

```
ea Command Prompt - mongo
Microsoft Windows [Version 10.0.19041.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kirti>mongo
MongoDB shell version v5.0.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "_id": UUID("9023a81d-4d61-4491-854c-9a17b8ce0e3d") }
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Warning: the "mongo" shell has been superseded by "mongosh",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:
  2021-07-30T22:41:10.447+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.movies.insertMany({
...   title: "Fight Club",
...   writer: "Chuck Palahniuk",
...   year: 1999,
...   actors: ["Brad Pitt", "Edward Norton"],
...   title: "Pulp Fiction",
...   writer: "Quentin Tarantino",
...   year: 1994,
```

```
Command Prompt - mongo
... writer: "Quentin Tarantino",
... year: 1994,
... actors: ["John Travolta", "Uma Thurman"],
... {title: "Inglourious 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",
... title: "The Hobbit: The Battle of the Five Armies",
... writer: "J.R.R. Tolkein",
... year: 2013,
... 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",
... },
... {title: "Pee Wee Herman's Big Adventure",
... {title: "Avatar"
... }]
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("6107f370ed5b17447bdd7077"),
        ObjectId("6107f370ed5b17447bdd7078"),
        ObjectId("6107f370ed5b17447bdd7079"),
        ObjectId("6107f370ed5b17447bdd707a"),
        ObjectId("6107f370ed5b17447bdd707b"),
        ObjectId("6107f370ed5b17447bdd707c"),
        ObjectId("6107f370ed5b17447bdd707d"),
        ObjectId("6107f370ed5b17447bdd707e")
    ]
}
> db.movies.find()
{
    "_id" : ObjectId("6107f370ed5b17447bdd7077"),
    "title" : "Fight Club",
    "writer" : "Chuck Palahniuk",
    "year" : 1999,
    "actors" : [ "Brad Pitt", "Edward Norton" ]
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd7078"),
    "title" : "Pulp Fiction",
    "writer" : "Quentin Tarantino",
    "year" : 1994,
    "actors" : [ "John Travolta", "Uma Thurman" ]
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd7079"),
    "title" : "Inglourious Basterds",
    "writer" : "Quentin Tarantino",
    "year" : 2009,
    "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ]
}
28°C Light rain ENG 22:17
02-08-2021
```

```
 Command Prompt - mongo
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd7079"),
  "title" : "Inglourious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707c"),
  "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("6107f370ed5b17447bdd707d"),
  "title" : "Pee Wee Herman's Big Adventure"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707e"),
  "title" : "Avatar"
}

> db.movies.find({writer: "Quentin Tarantino"})
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd7079"),
  "title" : "Inglourious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

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

{
  "_id" : ObjectId("6107f370ed5b17447bdd7077"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}

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

{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707c"),
  "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"
}

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

{
  "_id" : ObjectId("6107f370ed5b17447bdd7077"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}

{
  "db.movies.find({year: {$gt: 2010}})"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}

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

```
 Command Prompt - mongo
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707c"),
  "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"
}

{
  "db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707a")},{$set: {synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprint of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
```

```
PS Command Prompt - mongo
{
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}

{
  "_id" : ObjectId("6107f370ed5b17447bdd707c"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "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.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707a")},{$set: {synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprinted group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707a")},{$set: {synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprinted group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "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 sprinted group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707b")},{$set: {synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring"}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707b")},{$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"}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "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.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd7078")},{$set: {actors: ["Samuel L. Jackson", "John travolta", "Uma Thurman"]}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd7078")},{$set: {actors: ["Samuel L. Jackson", "John travolta", "Uma Thurman"]}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "Samuel L. Jackson",
    "John travolta",
    "Uma Thurman"
  ]
}
28°C Light rain 22:19 ENG IN 02-08-2021
```

```
PS Command Prompt - mongo
{
  "db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd707b")},{$set: {synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring"}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit",
  "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring"
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd7078")},{$set: {actors: ["Samuel L. Jackson", "John travolta", "Uma Thurman"]}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd7078")},{$set: {actors: ["Samuel L. Jackson", "John travolta", "Uma Thurman"]}})
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "Samuel L. Jackson",
    "John travolta",
    "Uma Thurman"
  ]
}
28°C Light rain 22:20 ENG IN 02-08-2021
```

