e2758de11a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("6106361cce0235e2758de11b"), "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 against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness" }

{ "\_id" : ObjectId("6106361cce0235e2758de11c"), "title" : "Pee Wee Herman's Big Adventure" }

{ "\_id" : ObjectId("6106361cce0235e2758de11d"), "title" : "Avatar" }

**2.**

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

{ "\_id" : ObjectId("6106361cce0235e2758de118"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1999, "actors" : [ "John Travolta", "Uma Thurman" ] }

**3.**

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

{ "\_id" : ObjectId("6106361cce0235e2758de117"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Ekward Norton" ] }

**4.**

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

{ "\_id" : ObjectId("6106361cce0235e2758de119"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("6106361cce0235e2758de11a"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("6106361cce0235e2758de11b"), "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 against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness" }

**5.**

db.movies.find({year: {$lt: 2000}})

{ "\_id" : ObjectId("6106361cce0235e2758de117"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Ekward Norton" ] }

{ "\_id" : ObjectId("6106361cce0235e2758de118"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1999, "actors" : [ "John Travolta", "Uma Thurman" ] }

**6.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de117"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Ekward Norton"

]

}

{

"\_id" : ObjectId("6106361cce0235e2758de118"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1999,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"title" : "The Hobbit: An Unexpected Journey",

"writer" : "J.R.R. Tolkein",

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106361cce0235e2758de11a"),

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

"writer" : "J.R.R. Tolkein",

"year" : 2013,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106361cce0235e2758de11b"),

"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 against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness"

}

**Update Documents**

**1.**

db.movies.findOneAndUpdate({\_id: ObjectId("6106361cce0235e2758de119")}, {$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"}})

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"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"

}

**2.**

db.movies.findOneAndUpdate({\_id: ObjectId("6106361cce0235e2758de11a")}, {$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."}})

{

"\_id" : ObjectId("6106361cce0235e2758de11a"),

"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."

}

**3.**

db.movies.findOneAndUpdate({\_id: ObjectId("6106361cce0235e2758de118")}, {$set: {actors: ["Samuel L. Jackson", "John Travolta", "Uma Thurman"]}})

{

"\_id" : ObjectId("6106361cce0235e2758de118"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1999,

"actors" : [

"Samuel L. Jackson",

"John Travolta",

"Uma Thurman"

]

}

**Text Search**

**1.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de11a"),

"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("6106361cce0235e2758de11b"),

"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 against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness"

}

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"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"

}

**2.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de11a"),

"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."

}

**3.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de11b"),

"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 against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness"

}

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"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"

}

**4.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"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("6106361cce0235e2758de11a"),

"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."

}

**5.**

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

{

"\_id" : ObjectId("6106361cce0235e2758de119"),

"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"

}

**Delete Documents**

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

WriteResult({ "nRemoved" : 1 })

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

WriteResult({ "nRemoved" : 1 })

**Relationship**

**Insert Users Collection**

db.users.insertMany([

... {username: 'GoodGuyGreg',

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

... last\_name: 'Greg'},

... {username: 'ScumbagSteve',

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

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("610672381defc01e7ba96947"),

ObjectId("610672381defc01e7ba96948")

]

}

**Insert 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: 'Fork your repo on github',

... body: 'Sets to private'}

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("610677ee1defc01e7ba9694e"),

ObjectId("610677ee1defc01e7ba9694f"),

ObjectId("610677ee1defc01e7ba96950"),

ObjectId("610677ee1defc01e7ba96951"),

ObjectId("610677ee1defc01e7ba96952"),

ObjectId("610677ee1defc01e7ba96953")

]

}

**Insert comments collection**

db.comments.insertMany([

... {username: 'GoodGuyGreg',

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

... post: ObjectId("610677ee1defc01e7ba96951")},

... {username: 'GoodGuyGreg',

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

... post: ObjectId("610677ee1defc01e7ba96952")},

... {username: 'GoodGuyGreg',

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

... post: ObjectId("610677ee1defc01e7ba96953")},

... {username: 'ScumbagSteve',

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

... post: ObjectId("610677ee1defc01e7ba9694e")},

... {username: 'ScumbagSteve',

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

... post: ObjectId("610677ee1defc01e7ba96950")}

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("61067a681defc01e7ba96954"),

ObjectId("61067a681defc01e7ba96955"),

ObjectId("61067a681defc01e7ba96956"),

ObjectId("61067a681defc01e7ba96957"),

ObjectId("61067a681defc01e7ba96958")

]

}

**Querying Related Collection**

**1.**

db.users.find().pretty()

{

"\_id" : ObjectId("610672381defc01e7ba96947"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : ObjectId("610672381defc01e7ba96948"),

"username" : "ScumbagSteve",

"full\_name" : [

{

"first" : "Scumbag"

},

{

"last" : "Steve"

}

]

}

**2.**

db.posts.find().pretty()

{

"\_id" : ObjectId("610677ee1defc01e7ba9694e"),

"username" : "GoodGuyGreg",

"title" : "Passes out at party",

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

}

{

"\_id" : ObjectId("610677ee1defc01e7ba9694f"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96950"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96951"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96952"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96953"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

**3.**

db.posts.find({username: 'GoodGuyGreg'})

{ "\_id" : ObjectId("610677ee1defc01e7ba9694e"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }

{ "\_id" : ObjectId("610677ee1defc01e7ba9694f"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("610677ee1defc01e7ba96950"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a Pull Request" }

**4.**

db.posts.find({username: 'ScumbagSteve'}).pretty()

{

"\_id" : ObjectId("610677ee1defc01e7ba96951"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96952"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("610677ee1defc01e7ba96953"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

**5.**

db.comments.find().pretty()

{

"\_id" : ObjectId("61067a681defc01e7ba96954"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("610677ee1defc01e7ba96951")

}

{

"\_id" : ObjectId("61067a681defc01e7ba96955"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("610677ee1defc01e7ba96952")

}

{

"\_id" : ObjectId("61067a681defc01e7ba96956"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("610677ee1defc01e7ba96953")

}

{

"\_id" : ObjectId("61067a681defc01e7ba96957"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("610677ee1defc01e7ba9694e")

}

{

"\_id" : ObjectId("61067a681defc01e7ba96958"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("610677ee1defc01e7ba96950")

}

**6.**

db.comments.find({username: 'GoodGuyGreg'})

{ "\_id" : ObjectId("61067a681defc01e7ba96954"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : ObjectId("610677ee1defc01e7ba96951") }

{ "\_id" : ObjectId("61067a681defc01e7ba96955"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!", "post" : ObjectId("610677ee1defc01e7ba96952") }

{ "\_id" : ObjectId("61067a681defc01e7ba96956"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : ObjectId("610677ee1defc01e7ba96953") }

**7.**

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

{

"\_id" : ObjectId("61067a681defc01e7ba96957"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("610677ee1defc01e7ba9694e")

}

{

"\_id" : ObjectId("61067a681defc01e7ba96958"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("610677ee1defc01e7ba96950")

}

8.

db.posts.find({title: 'Reports a bug in your code'})

{ "\_id" : ObjectId("610677ee1defc01e7ba96950"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a Pull Request" }