

Mongodb assignments 1

```
Command Prompt - mongo
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

> use mongo_practice
switched to db mongo_practice
> use movies
switched to db movies
> db
movies
> db.movies.insertOne({title:"fight club",writer:"Chuck Palahniuko",year:1999,actors:["Brad Pitt","Edward Norton"]})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7caa0c9a10d193e692a61")
}
> db.movies.insertOne({title:"pulp fiction",writer:"Quentin Tarantino",year:1994,actors:["John Travolta","Uma Thurman"]})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7cb15c9a10d193e692a62")
}
> db.movies.insertOne({title:"Inglorious Basterds",writer:"Quentine Tarantino",year:2009,actors:["Brad Pitt","Diane Kruger","Eli Roth"]})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7cb8ec9a10d193e692a63")
}
> db.movies.insertOne({title:"The Hobbit:An Unexpected Journey",writer:"J.R.R.Tolkein",year:2012,franchise:"The Hobbit"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7cc6ec9a10d193e692a64")
}
> db.movies.insertOne({title:"The Hobbit:The Desolation Of Samug",writer:"J.R.R.Tolkein",year:2013,franchise:"The Hobbit"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7cd11c9a10d193e692a65")
}
> db.movies.insertOne({title:"The Hobbit:The Battle of the Five Armies",writer:"J.R.R.Tolkein",year:2012,franchise:"The Hobbit"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7cd8bc9a10d193e692a66")
}
> db.movies.insertOne({title:"Avatar"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7ce0ac9a10d193e692a67")
}
>
```

```
Command Prompt - mongo
> db.movies.find().pretty()
{
  "_id" : ObjectId("61e7b47b9e4c5fa79824ce3"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "Year" : 1999,
  "actors" : [
    "Brad pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61e7caa0c9a10d193e692a61"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61e7cb15c9a10d193e692a62"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("61e7cb8ec9a10d193e692a63"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentine Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
{
  "_id" : ObjectId("61e7cc6ec9a10d193e692a64"),
  "title" : "The Hobbit:An Unexpected Journey",
  "writer" : "J.R.R.Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61e7cd11c9a10d193e692a65"),
  "title" : "The Hobbit:The Desolation Of Samug",
  "writer" : "J.R.R.Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61e7cd8bc9a10d193e692a66"),
  "title" : "The Hobbit:The Battle of the Five Armies",
  "writer" : "J.R.R.Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61e7ce0ac9a10d193e692a67"),
  "title" : "Avatar",
  "writer" : "",
  "year" : 2009,
  "franchise" : ""
}
```

```
Command Prompt - mongo

{"title": "pulp fiction",
  "writer": "Quentin Tarantino",
  "year": 1994,
  "actors": [
    "John Travolta",
    "Uma Thurman"
  ]
}

{"_id": ObjectId("61e7cb8ec9a10d193e692a63"),
  "title": "Inglorious Basterds",
  "writer": "Quentin Tarantino",
  "year": 2009,
  "actors": [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

{"_id": ObjectId("61e7cc6ec9a10d193e692a64"),
  "title": "The Hobbit: An Unexpected Journey",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "franchise": "The Hobbit"
}

{"_id": ObjectId("61e7cd11c9a10d193e692a65"),
  "title": "The Hobbit: The Desolation Of Smaug",
  "writer": "J.R.R. Tolkien",
  "year": 2013,
  "franchise": "The Hobbit"
}

{"_id": ObjectId("61e7cd8bc9a10d193e692a66"),
  "title": "The Hobbit: The Battle of the Five Armies",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "franchise": "The Hobbit"
}

{"_id": ObjectId("61e7ce0ac9a10d193e692a67"), "title": "Avatar" }

{"_id": ObjectId("61e7d0bac9a10d193e692a68"),
  "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."
