> use assignment1

switched to db assignment1

> db.movies.insertOne({title:"Fight club",writer:"Chuck Palahniuko",year:1999,actors:["Brad pitt","Edward Norton"]})

{

"acknowledged" : true,

"insertedId" : ObjectId("61e7e5d0799d384cce224d6d")

}

> db.movies.insertMany([{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",synopsis:"Bilbo and Company are forced are 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"}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("61e7e5ee799d384cce224d6e"),

ObjectId("61e7e5ee799d384cce224d6f"),

ObjectId("61e7e5ee799d384cce224d70"),

ObjectId("61e7e5ee799d384cce224d71"),

ObjectId("61e7e5ee799d384cce224d72"),

ObjectId("61e7e5ee799d384cce224d73")

]

}

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

{

"\_id" : ObjectId("61e7e5d0799d384cce224d6d"),

"title" : "Fight club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6e"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6f"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d70"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d71"),

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

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

"year" : 2013,

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced are 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("61e7e5ee799d384cce224d72"),

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

}

{ "\_id" : ObjectId("61e7e5ee799d384cce224d73"), "title" : "Avatar" }

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

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6e"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6f"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad pitt",

"Diane Kruger",

"Eli Roth"

]

}

> db.movies.find({actors:"brad pitt"}).pretty()

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

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

{

"\_id" : ObjectId("61e7e5d0799d384cce224d6d"),

"title" : "Fight club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6f"),

"title" : "Inglorious Basterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad pitt",

"Diane Kruger",

"Eli Roth"

]

}

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

{

"\_id" : ObjectId("61e7e5ee799d384cce224d70"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d71"),

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

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

"year" : 2013,

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced are 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({$and:[{Year:{$gt:1900}},{Year:{$lt:2000}}]}).pretty()

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

... ;

... ]

uncaught exception: SyntaxError: expected property name, got '{' :

@(shell):1:16

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

... ]

uncaught exception: SyntaxError: expected property name, got '{' :

@(shell):1:16

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

{

"\_id" : ObjectId("61e7e5d0799d384cce224d6d"),

"title" : "Fight club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6e"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

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

{

"\_id" : ObjectId("61e7e5d0799d384cce224d6d"),

"title" : "Fight club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d6e"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"John Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d70"),

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

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

"year" : 2012,

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("61e7e5ee799d384cce224d71"),

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

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

"year" : 2013,

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced are 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.updateOne({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."}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.movies.updateOne({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."}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

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

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.movies.createIndex({title:"text",writer:"text",franchise:"text",synopsis:"text"})

{

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"createdCollectionAutomatically" : false,

"ok" : 1

}

> db.movies.find({$text: {$search: 'Bilbo'})

... }

uncaught exception: SyntaxError: missing } after property list :

@(shell):1:41

> db.movies.find({$text: {$search: 'Bilbo'}})

{ "\_id" : ObjectId("61e7e5ee799d384cce224d71"), "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("61e7e5ee799d384cce224d70"), "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." }

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

{

"\_id" : ObjectId("61e7e5ee799d384cce224d71"),

"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("61e7e5ee799d384cce224d70"),

"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."

}

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

{

"\_id" : ObjectId("61e7e5ee799d384cce224d71"),

"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({$text: {$search: 'Bilbo' -'Gandalph'}}).pretty()

Error: error: {

"ok" : 0,

"errmsg" : "\"$search\" had the wrong type. Expected string, found double",

"code" : 14,

"codeName" : "TypeMismatch"

}

> db.movies.find({$text: {$search: 'dwarves'}, {$search: 'hobbit'})

... }

uncaught exception: SyntaxError: expected property name, got '{' :

@(shell):1:45

> db.movies.find({$text: {$search: 'dwarves'}, {$search: 'hobbit'}})

uncaught exception: SyntaxError: expected property name, got '{' :

@(shell):1:45

> db.movies.find({$text: {$search: 'dwarves', 'hobbit'}})

uncaught exception: SyntaxError: missing : after property id :

@(shell):1:52

> db.movies.find({$text: {$search: 'dwarves' 'hobbit'}})

uncaught exception: SyntaxError: missing } after property list :

@(shell):1:43

> db.movies.find({$text: {$search: 'dwarves hobbit'}})

{ "\_id" : ObjectId("61e7e5ee799d384cce224d70"), "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("61e7e5ee799d384cce224d71"), "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({$text: {$search: 'dwarves hobbit'}}).pretty()

{

"\_id" : ObjectId("61e7e5ee799d384cce224d70"),

"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("61e7e5ee799d384cce224d71"),

"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({$text: {$search: 'gold dragon'}}).pretty()

{

"\_id" : ObjectId("61e7e5ee799d384cce224d70"),

"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."

