

MongoDB

Assignment-1

```
C:\Windows\System32\cmd.exe - mongo
> db
test
> use mongo_practice
switched to db mongo_practice
> use movies
switched to db movies
> db
movies
> db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"2009", actors:["John Travolta", "Uma Thurman"]})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkien", year:"2012",franchise:"The Hobbit"})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R. Tolkien", year:"2013", franchise:"The Hobbit"})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R. Tolkien", year:"2002", 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."})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Pee Wee Herman's Big Adventures"})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Avatar"})
WriteResult({ "nInserted" : 1 })
> db.movies.find()
{ "_id" : ObjectId("61e7c42a0faa042e333801ee"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }
{ "_id" : ObjectId("61e7c4400faa042e333801ef"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "John Travolta", "Uma Thurman" ] }
{ "_id" : ObjectId("61e7c44e0faa042e333801f0"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }
{ "_id" : ObjectId("61e7c4810faa042e333801f1"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkien", "year" : "2012", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : "2013", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("61e7c4bd0faa042e333801f3"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkien", "year" : "2002", "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("61e7c4ca0faa042e333801f4"), "title" : "Pee Wee Herman's Big Adventures" }
{ "_id" : ObjectId("61e7c4d40faa042e333801f5"), "title" : "Avatar" }
> db.movies.find({writer:"Quentin Tarantino"})
{ "_id" : ObjectId("61e7c4400faa042e333801ef"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "John Travolta", "Uma Thurman" ] }
{ "_id" : ObjectId("61e7c44e0faa042e333801f0"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }
```

```
Select C:\Windows\System32\cmd.exe - mongo
> db.movies.find({franchise:"The Hobbit"})
{ "id" : ObjectId("61e7c42e0faa042e333801f1"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkien", "year" : "2012", "franchise" : "The Hobbit" }
{ "id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : "2013", "franchise" : "The Hobbit" }
{ "id" : ObjectId("61e7c4bd0faa042e333801f3"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkien", "year" : "2002", "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("61e7c42e0faa042e333801f1"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }
> db.movies.find({ $or: [{ year: { $gt: "2010" } }, { year: { $lt: "2000" } } ] })
{ "id" : ObjectId("61e7c42e0faa042e333801f1"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }
{ "id" : ObjectId("61e7c4810faa042e333801f1"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkien", "year" : "2012", "franchise" : "The Hobbit" }
{ "id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : "2013", "franchise" : "The Hobbit" }
> db.movies.update({ _id:ObjectId("61e7c4810faa042e333801f1") }, { $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."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.update({ _id:ObjectId("61e7c4400faa042e333801ef") }, { $push:{actors:"Samuel L. Jackson"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.update({ _id:ObjectId("61e7c49a0faa042e333801f2") }, { $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."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.find({ synopsis:{ $regex:"Bilbo" }})
{ "id" : ObjectId("61e7c4810faa042e333801f1"), "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 spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }
{ "id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "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("61e7c4bd0faa042e333801f3"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkien", "year" : "2002", "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({ synopsis:{ $regex:"Gandalf" }})
{ "id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }
> db.movies.find({ $and: [{ synopsis: { $regex: "Bilbo" } }, { synopsis: { $not: /Gandalf/ } } ] })
{ "id" : ObjectId("61e7c4810faa042e333801f1"), "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 spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }
{ "id" : ObjectId("61e7c4bd0faa042e333801f3"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R. Tolkien", "year" : "2002", "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 15°C Partly sunny ENG 14:51 US 19-01-2022
```

```
Select C:\Windows\System32\cmd.exe - mongo
it", "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({ $or: [{ synopsis: { $regex: "dwarves" } }, { synopsis: { $regex: "hobbit" } } ] })
{ "id" : ObjectId("61e7c4810faa042e333801f1"), "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 spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }
{ "id" : ObjectId("61e7c49a0faa042e333801f2"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R. Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }
> db.movies.find({ $and: [{ synopsis: { $regex: "gold" } }, { synopsis: { $regex: "dragon" } } ] })
{ "id" : ObjectId("61e7c4810faa042e333801f1"), "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 spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }
> db.movies.remove({ _id:ObjectId("61e7c4ca0faa042e333801f4") })
WriteResult({ "nRemoved" : 1 })
> db.movies.remove({ _id:ObjectId("61e7c4d40faa042e333801f5") })
WriteResult({ "nRemoved" : 1 })
> db.users.insert({ _id:1, username:"GoodGuyGreg", first_name:"Good Guy", last_name:"Greg" })
WriteResult({ "nInserted" : 1 })
> db.users.insert({ _id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score" })