}

Type here to search 30°C Smoke ENG IN 2:27 PM 1/19/2022
```

```
Command Prompt - mongo

{"_id": ObjectId("61e7d0bac9a10d193e692a68"),
  "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(actors:"Brad Pitt").pretty()
uncaught exception: SyntaxError: unexpected token: ':'
@:(shell):1:14
> db.movies.find(actors:"Brad Pitt").pretty()
{"_id": ObjectId("61e7caa0c9a10d193e692a61"),
  "title": "Fight Club",
  "writer": "Chuck Palahniuk",
  "year": 1999,
  "actors": [
    "Brad Pitt",
    "Edward Norton"
  ]
}

{"_id": ObjectId("61e7cb8ec9a10d193e692a63"),
  "title": "Inglorious Basterds",
  "writer": "Quentin Tarantino",
  "year": 2009,
  "actors": [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

> db.movies.find(franchise:"The Hobbit").pretty()
{"_id": ObjectId("61e7cc6ec9a10d193e692a64"),
  "title": "The Hobbit: An Unexpected Journey",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "franchise": "The Hobbit"
}

{"_id": ObjectId("61e7cd11c9a10d193e692a65"),
  "title": "The Hobbit: The Desolation Of Smaug",
  "writer": "J.R.R. Tolkien",
  "year": 2013,
  "franchise": "The Hobbit"
}

{"_id": ObjectId("61e7cd8bc9a10d193e692a66"),
  "title": "The Hobbit: The Battle of the Five Armies",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "franchise": "The Hobbit"
}

Type here to search 30°C Smoke ENG IN 2:45 PM 1/19/2022
```

```
Command Prompt - mongo

{
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61e7d0bac9a10d193e692a60"),
  "title" : "The Hobbit:The Battle of the five Armise",
  "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."
}
> db.movies.find({$and:[{Year:{$gt:1900}},{Year:{$lt:2000}}]}).pretty()
{
  "_id" : ObjectId("61e7b47b9e4c5fa7982c4ce3"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "Year" : 1999,
  "actors" : [
    "Brad pitt",
    "Edward Norton"
  ]
}
> db.movies.find({$or:[{Year:{$lt:2000}},{Year:{$gt:2010}}]}).pretty()
{
  "_id" : ObjectId("61e7b47b9e4c5fa7982c4ce3"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "Year" : 1999,
  "actors" : [
    "Brad pitt",
    "Edward Norton"
  ]
}
> db.movies.updateOne({title:"The hobbit:An Unexpected Journey"},{$set:{synopsis:"A reluctant hobbit,Bilbo Baggins and sets out to the lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
> db.movies.updateOne({title:"The Hobbit:The Desolation of smaug"},{$set:{synopsis:"The dwarves ,along with Bilbo Baggins and Gandalf thr grey,continue their quest to reclaim Erebor,their homeland,from smag.Bilbo Baggins is in p
ossession of a mysterious and magical ring."}})
uncaught exception: SyntaxError: missing ) after property list :
@shell:1:65
> db.movies.updateOne({title:"The Hobbit:The Desolation of smaug"},{$set:{synopsis:"The dwarves ,along with Bilbo Baggins and Gandalf thr grey,continue their quest to reclaim Erebor,their homeland,from smag.Bilbo
Baggins is in possession of a mysterious and magical ring."}})
uncaught exception: SyntaxError: missing ) after property list :
@shell:1:82
> db.movies.updateOne({title:"The Hobbit:The Desolation of smaug "},{$set:{synopsis:"The dwarves ,along with Bilbo Baggins and Gandalf thr grey,continue their quest to reclaim Erebor,their homeland,from smag.Bil
bo Baggins is in possession of a mysterious and magical ring."}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
> db.movies.updateOne({title:"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
>
```

```
Command Prompt - mongo
ossession of a mysterious and magical ring."}})
uncaught exception: SyntaxError: missing ) after property list :
@shell:1:65
> db.movies.updateOne({title:"The Hobbit:The Desolation of smaug"},{$set:{synopsis:"The dwarves ,along with Bilbo Baggins and Gandalf thr grey,continue their quest to reclaim Erebor,their homeland,from smag.Bilbo
Baggins is in possession of a mysterious and magical ring."}})
uncaught exception: SyntaxError: missing ) after property list :
@shell:1:82
