









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

CREATE DATABASE

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> use mongo\_practice

switched to db mongo\_practice

> db.createCollection("movies")

{ "ok" : 1 }

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

mongo\_practice 0.000GB

INSERT

> 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"] }])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

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

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

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

WriteResult({ "nInserted" : 1 })

> db.movies.insert( { title: "The Hobbit: The Battle of the Five Armies", writer: "J.R.R. Tolkein", year: 2012, franchise: "The Hobbit", synopsis: "Bibo and compay forced to engage in a war against army" })

WriteResult({ "nInserted" : 1 })

> db.movies.insert([ { title: "Avatar"},{title: "Pee Wee Herman's Big Adventure "}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

QUERY / FIND DOCUMENTS

> db.movies.find()

{ "\_id" : ObjectId("60a56d85eee1f8352c636d71"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a56d85eee1f8352c636d72"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60a56f25eee1f8352c636d73"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : ObjectId("60a56f25eee1f8352c636d74"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56f7deee1f8352c636d75"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56ffceee1f8352c636d76"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit", "synopsis" : "Bibo and compay forced to engage in a war against army" }

{ "\_id" : ObjectId("60a57054eee1f8352c636d77"), "title" : "Avatar" }

{ "\_id" : ObjectId("60a57054eee1f8352c636d78"), "title" : "Pee Wee Herman's Big Adventure " }

>

>

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

{ "\_id" : ObjectId("60a56d85eee1f8352c636d72"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60a56f25eee1f8352c636d73"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

>

>

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

{ "\_id" : ObjectId("60a56d85eee1f8352c636d71"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a56f25eee1f8352c636d73"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

>

>

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

{ "\_id" : ObjectId("60a56f25eee1f8352c636d74"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56f7deee1f8352c636d75"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56ffceee1f8352c636d76"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit", "synopsis" : "Bibo and compay forced to engage in a war against army" }

>

>

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

{ "\_id" : ObjectId("60a56d85eee1f8352c636d71"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a56d85eee1f8352c636d72"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

>

>

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

{ "\_id" : ObjectId("60a56d85eee1f8352c636d71"), "title" : "Fight Club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60a56d85eee1f8352c636d72"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60a56f25eee1f8352c636d74"), "title" : "The Hobbit: An Unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56f7deee1f8352c636d75"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkein", "year" : 2013, "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60a56ffceee1f8352c636d76"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit", "synopsis" : "Bibo and compay forced to engage in a war against army" }

UPDATING DOCUMENTS

> db.movies.update({title:"Pulp Fiction"},{$set:{actors:["John Travolta","Uma Thurman", "Samuel L"]}})

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

>

> db.movies.update({title:"The Hobbit: An Unexpected Journey"},{$set:{synopsis:" it is a great story"}})

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

>

>

> db.movies.update({title:"The Hobbit: The Desolation of Smaug"},{$set:{synopsis:" it is one of the greatest stories available in market"}})

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

TEXT SEARCH

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.createIndex({title:"text", synopsis:"text"})

{

"numIndexesBefore" : 2,

"numIndexesAfter" : 2,

"note" : "all indexes already exist",

"ok" : 1,

"$clusterTime" : {

"clusterTime" : Timestamp(1621515831, 4000),

"signature" : {

"hash" : BinData(0,"O2xw5fVNBJmuDAI/S4f3HsIGFfM="),

"keyId" : NumberLong("6963684266622320643")

}

},

"operationTime" : Timestamp(1621515831, 4000)

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.find({$text:{$search:"Bilbo"}}).pretty()

{

"\_id" : ObjectId("60a56f7deee1f8352c636d75"),

"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("60a56ffceee1f8352c636d76"),

"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("60a56f25eee1f8352c636d74"),

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

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "A reluctant hobbit, Bilbo Baggins, set out to the Lonely Mountain with a spirited group of drarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.find({$text:{$search:"Gandalf"}}).pretty()

{

"\_id" : ObjectId("60a56f7deee1f8352c636d75"),

"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."