```
 Command Prompt - mongo
    "actors" : [
        "John Travolta",
        "Uma Thurman"
    ]
}
> db.movies.findOneAndUpdate({_id: ObjectId("6107f370ed5b17447bdd7078")}, {$set: {actors: ["Samuel L. Jackson", "John travolta", "Uma Thurman"]}})
{
    "_id" : ObjectId("6107f370ed5b17447bdd7078"),
    "title" : "Pulp Fiction",
    "writer" : "Quentin Tarantino",
    "year" : 1994,
    "actors" : [
        "Samuel L. Jackson",
        "John travolta",
        "Uma Thurman"
    ]
}
> db.movies.find({$text:{$search: "Bilbo"}}).pretty()
Error: error: {
    "ok" : 0,
    "errmsg" : "text index required for $text query",
    "code" : 27,
    "codeName" : "IndexNotFound"
}
> db.movies.find({$text: {$search: "Bilbo"}}).pretty()
Error: error: {
    "ok" : 0,
    "errmsg" : "text index required for $text query",
    "code" : 27,
    "codeName" : "IndexNotFound"
}
> db.movies.find({synopsis:/Bilbo/}).pretty
function() {
    this._prettyShell = true;
    return this;
}
> db.movies.find({synopsis:/Bilbo/}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707a"),
    "title" : "The Hobbit: An Unexpected Journey",
    "writer" : "J.R.R. tolkien",
    "year" : 2012,
    "franchise" : "The Hobbit",
    "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprinted group of dwarves to reclaim their mountain home - and the gold with in it - from the dragon Smaug"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit",
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Bagginsis in pos session of a mysterious and magical ring"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707c"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2012,
    "franchise" : "The Hobbit",
    "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising da rkness"
}
> db.movies.find({synopsis:/Gandalf/}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit",
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Bagginsis in pos session of a mysterious and magical ring"
}
> db.movies.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not: /Gandalf/}}]}).pretty()
uncaught exception: SyntaxError: expected property name, got '{'
@(shell):1:23
> db.movies.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not: /Gandalf/}}]}).pretty()
uncaught exception: SyntaxError: missing } after property list
@(shell):1:69
```

```
 Command Prompt - mongo
    "title" : "The Hobbit: An Unexpected Journey",
    "writer" : "J.R.R. tolkien",
    "year" : 2012,
    "franchise" : "The Hobbit",
    "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprinted group of dwarves to reclaim their mountain home - and the gold with in it - from the dragon Smaug"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "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 Bagginsis in pos session of a mysterious and magical ring"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707c"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "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 da rkness"
}
> db.movies.find({synopsis:/Gandalf/}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "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 Bagginsis in pos session of a mysterious and magical ring"
}
> db.movies.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not: /Gandalf/}}]}).pretty()
uncaught exception: SyntaxError: expected property name, got '{'
@(shell):1:23
> db.movies.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not: /Gandalf/}}]}).pretty()
uncaught exception: SyntaxError: missing } after property list
@(shell):1:69
```

```
 Command Prompt - mongo
uncaught exception: SyntaxError: missing ) after property list :
@(shell):1:69
> db.movies.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not: /Gandalf/}}]}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707a"),
    "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 sprinted group of dwarves to reclaim their mountain home - and the gold with in it - from the dragon Smaug"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707c"),
    "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 Bagginsis in pos session of a mysterious and magical ring"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "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 Bagginsis in pos session of a mysterious and magical ring"
}
> db.movies.find({$or: [{synopsis:/dwarves/},{synopsis: /hobbit/}]}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707a"),
    "title" : "The Hobbit: An Unexpected Journey",
    "writer" : "J.R.R. tolkein",
    "year" : 2012,
    "franchise" : "The Hobbit",
    "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a sprinted group of dwarves to reclaim their mountain home - and the gold with in it - from the dragon Smaug"
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd707b"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit",
    "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug, Bilbo Bagginsis in pos session of a mysterious and magical ring"
}
> db.movies.find({$and: [{synopsis:/gold/},{synopsis: /dragon/}]}).pretty()

```

