

mongodb assignment 1:

```
mongod --version
MongoDB server version: 5.0.1
=====
warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====

The server generated these startup warnings when booting:
2021-07-30T21:11:13.365+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
...
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
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
> db.createCollection("movies")
{
  "ok" : 1
}
> db.movies.insert({title: "Fight Club", writer: "Chuck Palahniuk", year: 1999, actors:["Brad Pitt", "Edward Norton"]})
{
  "nInserted" : 1
}
> db.collection.find()
> db.movies.insert({title: "Pulp Fiction", writer: "Quentin Tarantino", year: 1994, actors:["John Travolta", "Uma Thurman"]})
{
  "nInserted" : 1
}
> db.movies.insert({title: "Inglourious Basterds", writer: "Quentin Tarantino", year: 2009, actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})
{
  "nInserted" : 1
}
> db.movies.insertMany([{"title": "The Hobbit An Unexpected Journey", writer: "J.R.R Tolkein", year: 2012, franchise: "The Hobbit", synopsis: "Bilbo and Company are forced to engage in a war against an army of combatants and keep the lonely mountain from falling into the hands of a rising darkness."}, {"title": "The Desolation of Smaug", writer: "J.R.R Tolkein", year: 2013, franchise: "The Hobbit"}, {"title": "The Hobbit: The Battle of the five armies", writer: "J.R.R Tolkein", year: 2014, franchise: "The Hobbit", synopsis: "Bilbo and Company are forced to engage in a war against an army of combatants and keep the lonely mountain from falling into the hands of a rising darkness."}, {"title": "Avatar"}])
{
  "nInserted" : 3
}
uncaught exception: SyntaxError: missing } after property list
@(shell):i:118
> db.movies.insertMany([{"title": "The Hobbit An Unexpected Journey", writer: "J.R.R Tolkein", year: 2012, franchise: "The Hobbit", synopsis: "Bilbo and Company are forced to engage in a war against an army of combatants and keep the lonely mountain from falling into the hands of a rising darkness."}, {"title": "The Desolation of Smaug", writer: "J.R.R Tolkein", year: 2013, franchise: "The Hobbit"}, {"title": "The Hobbit: The Battle of the five armies", writer: "J.R.R Tolkein", year: 2014, franchise: "The Hobbit", synopsis: "Bilbo and Company are forced to engage in a war against an army of combatants and keep the lonely mountain from falling into the hands of a rising darkness."}, {"title": "Avatar"}])
{
  "nInserted" : 3
}
uncaught exception: SyntaxError: missing } after property list
@(shell):i:137
>
Go to Settings to activate Windows.
```

MongoDB Compass - localhost:27017/mongo_practice.movies

Connect View Collection Help

Local

4 DBS 2 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.1 Community

Filter your data

admin config local mongo_practice movies

mongo_practice.movies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA

OPTIONS FIND RESET ...

Displaying documents 1 - 8 of 8

Activate Windows
Go to Settings to activate Windows.

Documents

```
{
  "_id": {
    "$oid": "610425cae278d939d23b63b"
  },
  "title": "Fight Club",
  "writer": "Chuck Palahniuk",
  "year": 1999,
  "actors": ["Brad Pitt", "Edward Norton"]
}

{
  "_id": {
    "$oid": "610426b85e278d939d23b63c"
  },
  "title": "Pulp Fiction",
  "writer": "Quentin Tarantino",
  "year": 1994,
  "actors": ["John Travolta", "Uma Thurman"]
}

{
  "_id": {
    "$oid": "6104273a5e278d939d23b63d"
  },
  "title": "Inglourious Basterds",
  "writer": "Quentin Tarantino",
  "year": 2009,
  "actors": ["Brad Pitt", "Diane Kruger", "Eli Roth"]
}
```

> MONGOSH

Type here to search

Cloudy 25°C 10:14 PM 7/30/2021

MongoDB Compass - localhost:27017/mongo_practice.movies

Connect View Collection Help

Local

4 DBS 2 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.1 Community

Filter your data

> admin

> config

> local

> mongo_practice

movies

mongo_practice.movies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW

Displaying documents 1 - 8 of 8

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

...
{"_id": {"\$oid": "61042c685e278d939d23b640"}, "title": "the hobbit an unexpected journey", "writer": "J.R.R. Tolkein", "year": 2012, "franchise": "The hobbit"}
...
{"_id": {"\$oid": "61042c685e278d939d23b641"}, "title": "The hobbit: Desolation of Smaug", "writer": "J.R.R. Tolkein", "year": 2013, "franchise": "the Hobbit"}
...
{"_id": {"\$oid": "61042c685e278d939d23b642"}, "title": "The Hobbit: The battle of five armies", "writer": "J.R.R. Tolkein", "year": 2012, "franchise": "the Hobbit", "synopsis": "bilbo and company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling into the hands of rising darkness."}

Activate Windows
Go to Settings to activate Windows.