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.find({$text:{$search:"Bilbo -Gandalf"}}).pretty()

{

"\_id" : ObjectId("60a56ffceee1f8352c636d76"),

"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("60a56f25eee1f8352c636d74"),

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

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "A reluctant hobbit, Bilbo Baggins, set out to the Lonely Mountain with a spirited group of drarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.find({$text:{$search:"dwarves hobbit"}}).pretty()

{

"\_id" : ObjectId("60a56f7deee1f8352c636d75"),

"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("60a56f25eee1f8352c636d74"),

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

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "A reluctant hobbit, Bilbo Baggins, set out to the Lonely Mountain with a spirited group of drarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

}

{

"\_id" : ObjectId("60a56ffceee1f8352c636d76"),

"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"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.find({$text:{$search:"\"gold\" \"dragon\""}}).pretty()

{

"\_id" : ObjectId("60a56f25eee1f8352c636d74"),

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

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "A reluctant hobbit, Bilbo Baggins, set out to the Lonely Mountain with a spirited group of drarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

}

DELETE DOCUMENTS

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.deleteOne({title:"Avatar"})

{ "acknowledged" : true, "deletedCount" : 1 }

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.movies.deleteOne({title:"Pee Wee Herman's Big Adventure "})

{ "acknowledged" : true, "deletedCount" : 1 }

RELATIONSHIPS

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.user.insert([{username:"GoodGuyGreg",first\_name:"Good Guy", laste\_name:"Greg"},{username:"ScumbagSteve",full\_name:[{first:"Scumbag"},{last:"Steve"}]}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.insert([{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:"Reposts a bug in your code", body:"Sends you a Pull Request"},{username:"ScumbagSteve",title:"Forks your repo on github", body:"Sets to private"},{username:"ScumbagSteve",title:"Borrows something",body:"Sells it"},{username:"ScumbagSteve",title:"Borrows everything",body:"The end"}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 6,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.comments.insert([{username:"GoodGuyGreg",comment:"Hope you got a good deal!", post:ObjectId("60a687b75086e69c6362b0fb")},{username:"GoodGuyGreg",comment:"What's mine is yours!", post: ObjectId("60a687b75086e69c6362b0fc")}, {username:"GoodGuyGreg",comment:"Don't violate the licensing agreement!", post: ObjectId("60a687b75086e69c6362b0fa")},{username:"ScumbagSteve",comment:"It still isn't clean", post: ObjectId("60a687b75086e69c6362b0f7")},{username:"ScumbagSteve",comment:"Denied your PR cause I found a hack",post: ObjectId("60a687b75086e69c6362b0f9")}])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 5,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

Querying Related Collections

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.user.find().pretty()

{

"\_id" : ObjectId("60a671a95086e69c6362b0da"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"laste\_name" : "Greg"

}

{

"\_id" : ObjectId("60a671a95086e69c6362b0db"),

"username" : "ScumbagSteve",

"full\_name" : [

{

"first" : "Scumbag"

},

{

"last" : "Steve"

}

]

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.find().pretty()

{

"\_id" : ObjectId("60a687b75086e69c6362b0f7"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

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

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f8"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f9"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fa"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fb"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fc"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("60a687b75086e69c6362b0f7"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

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

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f8"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f9"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.find().pretty()

{

"\_id" : ObjectId("60a687b75086e69c6362b0f7"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

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

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f8"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f9"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fa"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fb"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fc"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("60a687b75086e69c6362b0f7"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

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

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f8"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0f9"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.posts.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("60a687b75086e69c6362b0fa"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fb"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60a687b75086e69c6362b0fc"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.comments.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("60a68eb85086e69c6362b102"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a687b75086e69c6362b0fb")

}

{

"\_id" : ObjectId("60a68eb85086e69c6362b103"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a687b75086e69c6362b0fc")

}

{

"\_id" : ObjectId("60a68eb85086e69c6362b104"),

"username" : "GoodGuyGreg",

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

"post" : ObjectId("60a687b75086e69c6362b0fa")

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY>

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.comments.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("60a68eb85086e69c6362b105"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60a687b75086e69c6362b0f7")

}

{

"\_id" : ObjectId("60a68eb85086e69c6362b106"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60a687b75086e69c6362b0f9")

}

MongoDB Enterprise atlas-bogjb2-shard-0:PRIMARY> db.comments.find({post:ObjectId("60a687b75086e69c6362b0f9")}).pretty()

{

"\_id" : ObjectId("60a690c35086e69c6362b110"),

"username" : "ScumbagSteve",

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

"post" : ObjectId("60a687b75086e69c6362b0f9")

}