> use mongo\_practice

switched to db mongo\_practice

> db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})

WriteResult({ "nInserted" : 1 })

> db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"1994", actors:["John Travolta", "Uma Thurman"]})

WriteResult({ "nInserted" : 1 })

> db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"1994", 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:"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."})

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

{

"\_id" : ObjectId("60a5389f093d7d16025d94e4"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuk",

"year" : "1999",

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("60a538b8093d7d16025d94e5"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("60a538d7093d7d16025d94e6"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("60a538ea093d7d16025d94e7"),

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

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a53906093d7d16025d94e8"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a5392e093d7d16025d94e9"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"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("60a5394b093d7d16025d94ea"),

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

}

{ "\_id" : ObjectId("60a5395d093d7d16025d94eb"), "title" : "Avatar" }

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

{ "\_id" : ObjectId("60a538b8093d7d16025d94e5"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60a538d7093d7d16025d94e6"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

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

{ "\_id" : ObjectId("60a5389f093d7d16025d94e4"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a538d7093d7d16025d94e6"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "1994", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

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

{ "\_id" : ObjectId("60a538ea093d7d16025d94e7"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a53906093d7d16025d94e8"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a5392e093d7d16025d94e9"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "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." }

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

{ "\_id" : ObjectId("60a5389f093d7d16025d94e4"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

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

{ "\_id" : ObjectId("60a5389f093d7d16025d94e4"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a538ea093d7d16025d94e7"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a53906093d7d16025d94e8"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit" }

>> db.movies.update({\_id:ObjectId("5c9f98e5e5c2dfe9b3729bfe")}, {$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" : 0, "nUpserted" : 0, "nModified" : 0 })

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

{

"\_id" : ObjectId("60a5389f093d7d16025d94e4"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuk",

"year" : "1999",

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("60a538b8093d7d16025d94e5"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("60a538d7093d7d16025d94e6"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("60a538ea093d7d16025d94e7"),

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

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a53906093d7d16025d94e8"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a5392e093d7d16025d94e9"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"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("60a5394b093d7d16025d94ea"),

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

> db.movies.update({\_id:ObjectId("60a53906093d7d16025d94e8")}, {$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 })

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

{

"\_id" : ObjectId("60a53906093d7d16025d94e8"),

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

"writer" : "J.R.R Tolkien",

"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."

}

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

{

"\_id" : ObjectId("60a5389f093d7d16025d94e4"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuk",

"year" : "1999",

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("60a538b8093d7d16025d94e5"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("60a538d7093d7d16025d94e6"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : "1994",

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("60a538ea093d7d16025d94e7"),

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

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

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a53906093d7d16025d94e8"),

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

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("60a5392e093d7d16025d94e9"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"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("60a5394b093d7d16025d94ea"),

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

}

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

{ "\_id" : ObjectId("60a53906093d7d16025d94e8"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "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." }

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

{ "\_id" : ObjectId("60a53906093d7d16025d94e8"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "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." }

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

{ "\_id" : ObjectId("60a53906093d7d16025d94e8"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "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." }

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

{ "\_id" : ObjectId("60a538ea093d7d16025d94e7"), "title" : "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("60a5395d093d7d16025d94eb"), "title" : "Avatar" }

> db.movies.remove({\_id:ObjectId("60a5394b093d7d16025d94ea")})

WriteResult({ "nRemoved" : 1 })

> db.movies.remove({\_id:ObjectId("60a5395d093d7d16025d94eb")})

WriteResult({ "nRemoved" : 1 })

> db.users.insert({\_id:1,username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

WriteResult({ "nInserted" : 1 })

> db.users.insert({\_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

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

WriteResult({ "nInserted" : 1 })

>

> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("60a5f4c7d5fe738f463ca37d")})

WriteResult({ "nInserted" : 1 })

> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("60a5f499d5fe738f463ca37c")})

WriteResult({ "nInserted" : 1 })

> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("60a5f4d8d5fe738f463ca37e")})

WriteResult({ "nInserted" : 1 })

> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("60a5f467d5fe738f463ca379")})

WriteResult({ "nInserted" : 1 })

> db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("60a5f487d5fe738f463ca37b")})

WriteResult({ "nInserted" : 1 })

>

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

{

"\_id" : 1,

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : 2,

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

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

{

"\_id" : ObjectId("60a5f467d5fe738f463ca379"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a5f475d5fe738f463ca37a"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a5f487d5fe738f463ca37b"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("60a5f499d5fe738f463ca37c"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60a5f4c7d5fe738f463ca37d"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("60a5f4d8d5fe738f463ca37e"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

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

{ "\_id" : ObjectId("60a5f467d5fe738f463ca379"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60a5f475d5fe738f463ca37a"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60a5f487d5fe738f463ca37b"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

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

{ "\_id" : ObjectId("60a5f499d5fe738f463ca37c"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : ObjectId("60a5f4c7d5fe738f463ca37d"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : ObjectId("60a5f4d8d5fe738f463ca37e"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

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

{

"\_id" : ObjectId("60a5f597d5fe738f463ca37f"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a5f4c7d5fe738f463ca37d")

}

{

"\_id" : ObjectId("60a5f5cfd5fe738f463ca380"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a5f499d5fe738f463ca37c")

}

{

"\_id" : ObjectId("60a5f673d5fe738f463ca381"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a5f4d8d5fe738f463ca37e")

}

{

"\_id" : ObjectId("60a5f6a3d5fe738f463ca382"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60a5f467d5fe738f463ca379")

}

{

"\_id" : ObjectId("60a5f6e9d5fe738f463ca383"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60a5f487d5fe738f463ca37b")

}> db.comments.find({comment:"Reports a bug in your code"})