> MONGOSH

Type here to search

25°C AQI 98 10:15 PM 7/30/2021

MongoDB Compass - localhost:27017/mongo_practice.movies

Connect View Collection Help

Local

4 DBS 2 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.1 Community

Filter your data

> admin

> config

> local

> mongo_practice

movies

mongo_practice.movies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW

Displaying documents 1 - 8 of 8

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

...
{"_id": {"\$oid": "61042abc5e278d939d23b63e"}, "title": "Pee Wee Herman's Big Adventure"}
...
{"_id": {"\$oid": "61042abc5e278d939d23b63f"}, "title": "Avatar"}
...
{"_id": {"\$oid": "61042c685e278d939d23b640"}, "title": "the hobbit an unexpected journey", "writer": "J.R.R. Tolkein", "year": 2012, "franchise": "The hobbit"}
...
{"_id": {"\$oid": "61042c685e278d939d23b641"}, "title": "The hobbit: Desolation of Smaug", "writer": "J.R.R. Tolkein", "year": 2013, "franchise": "the Hobbit"}

Activate Windows
Go to Settings to activate Windows.

> MONGOSH

Type here to search

25°C AQI 98 10:14 PM 7/30/2021

Query/find documents

```
 Command Prompt - mongo
"year" : 2012,
"franchise" : "the Hobbit",
"synopsis" : "bilbo and company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling in the hands of rising darkness."
}
> db.movies.find({writer: "Quentin Tarantino"})
{
  "_id" : ObjectId("610426b85e278d939d23b63c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.movies.find({writer: "Quentin Tarantino", "year": 2009})
{
  "_id" : ObjectId("6104273a5e278d939d23b63d"),
  "title" : "Inglourious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
> db.movies.find({actors: "Brad Pitt"})
{
  "_id" : ObjectId("610425ca5e278d939d23b63b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
> db.movies.find({franchise: "the Hobbit"}.$or franchise: "The hobbit")
caught exception: SyntaxError: missing } after property list:
@shell1::1:40
> db.movies.find({franchise: "the Hobbit"}, $or {franchise: "The hobbit"})
caught exception: SyntaxError: missing } after argument list:
@shell1::1:46
> db.movies.find({{{franchise: "the Hobbit"}, $or {franchise: "The hobbit"}}})
caught exception: SyntaxError: missing ] after element list:
@shell1::1:47
> db.movies.find({franchise: "the Hobbit"})
{
  "_id" : ObjectId("61042c605e278d939d23b641"),
  "title" : "The hobbit: Desolation of Smaug",
  "writer" : "J.R.R Tolkein",
  "year" : 2013,
  "franchise" : "the Hobbit"
}
{
  "_id" : ObjectId("61042c605e278d939d23b642"),
  "title" : "The Hobbit: The battle of five armies",
  "writer" : "J.R.R Tolkein",
  "year" : 2012,
  "franchise" : "the Hobbit",
  "synopsis" : "bilbo a
nd company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling in the hands of rising darkness."
}
>
```

Activate Windows
Go to Settings to activate Windows.

```
 Command Prompt - mongo
nd company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling in the hands of rising darkness." }
> db.movies.find({year: {$lt: 2000}}).pretty()
{
  "_id" : ObjectId("610425ca5e278d939d23b63b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("610426b85e278d939d23b63c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
>
```

Activate Windows
Go to Settings to activate Windows.

```
 Windows Type here to search O 25°C Cloudy 10:31 PM 7/30/2021
```

```

Command Prompt - mongo
> db.movies.find({$or: [{year: {$lt: 2000}}, {year: {$gt:2010}}]}).pretty()
{
    "_id" : ObjectId("610425ca5e278d939d23b63b"),
    "title" : "Fight Club",
    "writer" : "Chuck Palahniuk",
    "year" : 1999,
    "actors" : [
        "Brad Pitt",
        "Edward Norton"
    ]
}

{
    "_id" : ObjectId("610426b85e278d939d23b63c"),
    "title" : "Pulp Fiction",
    "writer" : "Quentin Tarantino",
    "year" : 1994,
    "actors" : [
        "John Travolta",
        "Uma Thurman"
    ]
}

{
    "_id" : ObjectId("61042c685e278d939d23b640"),
    "title" : "the hobbit an unexpected journey",
    "writer" : "J.R.R Tolkein",
    "year" : 2012,
    "franchise" : "The hobbit"
}

{
    "_id" : ObjectId("61042c685e278d939d23b641"),
    "title" : "The hobbit: Desolation of Smaug",
    "writer" : "J.R.R Tolkein",
    "year" : 2013,
    "franchise" : "the Hobbit"
}

{
    "_id" : ObjectId("61042c685e278d939d23b642"),
    "title" : "The Hobbit: The battle of five armies",
    "writer" : "J.R.R Tolkein",
    "year" : 2014,
    "franchise" : "the Hobbit",
    "synopsis" : "bilbo and company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling in the hands of rising darkness."
}
>

