













use mongo\_practice

|  |
| --- |
| **'switched to db mongo\_practice'** |

show dbs

mongo\_practice 72.00 KiB

admin 336.00 KiB

local 4.26 GiB

show collections

**movies**

db.books.find()

db.movies.find()

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |
| { \_id: ObjectId(**"634eae01f06a5f69fbbf03cb"**), |
| title: **'Pulp Fiction'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'John Travolta'**, **'Uma Thurman'** ] } |

|  |
| --- |
| { \_id: ObjectId(**"634eaee0f06a5f69fbbf03cc"**), |
| title: **'Inglorious Basterds'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'Brad Pitt'**, **'Diane Kruger'**, **'Eli Roth'** ] } |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| title: **'The Hobbit:An Unexpected Journey'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2012'**, |
| franchise: **'The Hobbit'** } |

|  |
| --- |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| title: **'The Hobbit:The Desolation of Smaug'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2013'**, |
| franchise: **'The Hobbit'** } |
| { \_id: ObjectId(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

|  |
| --- |
| { \_id: ObjectId(**"634eb71cf06a5f69fbbf03d0"**), |
| title: [ **'Pee Wee Herman\'s Big Adventure'**, **'avatar'** ] } |

db.movies.find({title:"Pulp Fiction"})

|  |
| --- |
| { \_id: ObjectId(**"634eae01f06a5f69fbbf03cb"**), |
| title: **'Pulp Fiction'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'John Travolta'**, **'Uma Thurman'** ] } |

db.movies.findOne()

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |

db.movies.findOne({year:""})

|  |
| --- |
| null |

db.movies.findOne({year:"2012"})

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| title: **'The Hobbit:An Unexpected Journey'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2012'**, |
| franchise: **'The Hobbit'** } |

db.movies.find({year:"2012"})

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| title: **'The Hobbit:An Unexpected Journey'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2012'**, |
| franchise: **'The Hobbit'** } |
| { \_id: ObjectId(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

db.movies.find()

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |
| { \_id: ObjectId(**"634eae01f06a5f69fbbf03cb"**), |
| title: **'Pulp Fiction'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'John Travolta'**, **'Uma Thurman'** ] } |

|  |
| --- |
| { \_id: ObjectId(**"634eaee0f06a5f69fbbf03cc"**), |
| title: **'Inglorious Basterds'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'Brad Pitt'**, **'Diane Kruger'**, **'Eli Roth'** ] } |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| title: **'The Hobbit:An Unexpected Journey'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2012'**, |
| franchise: **'The Hobbit'** } |

|  |
| --- |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| title: **'The Hobbit:The Desolation of Smaug'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2013'**, |
| franchise: **'The Hobbit'** } |
| { \_id: ObjectId(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

|  |
| --- |
| { \_id: ObjectId(**"634eb71cf06a5f69fbbf03d0"**), |
| title: [ **'Pee Wee Herman\'s Big Adventure'**, **'avatar'** ] } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eae01f06a5f69fbbf03cb"**), |
| title: **'Pulp Fiction'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'John Travolta'**, **'Uma Thurman'** ] } |
| { \_id: ObjectId(**"634eaee0f06a5f69fbbf03cc"**), |
| title: **'Inglorious Basterds'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'Brad Pitt'**, **'Diane Kruger'**, **'Eli Roth'** ] } |

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

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

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

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |
| { \_id: ObjectId(**"634eaee0f06a5f69fbbf03cc"**), |
| title: **'Inglorious Basterds'**, |
| writer: **'Quentin Tarantino'**, |
| year: **'2009'**, |
| actors: [ **'Brad Pitt'**, **'Diane Kruger'**, **'Eli Roth'** ] } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eac74f06a5f69fbbf03ca"**), |
| title: **'Fight club'**, |
| writer: **'Chuck Palahniuko'**, |
| year: **'1999'**, |
| actors: [ **'Brad Pitt'**, **'Edward Norton'** ] } |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| title: **'The Hobbit:An Unexpected Journey'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2012'**, |
| franchise: **'The Hobbit'** } |

|  |
| --- |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| title: **'The Hobbit:The Desolation of Smaug'**, |
| writer: **'J.R.R.Tolkein'**, |
| year: **'2013'**, |
| franchise: **'The Hobbit'** } |
| { \_id: ObjectId(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

db.findAndUpdate()

[**TypeError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) db.findAndUpdate is not a function

db.movies.findAndUpdate()

[**TypeError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) db.movies.findAndUpdate is not a function

db.movies.update()

|  |
| --- |
| **'DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.'** |

[**MongoshInvalidInputError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) [COMMON-10001] Missing required argument at position 0 (Collection.update)

db.collection.updateOne()