> db.movies.updateOne({title:"The Hobbit:The Desolation of smaug "},{$set:{synopsis:"The dwarves ,along with Bilbo Baggins and Gandalf thr grey,continue their quest to reclaim Erebor,their homeland,from smag.Bil
bo Baggins is in possession of a mysterious and magical ring."}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
> db.movies.updateOne({title:"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
> db.movies.createIndex({title:"text",writer:"text",franchise:"text",synopsis:"text"})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
> db.movies.find({$text:{$search:"Bilbo"}})
uncaught exception: SyntaxError: '' literal not terminated before end of script :
@shell:1:41
> db.movies.find({$text:{$search:"Bilbo"}})
> db.movies.find({$text:{$search:"Gandalf"}})
> db.movies.find({$text:{$search:"Bilbo -Gandalf"}})
> db.movies.find({$text:{$search:"dwarves hobbit"}})
{ "_id" : ObjectId("61e7cd11c9a10d193e692a65"), "title" : "The Hobbit:The Desolation Of Smaug", "writer" : "J.R.R.Tolkein", "year" : 2013, "franchise" : "The Hobbit" }
{ "_id" : ObjectId("61e7cc6ec9a10d193e692a64"), "title" : "The Hobbit:An Unexpected Journey", "writer" : "J.R.R.Tolkein", "year" : 2012, "franchise" : "The Hobbit" }
{ "_id" : ObjectId("61e7d0bac9a10d193e692a60"), "title" : "The Hobbit:The Battle of the five Armise", "writer" : "J.R.R.Tolkein", "year" : 2012, "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are fo
rced to engage in a war against an array of combatants and keep the lonely mountain from falling into the hands of a rising darkness." }
{ "_id" : ObjectId("61e7d0bac9a10d193e692a60"), "title" : "The Hobbit:The Battle of the Five Armies", "writer" : "J.R.R.Tolkein", "year" : 2012, "franchise" : "The Hobbit" }
> db.movies.find({$text:{$search:"gold dragon"}})
> db.movies.remove({title:"Pee Wee Herman's Big Adventure"})
WriteResult({ "nRemoved" : 0 })
> db.movies.remove({title:"Avatar"})
WriteResult({ "nRemoved" : 1 })
> use users
switched to db users
> db
users
> db.users.insertOne({username:"GoodGuyGreg",first_name:"Good Guy",last_name:"Greg"},{username:"ScumbagSteve",full_name:"",first:"Scumbag",last:"Steve"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("61e7fba1c9a10d193e692a69")
}
> use posts
switched to db posts
> db
posts
>
```

```
Command Prompt - mongo
> db.posts.insertOne({username:"GoodGuyGreg",title:"Passes out at party",body:"Makes up early and cleans house"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80a00c9a10d193e692a6c")
})
> db.posts.insertOne({username:"GoodGuyGreg",title:"Steals your identity",body:"Raises your credit score"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80a3ac9a10d193e692a6d")
})
> db.posts.insertOne({username:"GoodGuyGreg",title:"reports a bug in your code",body:"Sends you a pull request"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80a65c9a10d193e692a6e")
})
> db.posts.insertOne({username:"ScumbagSteve",title:"Borrows something",body:"Sells it"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80a92c9a10d193e692a6f")
})
> db.posts.insertOne({username:"ScumbagSteve",title:"Borrows everything",body:"The end"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80abcc9a10d193e692a70")
})
> db.posts.insertOne({username:"ScumbagSteve",title:"Forks your repo on github",body:"sets to private"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80b11c9a10d193e692a71")
})
> use collection
switched to db collection
> db.collection.insertOne({username:"GoodGuyGreg",comment:"Hope you got a good idea",post:"61e80a92c9a10d193e692a6f"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80badc9a10d193e692a72")
})
> db.collection.insertOne({username:"GoodGuyGreg",comment:"what mine is your!",post:"61e80abcc9a10d193e692a70"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80c40c9a10d193e692a73")
})
> db.posts.insertOne({username:"GoodGuyGreg",comment:"Don't violate the licensing agreements",body:"61e80b11c9a10d193e692a71"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80d55c9a10d193e692a74")
})