```
 Command Prompt - mongo
> db.movies.find({$and: [{synopsis:/gold/},{synopsis: /dragon/}]}).pretty()
{
    "_id" : ObjectId("6107f370ed5b17447bdd707a"),
    "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 sprinted group of dwarves to reclaim their mountain home - and the gold with in it - from the dragon Smaug"
}
> db.movies.remove({title: "Pee Wee Herman's Big Adventure"})
WriteResult({ "nRemoved" : 0 })
> db.movies.remove({title: "Avatar"})
WriteResult({ "nRemoved" : 0 })
> db.users.insertMany([
... {username: "GoogGuyGrey",
... first_name: "Good Guy",
... last_name: "Grey",
... username: "ScumbagSteve",
... full_name: [ {first: "Scumbag"}, {last: "Steve"} ] }
...
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("610808a8ed5b17447bdd707f")
    ]
})
> db.collection.drop()
false
> show collections
movies
users
users
> db.users.remove()
uncaught exception: Error: remove needs a query :
DBCollection.prototype._parseRemove@src/mongo/shell/collection.js:364:15
DBCollection.prototype.remove@src/mongo/shell/collection.js:394:18
@(shell):1:1
> db.movies.find()
{
    "_id" : ObjectId("6107f370ed5b17447bdd7077"),
    "title" : "Fight Club",
    "writer" : "Chuck Palahniuk",
    "year" : 1999,
    "actors" : [ "Brad Pitt", "Edward Norton" ]
}
{
    "_id" : ObjectId("6107f370ed5b17447bdd7078"),
    "title" : "Pulp Fiction",
    "writer" : "Quentin Tarantino",
    "year" : 1994,
    "actors" : [ "Samuel L. Jackson", "John travolta", "Uma Thurman", "Morgan Freeman", "Tim Roth", "Bruce Campbell", "Michael Parks", "Ving Rhames", "Chris Bauer", "Lorraine Bracco", "Joey Pantaleo", "Alfred Molina", "Daryl Sabara", "Stephen Root", "Terry Crews", "Sam Rockwell", "Tim Blake Nelson", "Mike Quinn", "Jordan Peele", "Nate Parker", "Kurtwood Smith", "Jeffrey Wright", "Eduardo Capetillo", "Luis Guzman", "Carmen Ejogo", "Natalie Venetia Belcon", "Lorraine Bracco", "Joey Pantaleo", "Alfred Molina", "Daryl Sabara", "Stephen Root", "Terry Crews", "Sam Rockwell", "Tim Blake Nelson", "Mike Quinn", "Jordan Peele", "Nate Parker", "Kurtwood Smith", "Jeffrey Wright", "Eduardo Capetillo", "Luis Guzman", "Carmen Ejogo", "Natalie Venetia Belcon" ]
}

```

```
 Command Prompt - mongo
DBCollection.prototype.remove@src/mongo/shell/collection.js:394:18
@(shell):1:1
> db.movies.find()
{
  "_id" : ObjectId("6107f370ed5b17447bdd7077"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd7078"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "Samuel L. Jackson",
    "John Travolta",
    "Uma Thurman"
  ]
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd7079"),
  "title" : "Inglourious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd707a"),
  "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 sprinted group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug"
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd707b"),
  "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("6107f370ed5b17447bdd707c"),
  "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 array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness"
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd707d"),
  "title" : "Pee Wee Herman's Big Adventure"
},
{
  "_id" : ObjectId("6107f370ed5b17447bdd707e"),
  "title" : "Avatar"
}
> db.movies.remove({title: "Pee Wee Herman's Big Adventure"})
WriteResult({
  "nRemoved" : 1
})
> db.movies.remove({title: "Avatar"})
WriteResult({
  "nRemoved" : 1
})
> db.users.delete()
caught exception: TypeError: db.users.delete is not a function
@(shell):1:1
> db.users.drop()
true
> show collections
movies
> db.users.insertMany([
  {
    "username" : "GoodGuyGrey",
    "first_name" : "Good Guy",
    "last_name" : "Grey"
  }
])
...
> db.users.drop()
...
...
>
```

```
 Command Prompt - mongo
false
> show collections
movies
> db.users.insertMany([
  {
    "username" : "GoodGuyGrey",
    "first_name" : "Good Guy",
    "last_name" : "Grey",
    "full_name" : ["Good Guy", "Grey"]
  }
])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61080f59ed5b17447bdd7080"),
    ObjectId("61080f59ed5b17447bdd7081")
  ]
}
> db.posts.insertMany([
  {
    "username" : "GoodGuyGrey",
    "title" : "Passes out at party",
    "body" : "Wakes up early and cleans house",
    "full_name" : ["Good Guy", "Grey"]
  },
  {
    "username" : "GoodGuyGrey",
    "title" : "Steals your identity",
    "body" : "Raises your credit score",
    "full_name" : ["Good Guy", "Grey"]
  },
  {
    "username" : "GoodGuyGrey",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request",
    "full_name" : ["Good Guy", "Grey"]
  },
  {
    "username" : "ScumbagSteve",
    "title" : "Borrow something",
    "body" : "Sells it",
    "full_name" : ["Scumbag Steve", "Grey"]
  },
  {
    "username" : "Scumbag Steve",
    "title" : "Borrow everything",
    "body" : "The end",
    "full_name" : ["Scumbag Steve", "Grey"]
  },
  {
    "username" : "ScumbagSteve",
    "title" : "Forks your repo on github",
    "body" : "Sets to private",
    "full_name" : ["Scumbag Steve", "Grey"]
  }
])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61081270ed5b17447bdd7082"),
    ObjectId("61081270ed5b17447bdd7083")
  ]
}
```