[**MongoshInvalidInputError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) [COMMON-10001] Missing required argument at position 0 (Collection.updateOne)

db.movies.updateOne({title:"The Hobbit:An Unexpected Journey"},{$set{synopsis:"A reluctant hobbit Bilbo Baggins, sets out to the lonely mountain with a sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragn Smaug"}})

[**Error:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<omitted>...)} could not be cloned.

db.movies.find({synopsis})

[**ReferenceError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) synopsis is not defined

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

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| 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 sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragn Smaug'** } |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| 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(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

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

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

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| 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 sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragn Smaug'** } |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| 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(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| 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.'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| 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 sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragn Smaug'** } |
| { \_id: ObjectId(**"634eb3fff06a5f69fbbf03cf"**), |
| 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 combatents and keep the Lonely Mountain from falling into the hands of a rising darkness'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| 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 sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragn Smaug'** } |
| { \_id: ObjectId(**"634eb08cf06a5f69fbbf03ce"**), |
| 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.'** } |

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

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

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

|  |
| --- |
| { \_id: ObjectId(**"634eb004f06a5f69fbbf03cd"**), |
| 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 sprited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug'** } |

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

[**Error:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<omitted>...)} could not be cloned.

db.movies.remove({\_id

634eb71cf06a5f69fbbf03d0})

"\_id": {

"$oid": "634eb71cf06a5f69fbbf03d0"

},

"title": [

"Pee Wee Herman's Big Adventure",

"avatar"

]

})

"\_id": {

"$oid": "634eb71cf06a5f69fbbf03d0"

},

"title": [

"Pee Wee Herman's Big Adventure",

"avatar"

]

})

[**Error:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<omitted>...)} could not be cloned.

db.movies.remove({\_id

634eb71cf06a5f69fbbf03d0})

"\_id": {

"$oid": "634eb71cf06a5f69fbbf03d0"

},

"title": [

"Pee Wee Herman's Big Adventure",

"avatar"

]

})

"\_id": {

"$oid": "634eb71cf06a5f69fbbf03d0"

},

"title": [

"Pee Wee Herman's Big Adventure",

"avatar"

]

})

[**Error:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<omitted>...)} could not be cloned.

db.movies.remove({\_id: objectId('634eb71cf06a5f69fbbf03d0')})

[**ReferenceError:**](file:///C:\Users\vaisukum\AppData\Local\MongoDBCompass\app-1.33.1\resources\app.asar\build\index.html) objectId is not defined

db.createCollection("users")

|  |
| --- |
| { ok: 1 } |

show collections

**movies**  
**users**

show dbs

mongo\_practice 80.00 KiB

admin 336.00 KiB

local 4.26 GiB

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

|  |
| --- |
| **'DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.'** |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa7a08f0e43fd0cccd678"**) } } |

db.users.insert({username:"ScumbagSteve",full\_name:"",first\_name:"Scumbag",last\_name:"Steve"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa82d8f0e43fd0cccd679"**) } } |

db.createCollection("posts")

|  |
| --- |
| { ok: 1 } |

show collections

**movies**  
**posts**  
**users**

show dbs

mongo\_practice 152.00 KiB

admin 336.00 KiB

local 4.26 GiB

db.posts.insert({username:"GoodGuyGreg",title:"Passes out at party",body:"Wakes up early and cleans house"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa8d48f0e43fd0cccd67a"**) } } |

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

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa9108f0e43fd0cccd67b"**) } } |

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

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa9418f0e43fd0cccd67c"**) } } |

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

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa97a8f0e43fd0cccd67d"**) } } |

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

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa9908f0e43fd0cccd67e"**) } } |

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

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fa9bd8f0e43fd0cccd67f"**) } } |

db.createCollection("comments")

|  |
| --- |
| { ok: 1 } |

db.comments.insert({username:"GoodGuyGreg",comment:"Hope you got a good deal!",post:"634fa9908f0e43fd0cccd67e"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634faaaa8f0e43fd0cccd680"**) } } |

db.comments.insert({username:"GoodGuyGreg",comment:"What's mine is yours!",post:"634fa9908f0e43fd0cccd67e"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fabda8f0e43fd0cccd681"**) } } |

db.comments.insert({username:"GoodGuyGreg",comment:"Don't violate the licensing agreement!",post:"634fa9bd8f0e43fd0cccd67f"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634fac248f0e43fd0cccd682"**) } } |

db.comments.insert({username:"ScumbagSteve",comment:"It still is'nt clean",post:"634fa8d48f0e43fd0cccd67a"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634faca28f0e43fd0cccd683"**) } } |

db.comments.insert({username:"ScumbagSteve",comment:"Denied your PR cause I found a hack",post:"634fa9418f0e43fd0cccd67c"})

|  |
| --- |
| { acknowledged: true, |
| insertedIds: { **'0'**: ObjectId(**"634facf58f0e43fd0cccd684"**) } } |

db.users.find()