}

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

WriteResult({ "nRemoved" : 1 })

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

WriteResult({ "nRemoved" : 1 })

> db.users.insertMany([{username:"GoodGuyGreg",first\_name:"Good Guy",last\_name:"Greg"},{username:"ScumbagSteve",full\_name:"",first:"Scumbag",last:"Steve"}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("61e7ecdb799d384cce224d74"),

ObjectId("61e7ecdb799d384cce224d75")

]

}

> 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 : "Forks your repo on github" ,body :"Sets to private"}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("61e7ecf4799d384cce224d76"),

ObjectId("61e7ecf4799d384cce224d77"),

ObjectId("61e7ecf4799d384cce224d78"),

ObjectId("61e7ecf4799d384cce224d79"),

ObjectId("61e7ecf4799d384cce224d7a"),

ObjectId("61e7ecf4799d384cce224d7b")

]

}

> db.comments.insertMany([{username : "GoodGuyGreg",comment : "Hope you got a good deal!",post:"61e7ecf4799d384cce224d79"},{username : "GoodGuyGreg",comment : "What's mine is yours!",post : "61e7ecf4799d384cce224d79"},{username : "GoodGuyGreg",comment : "Don't violate the licensing agreement!",post : "61e7ecf4799d384cce224d7b"},{username : "ScumbagSteve",comment : "It still isn't clean",post : ""61e7ecf4799d384cce224d76"},{username : "ScumbagSteve",comment : "Denied your PR cause I found a hack",post : ""61e7ecf4799d384cce224d78"}])

uncaught exception: SyntaxError: identifier starts immediately after numeric literal :

@(shell):1:397

> db.comments.insertMany([{username : "GoodGuyGreg",comment : "Hope you got a good deal!",post:"61e7ecf4799d384cce224d79"},{username : "GoodGuyGreg",comment : "What's mine is yours!",post : "61e7ecf4799d384cce224d79"},{username : "GoodGuyGreg",comment : "Don't violate the licensing agreement!",post : "61e7ecf4799d384cce224d7b"},{username : "ScumbagSteve",comment : "It still isn't clean",post : "61e7ecf4799d384cce224d76"},{username : "ScumbagSteve",comment : "Denied your PR cause I found a hack",post : "61e7ecf4799d384cce224d78"}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("61e7f128799d384cce224d7c"),

ObjectId("61e7f128799d384cce224d7d"),

ObjectId("61e7f128799d384cce224d7e"),

ObjectId("61e7f128799d384cce224d7f"),

ObjectId("61e7f128799d384cce224d80")

]

}

> b.users.find().pretty()

uncaught exception: ReferenceError: b is not defined :

@(shell):1:1

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

{

"\_id" : ObjectId("61e7ecdb799d384cce224d74"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : ObjectId("61e7ecdb799d384cce224d75"),

"username" : "ScumbagSteve",

"full\_name" : "",

"first" : "Scumbag",

"last" : "Steve"

}

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

{

"\_id" : ObjectId("61e7ecf4799d384cce224d76"),

"username" : "GoodGuyGreg",

"title" : "Passes out at party",

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

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d77"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d78"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d79"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d7a"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d7b"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

> db.posts.find({author: 'GoodGuyGreg'}).pretty()

> '

uncaught exception: SyntaxError: '' literal not terminated before end of script :

@(shell):1:1

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

{

"\_id" : ObjectId("61e7ecf4799d384cce224d76"),

"username" : "GoodGuyGreg",

"title" : "Passes out at party",

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

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d77"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d78"),

"username" : "GoodGuyGreg",

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

"body" : "Sends you a Pull Request"

}

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

{

"\_id" : ObjectId("61e7ecf4799d384cce224d79"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d7a"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("61e7ecf4799d384cce224d7b"),

"username" : "ScumbagSteve",

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

"body" : "Sets to private"

}

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

{

"\_id" : ObjectId("61e7f128799d384cce224d7f"),

"username" : "ScumbagSteve",

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

"post" : "61e7ecf4799d384cce224d76"

}

{

"\_id" : ObjectId("61e7f128799d384cce224d80"),

"username" : "ScumbagSteve",

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

"post" : "61e7ecf4799d384cce224d78"

}

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

> db.comments.find({post: "61e7f128799d384cce224d80"}).pretty()

> db.comments.find({post: "61e7ecf4799d384cce224d78"}).pretty()

{

"\_id" : ObjectId("61e7f128799d384cce224d80"),

"username" : "ScumbagSteve",

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

"post" : "61e7ecf4799d384cce224d78"

}

>