```
PS Command Prompt - mongo
...
{
    "acknowledged": true,
    "insertedIds": [
        ObjectId("61081270ed5b17447bdd7082"),
        ObjectId("61081270ed5b17447bdd7083"),
        ObjectId("61081270ed5b17447bdd7084"),
        ObjectId("61081270ed5b17447bdd7085"),
        ObjectId("61081270ed5b17447bdd7086"),
        ObjectId("61081270ed5b17447bdd7087")
    ]
}
> db.comments.insertMany([
... {username: "GoodGuyGrey",
... comment: "Hope you got a good deal!",
... post: ObjectId("61081270ed5b17447bdd7085")},
... {username: "GoodGuyGrey",
... comment: "What's mine is yours!",
... post: ObjectId("61081270ed5b17447bdd7086")},
... {username: "GoodGuyGrey",
... comment: "Don't violate the licensing agreement!",
... post: ObjectId("61081270ed5b17447bdd7087")},
... {username: "ScumbagSteve",
... comment: "It still isn't clean",
... post: ObjectId("61081270ed5b17447bdd7082")},
... {username: "ScumbagSteve",
...
uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):14:25
@ show collections
movies
posts
users
> db.comments.insertMany([
... {username: "GoodGuyGrey",
... comment: "Hope you got a good deal!",
... post: ObjectId("61081270ed5b17447bdd7085")},
... {username: "
...
uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):5:12
> db.comments.insertMany([
... {username: "GoodGuyGrey",
... comment: "Hope you got a good deal!",
... post: ObjectId("61081270ed5b17447bdd7085")}]
```

```
 Command Prompt - mongo
@shell:3:14
> db.comments.insertMany([
... {username: "GoodGuyGrey",
... comment: "Hope you got a good deal!",
... post: ObjectId("61081270ed5b17447bdd7085")},
... {username: "GoodGuyGrey",
... comment: "What's mine is yours!",
... post: ObjectId("61081270ed5b17447bdd7086")},
... {username: "GoodGuyGrey",
... comment: "Don't violate the licensing agreement!",
... post: ObjectId("61081270ed5b17447bdd7087")},
... {username: "ScumbagSteve",
... comment: "It still isn't clean",
... post: ObjectId("61081270ed5b17447bdd7082")},
... {username: "ScumbagSteve",
... comment: "Denied your PR cause I found a hack",
... post: ObjectId("61081270ed5b17447bdd7084")}
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61081ccced5b17447bdd7088"),
    ObjectId("61081ccced5b17447bdd7089"),
    ObjectId("61081ccced5b17447bdd708a"),
    ObjectId("61081ccced5b17447bdd708b"),
    ObjectId("61081ccced5b17447bdd708c")
  ]
}
> db.users.find().pretty()
{
  "_id" : ObjectId("61080f59ed5b17447bdd7080"),
  "username" : "GoodGuyGrey",
  "first_name" : "Good Guy",
  "last_name" : "Grey"
}
{
  "_id" : ObjectId("61080f59ed5b17447bdd7081"),
  "username" : "ScumbagSteve",
  "full_name" : [
    {
      "first" : "Scumbag"
    }
  ]
}
```

```
 Command Prompt - mongo
{
    "first" : "Scumbag",
},
{
    "last" : "Steve"
}
}
> db.posts.find().pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7082"),
    "username" : "GoodGuyGrey",
    "title" : "Passes out at party",
    "body" : "Wakes up early and cleans house"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7083"),
    "username" : "GoodGuyGrey",
    "title" : "Steals your identity",
    "body" : "Raises your credit score"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7084"),
    "username" : "GoodGuyGrey",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7085"),
    "username" : "ScumbagSteve",
    "title" : "Borrows something",
    "body" : "Sells it"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7086"),
    "username" : "Scumbag Steve",
    "title" : "Borrows everything",
    "body" : "The end"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7087"),
}

```

