**Assignment 1**

**Insert Documents**

> db.movies.insert([{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"}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 8,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

**Query/Find Documents**

1. > db.movies.find().pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2274"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2275"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2276"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2278"),

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

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

"year" : 2013,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2279"),

"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."

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f227a"),

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

}

{ "\_id" : ObjectId("6106ef3572e52c0a728f227b"), "title" : "Avatar" }

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

{

"\_id" : ObjectId("6106ef3572e52c0a728f2275"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2276"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

1. > db.movies.find({actors:/Brad Pitt/}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2274"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2276"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

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

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2278"),

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

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

"year" : 2013,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2279"),

"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."

}

1. > db.movies.find({$and:[{year:{$lt:2000}},{year:{$gt:1989}}]}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2274"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2275"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

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

{

"\_id" : ObjectId("6106ef3572e52c0a728f2274"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2275"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2278"),

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

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

"year" : 2013,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2279"),

"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."

}

**Update Documets**

1. > 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."}})

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

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

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

**Text Search**

1. > db.movies.find({synopsis:/Bilbo/}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2278"),

"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("6106ef3572e52c0a728f2279"),

"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."

}

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

"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."

}

1. > db.movies.find({synopsis:/Gandalf/}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2278"),

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

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

"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("6106ef3572e52c0a728f2279"),

"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."

}

1. > db.movies.find({$or:[{synopsis:/dwarves/},{synopsis:/hobbit/}]}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

"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."

}

1. > db.movies.find({$and:[{synopsis:/gold/},{synopsis:/dragon/}]}).pretty()

{

"\_id" : ObjectId("6106ef3572e52c0a728f2277"),

"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**

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

WriteResult({ "nRemoved" : 1 })

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

WriteResult({ "nRemoved" : 1 })

**Relationships**

> db.posts.insert([ {username:"GoodGuyGreg", title:"Passes out 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:"SumbagSteve", title:"Borrows everything", body:"The end"}, {username:"SumbagSteve", title:"Forks your repo on github", body:"Sets to private"}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 6,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

> db.comments.insert([ {username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("61077094f2ec5752a462beee")}, {username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("61077094f2ec5752a462beef")}, {username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("61077094f2ec5752a462bef0")}, {username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("61077094f2ec5752a462beeb")}, {username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("61077094f2ec5752a462beed")}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 5,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

**Querying related collections**

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

{

"\_id" : ObjectId("61076d3e66dd3ac8a9349b0b"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : ObjectId("61076d3e66dd3ac8a9349b0c"),

"username" : "ScumbagSteve",

"full\_name" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

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

{

"\_id" : ObjectId("61077094f2ec5752a462beeb"),

"username" : "GoodGuyGreg",

"title" : "Passes out a party",

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

}

{

"\_id" : ObjectId("61077094f2ec5752a462beec"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("61077094f2ec5752a462beed"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

{

"\_id" : ObjectId("61077094f2ec5752a462beee"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("61077094f2ec5752a462beef"),

"username" : "SumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("61077094f2ec5752a462bef0"),

"username" : "SumbagSteve",

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

"body" : "Sets to private"

}

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

{

"\_id" : ObjectId("61077094f2ec5752a462beeb"),

"username" : "GoodGuyGreg",

"title" : "Passes out a party",

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

}

{

"\_id" : ObjectId("61077094f2ec5752a462beec"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("61077094f2ec5752a462beed"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

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

{

"\_id" : ObjectId("61077094f2ec5752a462beee"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("61077094f2ec5752a462beef"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("61077094f2ec5752a462bef0"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

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

{

"\_id" : ObjectId("61077daaf2ec5752a462befb"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462beee")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462befc"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462beef")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462befd"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462bef0")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462befe"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("61077094f2ec5752a462beeb")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462beff"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("61077094f2ec5752a462beed")

}

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

{

"\_id" : ObjectId("61077daaf2ec5752a462befb"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462beee")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462befc"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462beef")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462befd"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("61077094f2ec5752a462bef0")

}

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

{

"\_id" : ObjectId("61077daaf2ec5752a462befe"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("61077094f2ec5752a462beeb")

}

{

"\_id" : ObjectId("61077daaf2ec5752a462beff"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("61077094f2ec5752a462beed")

}

1. > db.comments.find({post:ObjectId("61077094f2ec5752a462beed")}).pretty()

{

"\_id" : ObjectId("61077daaf2ec5752a462beff"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("61077094f2ec5752a462beed")

}