```

Activate Windows
Go to Settings to activate Windows.

update documents

```

Command Prompt - mongo
"synopsis" : "bilbo and company are forced to engage in a war against and army of combatants and keep the lonely mountain from falling in the hands of rising darkness."
> db.movies.update({title: "the hobbit an unexpected journey"}, {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 Smug."})
writeResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.update({title: "The hobbit: Desolation of Smaug"}, {synopsis: "The dwarves, along with Bilbo Baggins and Gandalf the Gray, continue their quest to reclaim Erebor, their homeland, f
rom Smug. Bilbo Baggins is in possession of a mysterious and magical ring."})
writeResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

Activate Windows
Go to Settings to activate Windows.

Text search

```
 Command Prompt - mongo
> db.movies.find({$text: {$search: "Gandalf"}})
Error: error: {
    "ok" : 0,
    "errmsg" : "text index required for $text query",
    "code" : 27,
    "codeName" : "IndexNotFound"
}
> db.movies.createIndex({synopsis: "text", type:"text"})
{
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "createdCollectionAutomatically" : false,
    "ok" : 1
}
> db.movies.find({$text: {$search: "Bilbo Baggins is in possession of a mysterious and magical ring."}})
{
    "_id" : ObjectId("61042c685e278d939d23b641"),
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Gray, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
{
    "_id" : ObjectId("61042c685e278d939d23b640"),
    "synopsis" : "A reluctant, hobbit Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home and the gold within it- from the dragon Smaug."
}
> db.movies.find({$text: {$search: "Gandalf the Gray, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}})
{
    "_id" : ObjectId("61042c685e278d939d23b641"),
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Gray, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
> db.movies.find({$text: {$search: "Bilbo Baggins and Gandalf"}})
{
    "_id" : ObjectId("61042c685e278d939d23b640"),
    "synopsis" : "A reluctant, hobbit Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home and the gold within it- from the dragon Smaug."
}
> db.movies.find({$text: {$search: "dwarves hobbit"}})
{
    "_id" : ObjectId("61042c685e278d939d23b640"),
    "synopsis" : "A reluctant, hobbit Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home and the gold within it- from the dragon Smaug."
}
{
    "_id" : ObjectId("61042c685e278d939d23b641"),
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Gray, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
> db.movies.find({$text: {$search: "gold dragon"}})
{
    "_id" : ObjectId("61042c685e278d939d23b640"),
    "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."
}
>
```

delete

```
 Command Prompt - mongo
> db.movies.remove({title: "Pee Wee Herman's Big Adventure"})
WriteResult({ "nRemoved" : 1 })
> db.movies.remove({title: "Avatar"})
WriteResult({ "nRemoved" : 1 })
>
```

relationships

```
 Command Prompt - mongo
> db.createCollection("users")
{ "ok" : 1 }
> db.users.insert({username: "GoodGuyGreg", first_name: "Good Guy", last_name: "Greg"})
WriteResult({ "nInserted" : 1 })
> db.users.insert({username: "ScumbagSteve", full_name: [{"first: "Scumbag", last: "Steve"}]})
WriteResult({ "nInserted" : 1 })
> db.users.find().pretty()
{
  "_id" : ObjectId("61043d935e278d939d23b643"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : ObjectId("61043d935e278d939d23b644"),
  "username" : "ScumbagSteve",
  "full_name" : [
    {
      "first" : "Scumbag",
      "last" : "Steve"
    }
  ]
}
>
```

Activate Windows
Go to Settings to activate Windows.

```
 Command Prompt - mongo
> 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"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("610440cf5e278d939d23b646"),
    ObjectId("610440cf5e278d939d23b647")
  ]
}
> db.posts.insertMany([{"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"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("610441725e278d939d23b648"),
    ObjectId("610441725e278d939d23b649"),
    ObjectId("610441725e278d939d23b64a")
  ]
}
> db.posts.insert({username: "ScumbagSteve", title: "Forks your repo on github", body: "Sets to private"})
uncaught exception: SyntaxError: unexpected token: '{' :
@(shell):1:15
> db.posts.insert({username: "ScumbagSteve", title: "Forks your repo on github", body: "Sets to private"})
WriteResult({ "nInserted" : 1 })
>
```

Activate Windows
Go to Settings to activate Windows.

