

MongoDB Exercise in mongo shell - Solutions

Connect to a running mongo instance, use a database named **mongo_practice**.

Insert Documents

```
C:\Windows\System32\cmd.exe - mongo
> use movies_practice
switched to db movies_practice
> db.createCollection('movies')
{ "ok" : 1 }
> db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuko", year: "1999", actors:["Brad Pitt", "Edward Norton"]})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"1994", 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:"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."})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Pee Wee Herman's Big Adventure"})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:"Avatar"})
WriteResult({ "nInserted" : 1 })
```

Query / Find Documents

Query the **movies** collection to

1. get all documents

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find().pretty()
{
  "_id" : ObjectId("60a54e92db6a5c3856946a5a"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : "1999",
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5b"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : "1994",
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("60a54f12db6a5c3856946a5c"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : "2009",
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
```

```

C:\Windows\System32\cmd.exe - mongo
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkien",
  "year" : "2012",
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkien",
  "year" : "2013",
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("60a55023db6a5c3856946a5f"),
  "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."
}
{
  "_id" : ObjectId("60a55056db6a5c3856946a60"),
  "title" : "Pee Wee Herman's Big Adventure"
}
{ "_id" : ObjectId("60a55082db6a5c3856946a61"), "title" : "Avatar" }

```

2. get all documents with writer set to "Quentin Tarantino"

```

C:\Windows\System32\cmd.exe - mongo
> db.movies.find({writer:"Quentin Tarantino"}).pretty()
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5b"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : "1994",
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("60a54f12db6a5c3856946a5c"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : "2009",
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

```

3. get all documents where actors include "Brad Pitt"

```

C:\Windows\System32\cmd.exe - mongo
> db.movies.find({actors:"Brad Pitt"}).pretty()
{
  "_id" : ObjectId("60a54e92db6a5c3856946a5a"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : "1999",
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("60a54f12db6a5c3856946a5c"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : "2009",
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}

```

4. get all documents with franchise set to "The Hobbit"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({franchise:"The Hobbit"}).pretty()
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkien",
  "year" : "2013",
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("60a55023db6a5c3856946a5f"),
  "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."
}
```

5. get all movies released in the 90s

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({year:{$gt:"1990", $lt:"2000"}}).pretty()
{
  "_id" : ObjectId("60a54e92db6a5c3856946a5a"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : "1999",
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5b"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : "1994",
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
```

6. get all movies released before the year 2000 or after 2010

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]}).pretty()
{
  "_id" : ObjectId("60a54e92db6a5c3856946a5a"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : "1999",
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5b"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : "1994",
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit"
}
```

```

C:\Windows\System32\cmd.exe - mongo
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R Tolkien",
  "year" : "2013",
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("60a55023db6a5c3856946a5f"),
  "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."
}

```

Update Documents

1. add a synopsis to "The Hobbit: An Unexpected Journey" : "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."

```

C:\Windows\System32\cmd.exe - mongo
> db.movies.update({"title" : "The Hobbit: An unexpected Journey"}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

2. add a synopsis to "The Hobbit: The Desolation of Smaug" : "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."

```

C:\Windows\System32\cmd.exe - mongo
> db.movies.update({"title" : "The Hobbit: The Desolation of Smaug"}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

```

C:\Windows\System32\cmd.exe - mongo
> db.movies.update({"title" : "Pulp Fiction"}, {$push:{actors:"Samuel L. Jackson"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

To view the changes made in **movies** collection

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find().pretty()
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5a"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : "1999",
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("60a54ee4db6a5c3856946a5b"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : "1994",
  "actors" : [
    "John Travolta",
    "Uma Thurman",
    "Samuel L. Jackson"
  ]
}
{
  "_id" : ObjectId("60a54f12db6a5c3856946a5c"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : "2009",
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
```

```
C:\Windows\System32\cmd.exe - mongo
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "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("60a54fb5db6a5c3856946a5e"),
  "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("60a55023db6a5c3856946a5f"),
  "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."
}
{
  "_id" : ObjectId("60a55056db6a5c3856946a60"),
  "title" : "Pee Wee Herman's Big Adventure"
}
{ "_id" : ObjectId("60a55082db6a5c3856946a61"), "title" : "Avatar" }
```

Text Search

1. find all movies that have a synopsis that contains the word "Bilbo"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({synopsis:{$regex:"Bilbo"}}).pretty()
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."
}
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "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("60a55023db6a5c3856946a5f"),
  "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."
}
```

2. find all movies that have a synopsis that contains the word "Gandalf"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({synopsis:{$regex:"Gandalf"}}).pretty()
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "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."
}
```