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score" })
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request" })
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it" })
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end" })
WriteResult({ "nInserted" : 1 })
> db.posts.insert({ username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private" })
WriteResult({ "nInserted" : 1 })
> db.posts.find()
{ "id" : ObjectId("61e7c9340faa042e333801f6"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }
{ "id" : ObjectId("61e7c94a0faa042e333801f7"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "id" : ObjectId("61e7c9590faa042e333801f8"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }
{ "id" : ObjectId("61e7c9690faa042e333801f9"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }
{ "id" : ObjectId("61e7c9700faa042e333801fa"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }
{ "id" : ObjectId("61e7c9810faa042e333801fb"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }
> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("61e7c9690faa042e333801f9") })
Type here to search 15°C Partly sunny ENG 14:52 US 19-01-2022
```

```
Select C:\Windows\System32\cmd.exe - mongo
> db.comments.insert({ username: "GoodGuyGreg", comment: "Hope you got a good deal!", post: ObjectId("61e7c9690faa042e333801f9") })
WriteResult({ "nInserted" : 1 })
> db.comments.insert({ username: "GoodGuyGreg", comment: "What's mine is yours!", post: ObjectId("61e7c9760faa042e333801fa") })
WriteResult({ "nInserted" : 1 })
> db.comments.insert({ username: "GoodGuyGreg", comment: "Don't violate the licensing agreement!", post: ObjectId("61e7c9810faa042e333801fb") })
WriteResult({ "nInserted" : 1 })
> db.comments.insert({ username: "ScumbagSteve", comment: "It still isn't clean", post: ObjectId("61e7c9340faa042e333801f6") })
WriteResult({ "nInserted" : 1 })
> db.comments.insert({ username: "ScumbagSteve", comment: "Denied your PR cause I found a hack", post: ObjectId("61e7c9590faa042e333801f8") })
WriteResult({ "nInserted" : 1 })
> db.users.find().pretty()
{
  "_id" : 1,
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : 2,
  "username" : "ScumbagSteve",
  "full_name" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
> db.posts.find().pretty()
{
  "_id" : ObjectId("61e7c9340faa042e333801f6"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at Party",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("61e7c94a0faa042e333801f7"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("61e7c9590faa042e333801f8"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a pull request"
}
{
  "_id" : ObjectId("61e7c9690faa042e333801f9"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("61e7c9760faa042e333801fa"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("61e7c9810faa042e333801fb"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
> db.posts.find({ username: "GoodGuyGreg" })
{ "_id" : ObjectId("61e7c9340faa042e333801f6"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }
{ "_id" : ObjectId("61e7c94a0faa042e333801f7"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("61e7c9590faa042e333801f8"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }
> db.posts.find({ username: "ScumbagSteve" })
{ "_id" : ObjectId("61e7c9690faa042e333801f9"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("61e7c9760faa042e333801fa"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("61e7c9810faa042e333801fb"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }
> db.comments.find().pretty()
{
  "_id" : ObjectId("61e7d0990faa042e333801fc"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("61e7c9690faa042e333801f9")
}
{
  "_id" : ObjectId("61e7d1330faa042e333801fd"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("61e7c9760faa042e333801fa")
}
```

```
Select C:\Windows\System32\cmd.exe - mongo
{
  "_id" : ObjectId("61e7d0990faa042e333801fc"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("61e7c9690faa042e333801f9")
}
{
  "_id" : ObjectId("61e7d1330faa042e333801fd"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("61e7c9760faa042e333801fa")
}
```



```
Select C:\Windows\System32\cmd.exe - mongo
"comment" : "What's mine is yours!",
"post" : ObjectId("61e7c9760faa042e333801fa")
}
{
  "_id" : ObjectId("61e7d1940faa042e333801fe"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : ObjectId("61e7c9810faa042e333801fb")
}
{
  "_id" : ObjectId("61e7d1e80faa042e333801ff"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : ObjectId("61e7c9340faa042e333801f6")
}
{
  "_id" : ObjectId("61e7d2370faa042e33380200"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("61e7c9590faa042e333801f8")
}
> db.comments.find({username:"GoodGuyGreg"})
{ "_id" : ObjectId("61e7d0990faa042e333801fc"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : ObjectId("61e7c9690faa042e333801f9") }
{ "_id" : ObjectId("61e7d1330faa042e333801fd"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!", "post" : ObjectId("61e7c9760faa042e333801fa") }
{ "_id" : ObjectId("61e7d1940faa042e333801fe"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : ObjectId("61e7c9810faa042e333801fb") }
> db.comments.find({username:"ScumbagSteve"})
{ "_id" : ObjectId("61e7d1e80faa042e333801ff"), "username" : "ScumbagSteve", "comment" : "It still isn't clean", "post" : ObjectId("61e7c9340faa042e333801f6") }
{ "_id" : ObjectId("61e7d2370faa042e33380200"), "username" : "ScumbagSteve", "comment" : "Denied your PR cause I found a hack", "post" : ObjectId("61e7c9590faa042e333801f8") }
> db.comments.find({post:ObjectId("61e7c9590faa042e333801f8")}).pretty
function() {
  this.__prettyShell = true;
  return this;
}
>
```

Windows taskbar: Type here to search, 15°C Partly sunny, 14:52, 19-01-2022