```
 Command Prompt - mongo
> db.users.find().pretty()
{
  "_id" : ObjectId("61043d3f5e278d939d23b643"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"

  "_id" : ObjectId("61043d095e278d939d23b644"),
  "username" : "ScumbagSteve",
  "full_name" : [
    {
      "first" : "Scumbag",
      "last" : "Steve"
    }
  ]
}
>
```

```
 Command Prompt - mongo
@(shell):1:15
> db.posts.insert({username: "ScumbagSteve", title: "Forks your repo on github", body: "Sets to private"})
> writeResult({ "nInserted" : 1 })
> db.posts.find().pretty()
{
  "_id" : ObjectId("61043e585e278d939d23b645"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"

  "_id" : ObjectId("610440cf5e278d939d23b646"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"

  "_id" : ObjectId("610441725e278d939d23b647"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"

  "_id" : ObjectId("610441725e278d939d23b648"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a pull request"

  "_id" : ObjectId("610441725e278d939d23b649"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"

  "_id" : ObjectId("610441725e278d939d23b64a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"

  "_id" : ObjectId("610441e25e278d939d23b64b"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
>
```

```
Command Prompt - mongo
> db.posts.find({username: "GoodGuyGreg"}).pretty()
{
    "_id" : 1,
    "username" : "GoodGuyGreg",
    "title" : "Passes out at party",
    "body" : "Wakes up early and cleans house"

    "_id" : 2,
    "username" : "GoodGuyGreg",
    "title" : "Steals your identity",
    "body" : "Raises your credit score"

    "_id" : 3,
    "username" : "GoodGuyGreg",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request"
}
> db.posts.find({username: "ScumbagSteve"}).pretty()
{
    "_id" : 4,
    "username" : "ScumbagSteve",
    "title" : "Borrows Something",
    "body" : "Sells it"

    "_id" : 5,
    "username" : "ScumbagSteve",
    "title" : "Borrows everything",
    "body" : "The end"

    "_id" : 6,
    "username" : "ScumbagSteve",
    "title" : "Forks your repo on github",
    "body" : "Sets it to private"
}

>
```

```
Command Prompt - mongo
> db.comments.find()
{
    "_id" : ObjectId("61058f2e137e9edafa7ba262"),
    "username" : "GoodGuyGreg",
    "comment" : "Hope you got a good deal!",
    "post" : 4 }

    "_id" : ObjectId("61058f2e137e9edafa7ba263"),
    "username" : "GoodGuyGreg",
    "comment" : "What's mine is yours!",
    "post" : 5 }

    "_id" : ObjectId("61058fc137e9edafa7ba264"),
    "username" : "GoodGuyGreg",
    "comment" : "Don't violate the licensing agreement!",
    "post" : 6 }

    "_id" : ObjectId("61058fc137e9edafa7ba265"),
    "username" : "ScumbagSteve",
    "comment" : "It still isn't clean",
    "post" : 1 }

    "_id" : ObjectId("61058fc137e9edafa7ba266"),
    "username" : "ScumbagSteve",
    "comment" : "Denied your PR cause I found a hack",
    "post" : 3 }

> db.comments.find({username: "GoodGuyGreg"}).pretty()
{
    "_id" : ObjectId("61058f2e137e9edafa7ba262"),
    "username" : "GoodGuyGreg",
    "comment" : "Hope you got a good deal!",
    "post" : 4 }

    "_id" : ObjectId("61058f2e137e9edafa7ba263"),
    "username" : "GoodGuyGreg",
    "comment" : "What's mine is yours!",
    "post" : 5 }

    "_id" : ObjectId("61058fc137e9edafa7ba264"),
    "username" : "GoodGuyGreg",
    "comment" : "Don't violate the licensing agreement!",
    "post" : 6 }

}

>
```

```
Command Prompt - mongo
> db.posts.aggregate([{$match: {title: "Reports a bug in your code"}}, {$lookup: {from: 'comments', localField:'_id', foreignField: 'post', as: 'comments'}}]).pretty()
{
    "_id" : 3,
    "username" : "GoodGuyGreg",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request",
    "comments" : [
        {
            "_id" : ObjectId("61058fc137e9edafa7ba266"),
            "username" : "ScumBagSteve",
            "comment" : "Denied your PR cause I found a hack",
            "post" : 3
        }
    ]
}
>
```