> db.posts.insertOne({username:"GoodGuyGreg",comment:"It still isn't clean",body:"61e80a00c9a10d193e692a6c"})
({
  "acknowledged" : true,
```

```
Command Prompt - mongo
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80d55c9a10d193e692a74")
})
> db.posts.insertOne({username:"GoodGuyGreg",comment:"It still isn't clean",body:"61e80a00c9a10d193e692a6c"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80e25c9a10d193e692a75")
})
> db.posts.insertOne({username:"GoodGuyGreg",comment:"Denied your PR cause I found a hack",body:"61e80a65c9a10d193e692a6e"})
({
  "acknowledged" : true,
  "insertedId" : ObjectId("61e80e7ec9a10d193e692a76")
})
> db.user.find()
> db.posts.find().pretty()
{
  "_id" : ObjectId("61e80a00c9a10d193e692a6c"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Makes up early and cleans house"
}
{
  "_id" : ObjectId("61e80a3ac9a10d193e692a6d"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("61e80a65c9a10d193e692a6e"),
  "username" : "GoodGuyGreg",
  "title" : "reports a bug in your code",
  "body" : "Sends you a pull request"
}
{
  "_id" : ObjectId("61e80a92c9a10d193e692a6f"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("61e80abcc9a10d193e692a70"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("61e80b11c9a10d193e692a71"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
```

```
Command Prompt - mongo

{
  "_id" : ObjectId("61e80b11c9a10d193e692a71"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "sets to private"
}

{
  "_id" : ObjectId("61e80d55c9a10d193e692a74"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreements",
  "body" : "61e80b11c9a10d193e692a71"
}

{
  "_id" : ObjectId("61e80e75c9a10d193e692a75"),
  "username" : "GoodGuyGreg",
  "comment" : "It still isn't clean",
  "body" : "61e80a00c9a10d193e692a6c"
}

{
  "_id" : ObjectId("61e80e7ec9a10d193e692a76"),
  "username" : "GoodGuyGreg",
  "comment" : "Denied your PR cause I found a hack",
  "body" : "61e80a5c9a10d193e692a6e"
}
}
db.users.find().pretty()
> db.posts.find((username:"GoodGuyGreg")).pretty()
{
  "_id" : ObjectId("61e80a00c9a10d193e692a6c"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Makes up early and cleans house"
}

{
  "_id" : ObjectId("61e80a3ac9a10d193e692a6d"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}

{
  "_id" : ObjectId("61e80a65c9a10d193e692a6e"),
  "username" : "GoodGuyGreg",
  "title" : "reports a bug in your code",
  "body" : "Sends you a pull request"
}

{
  "_id" : ObjectId("61e80d55c9a10d193e692a74"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreements",
  "body" : "61e80b11c9a10d193e692a71"
}
}
```

```
Command Prompt - mongo

{
  "comment" : "It still isn't clean",
  "body" : "61e80a00c9a10d193e692a6c"
}

{
  "_id" : ObjectId("61e80e7ec9a10d193e692a76"),
  "username" : "GoodGuyGreg",
  "comment" : "Denied your PR cause I found a hack",
  "body" : "61e80a5c9a10d193e692a6e"
}
}
> db.comments.find().pretty()
> db.comment.find().pretty()
> db
collection
> db.collection.find().pretty()
{
  "_id" : ObjectId("61e801d1c9a10d193e692a6b"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good idea",
  "post" : "61e7fdcf9a10d193e692a6a"
}

{
  "_id" : ObjectId("61e80badc9a10d193e692a72"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good idea",
  "post" : "61e80a92c9a10d193e692a6f"
}

{
  "_id" : ObjectId("61e80c40c9a10d193e692a73"),
  "username" : "GoodGuyGreg",
  "comment" : "what mine is your!",
  "post" : "61e80abcc9a10d193e692a70"
}
}
> db.collection.find((username:"GoodGuyGreg")).pretty()
{
  "_id" : ObjectId("61e801d1c9a10d193e692a6b"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good idea",
  "post" : "61e7fdcf9a10d193e692a6a"
}

{
  "_id" : ObjectId("61e80badc9a10d193e692a72"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good idea",
  "post" : "61e80a92c9a10d193e692a6f"
}

{
  "_id" : ObjectId("61e80c40c9a10d193e692a73"),
  "username" : "GoodGuyGreg",
  "comment" : "what mine is your!",
  "post" : "61e80abcc9a10d193e692a70"
}
}
```