|  |
| --- |
| { \_id: ObjectId(**"634fa7a08f0e43fd0cccd678"**), |
| username: **'GoodGuyGreg'**, |
| first\_name: **'Good Guy'**, |
| last\_name: **'Greg'** } |
| { \_id: ObjectId(**"634fa82d8f0e43fd0cccd679"**), |
| username: **'ScumbagSteve'**, |
| full\_name: **''**, |
| first\_name: **'Scumbag'**, |
| last\_name: **'Steve'** } |

db.posts.find()

|  |
| --- |
| { \_id: ObjectId(**"634fa8d48f0e43fd0cccd67a"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Passes out at party'**, |
| body: **'Wakes up early and cleans house'** } |
| { \_id: ObjectId(**"634fa9108f0e43fd0cccd67b"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Steals your identity'**, |
| body: **'Raises your credit score'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fa9418f0e43fd0cccd67c"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Reports a bug in your code'**, |
| body: **'Sends you a pull request'** } |
| { \_id: ObjectId(**"634fa97a8f0e43fd0cccd67d"**), |
| username: **'ScumbagSteve'**, |
| title: **'Borrows everything'**, |
| body: **'Sells it'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fa9908f0e43fd0cccd67e"**), |
| username: **'ScumbagSteve'**, |
| title: **'Borrows everything'**, |
| body: **'The end'** } |
| { \_id: ObjectId(**"634fa9bd8f0e43fd0cccd67f"**), |
| username: **'ScumbagSteve'**, |
| title: **'Forks your repo on github'**, |
| body: **'Sets to private'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634fa8d48f0e43fd0cccd67a"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Passes out at party'**, |
| body: **'Wakes up early and cleans house'** } |
| { \_id: ObjectId(**"634fa9108f0e43fd0cccd67b"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Steals your identity'**, |
| body: **'Raises your credit score'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fa9418f0e43fd0cccd67c"**), |
| username: **'GoodGuyGreg'**, |
| title: **'Reports a bug in your code'**, |
| body: **'Sends you a pull request'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634fa97a8f0e43fd0cccd67d"**), |
| username: **'ScumbagSteve'**, |
| title: **'Borrows everything'**, |
| body: **'Sells it'** } |
| { \_id: ObjectId(**"634fa9908f0e43fd0cccd67e"**), |
| username: **'ScumbagSteve'**, |
| title: **'Borrows everything'**, |
| body: **'The end'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fa9bd8f0e43fd0cccd67f"**), |
| username: **'ScumbagSteve'**, |
| title: **'Forks your repo on github'**, |
| body: **'Sets to private'** } |

db.comments.find()

|  |
| --- |
| { \_id: ObjectId(**"634faaaa8f0e43fd0cccd680"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'Hope you got a good deal!'**, |
| post: **'634fa9908f0e43fd0cccd67e'** } |
| { \_id: ObjectId(**"634fabda8f0e43fd0cccd681"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'What\'s mine is yours!'**, |
| post: **'634fa9908f0e43fd0cccd67e'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fac248f0e43fd0cccd682"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'Don\'t violate the licensing agreement!'**, |
| post: **'634fa9bd8f0e43fd0cccd67f'** } |
| { \_id: ObjectId(**"634faca28f0e43fd0cccd683"**), |
| username: **'ScumbagSteve'**, |
| comment: **'It still is\'nt clean'**, |
| post: **'634fa8d48f0e43fd0cccd67a'** } |

|  |
| --- |
| { \_id: ObjectId(**"634facf58f0e43fd0cccd684"**), |
| username: **'ScumbagSteve'**, |
| comment: **'Denied your PR cause I found a hack'**, |
| post: **'634fa9418f0e43fd0cccd67c'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634faaaa8f0e43fd0cccd680"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'Hope you got a good deal!'**, |
| post: **'634fa9908f0e43fd0cccd67e'** } |
| { \_id: ObjectId(**"634fabda8f0e43fd0cccd681"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'What\'s mine is yours!'**, |
| post: **'634fa9908f0e43fd0cccd67e'** } |

|  |
| --- |
| { \_id: ObjectId(**"634fac248f0e43fd0cccd682"**), |
| username: **'GoodGuyGreg'**, |
| comment: **'Don\'t violate the licensing agreement!'**, |
| post: **'634fa9bd8f0e43fd0cccd67f'** } |

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

|  |
| --- |
| { \_id: ObjectId(**"634faca28f0e43fd0cccd683"**), |
| username: **'ScumbagSteve'**, |
| comment: **'It still is\'nt clean'**, |
| post: **'634fa8d48f0e43fd0cccd67a'** } |
| { \_id: ObjectId(**"634facf58f0e43fd0cccd684"**), |
| username: **'ScumbagSteve'**, |
| comment: **'Denied your PR cause I found a hack'**, |
| post: **'634fa9418f0e43fd0cccd67c'** } |