```
 Command Prompt - mongo
{
    "_id" : ObjectId("61081270ed5b17447bdd7087"),
    "username" : "ScumbagSteve",
    "title" : "Forks your repo ob github",
    "body" : "Sets to private"
}
> db.posts.find({username: "GoodGuyGrey"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7082"),
    "username" : "GoodGuyGrey",
    "title" : "Passes out at party",
    "body" : "Wakes up early and cleans house"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7083"),
    "username" : "GoodGuyGrey",
    "title" : "Steals your identity",
    "body" : "Raises your credit score"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7084"),
    "username" : "GoodGuyGrey",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request"
}
> db.posts.find({username: "ScumbagSteve"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7085"),
    "username" : "ScumbagSteve",
    "title" : "Borrows something",
    "body" : "Sells it"
}
{
    "_id" : ObjectId("61081270ed5b17447bdd7086"),
    "username" : "ScumbagSteve",
    "title" : "Forks your repo ob github",
    "body" : "Sets to private"
}
> db.posts.find({username: "Scumbag Steve"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7087"),
}

```

```
PS Command Prompt - mongo
{
    "body" : "Sets to private"
}
> db.posts.find({username: "Scumbag Steve"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7086"),
    "username" : "Scumbag Steve",
    "title" : "Borrows everything",
    "body" : "The end"
}
> db.comments.find()
{
    "_id" : ObjectId("61081270ed5b17447bdd7088"),
    "username" : "GoodGuyGrey",
    "comment" : "Hope you got a good deal!",
    "post" : ObjectId("61081270ed5b17447bdd7085")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd7089"),
    "username" : "GoodGuyGrey",
    "comment" : "What's mine is yours!",
    "post" : ObjectId("61081270ed5b17447bdd7086")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708a"),
    "username" : "GoodGuyGrey",
    "comment" : "Don't violate the licensing agreement!",
    "post" : ObjectId("61081270ed5b17447bdd7087")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708b"),
    "username" : "ScumbagSteve",
    "comment" : "It still isn't clean",
    "post" : ObjectId("61081270ed5b17447bdd7082")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708c"),
    "username" : "ScumbagSteve",
    "comment" : "Denied your PR cause I found a hack",
    "post" : ObjectId("61081270ed5b17447bdd7084")
}
> db.comments.find({username: "GoodGuyGrey"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7088"),
    "username" : "GoodGuyGrey",
    "comment" : "Hope you got a good deal!",
    "post" : ObjectId("61081270ed5b17447bdd7085")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd7089"),
    "username" : "GoodGuyGrey",
    "comment" : "What's mine is yours!",
    "post" : ObjectId("61081270ed5b17447bdd7086")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708a"),
    "username" : "GoodGuyGrey",
    "comment" : "Don't violate the licensing agreement!",
    "post" : ObjectId("61081270ed5b17447bdd7087")
}
> db.comments.find({username: "ScumbagSteve"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd708b"),
    "username" : "ScumbagSteve",
    "comment" : "It still isn't clean",
    "post" : ObjectId("61081270ed5b17447bdd7082")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708c"),
    "username" : "ScumbagSteve",
    "comment" : "Denied your PR cause I found a hack",
    "post" : ObjectId("61081270ed5b17447bdd7084")
}
```

```
PS Command Prompt - mongo
{
    "comment" : "What's mine is yours!",
    "post" : ObjectId("61081270ed5b17447bdd7086")
}
{
    "_id" : ObjectId("61081270ed5b17447bdd708a"),
    "username" : "GoodGuyGrey",
    "comment" : "Don't violate the licensing agreement!",
    "post" : ObjectId("61081270ed5b17447bdd7087")
}
db.comments.find({username: "ScumbagSteve"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd708b"),
    "username" : "ScumbagSteve",
    "comment" : "It still isn't clean",
    "post" : ObjectId("61081270ed5b17447bdd7082")
},
{
    "_id" : ObjectId("61081270ed5b17447bdd708c"),
    "username" : "ScumbagSteve",
    "comment" : "Denied your PR cause I found a hack",
    "post" : ObjectId("61081270ed5b17447bdd7084")
}
> db.comments.find({post: "Reports a bug in your code"}).pretty()
> db.comments.find({title: "Reports a bug in your code"}).pretty()
> db.posts.find({title: "Reports a bug in your code"}).pretty()
{
    "_id" : ObjectId("61081270ed5b17447bdd7084"),
    "username" : "GoodGuyGrey",
    "title" : "Reports a bug in your code",
    "body" : "Sends you a Pull Request"
}
>
```