3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}]]}).pretty()
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."
}
{
  "_id" : ObjectId("60a55023db6a5c3856946a5f"),
  "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."
}
```

4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({$or:[{synopsis:{regex:"dwarves"}}, {synopsis:{regex:"hobbit"}}]}).pretty()
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim t
  heir mountain home - and the gold within it - from the dragon Smaug."
}
{
  "_id" : ObjectId("60a54fb5db6a5c3856946a5e"),
  "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 homela
  nd, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
```

5. find all movies that have a synopsis that contains the word "gold" and "dragon"

```
C:\Windows\System32\cmd.exe - mongo
> db.movies.find({$and:[{synopsis:{regex:"gold"}}, {synopsis:{regex:"dragon"}}]}).pretty()
{
  "_id" : ObjectId("60a54f60db6a5c3856946a5d"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : "2012",
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim t
  heir mountain home - and the gold within it - from the dragon Smaug."
}
```

Delete Documents

1. delete the movie "Pee Wee Herman's Big Adventure"

```
> db.movies.find({"title": "Pee Wee Herman's Big Adventure"}).pretty()
{
  "_id" : ObjectId("60a55056db6a5c3856946a60"),
  "title" : "Pee Wee Herman's Big Adventure"
}
> db.movies.remove({"title": "Pee Wee Herman's Big Adventure"})
WriteResult({"nRemoved" : 1 })
```

2. delete the movie "Avatar"

```
> db.movies.find({"title": "Avatar"}).pretty()
{ "_id" : ObjectId("60a55082db6a5c3856946a61"), "title" : "Avatar" }
> db.movies.remove({"title": "Avatar"})
WriteResult({"nRemoved" : 1 })
```

Relationships

Insert the following documents into a **users** collection

```
C:\Windows\System32\cmd.exe - mongo
> db.users.insert({username:"GoodGuyGreg", first_name:"Good Guy", last_name:"Greg"})
WriteResult({ "nInserted" : 1 })
> db.users.insert({username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})
WriteResult({ "nInserted" : 1 })
```

Insert the following documents into a **posts** collection

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Wakes up early and cleans house"})
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 })
```

Insert the following documents into a **comments** collection

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title":"Borrows something"}).pretty()
{
  "_id" : ObjectId("60a5ad53f6127945dd426c5a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("60a5ad53f6127945dd426c5a")})
WriteResult({ "nInserted" : 1 })
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title":"Borrows everything"}).pretty()
{
  "_id" : ObjectId("60a5ad5cf6127945dd426c5b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
> db.comments.insert({ username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("60a5ad5cf6127945dd426c5b")})
WriteResult({ "nInserted" : 1 })
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title":"Forks your repo on github"}).pretty()
{
  "_id" : ObjectId("60a5ad6cf6127945dd426c5c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
> db.comments.insert({ username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("60a5ad6cf6127945dd426c5c")})
WriteResult({ "nInserted" : 1 })
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title" : "Passes out at Party"}).pretty()
{
  "_id" : ObjectId("60a5acc7f6127945dd426c57"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at Party",
  "body" : "Wakes up early and cleans house"
}
> db.comments.insert({ username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("60a5acc7f6127945dd426c57")})
WriteResult({ "nInserted" : 1 })
```



```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title" : "Reports a bug in your code"}).pretty()
{
  "_id" : ObjectId("60a5acf6f6127945dd426c59"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
> db.comments.insert({ username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("60a5acf6f6127945dd426c59")
WriteResult({ "nInserted" : 1 })
```

Querying related collections

1. find all users

```
C:\Windows\System32\cmd.exe - mongo
> db.users.find().pretty()
{
  "_id" : ObjectId("60a5aa3df6127945dd426c55"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : ObjectId("60a5aa73f6127945dd426c56"),
  "username" : "ScumbagSteve",
  "fullname" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
```

2. find all posts

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find().pretty()
{
  "_id" : ObjectId("60a5acc7f6127945dd426c57"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at Party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("60a5acd9f6127945dd426c58"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("60a5acf6f6127945dd426c59"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
{
  "_id" : ObjectId("60a5ad53f6127945dd426c5a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("60a5ad5cf6127945dd426c5b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("60a5ad6cf6127945dd426c5c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
```

3. find all posts that was authored by "GoodGuyGreg"

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"username" : "GoodGuyGreg"}).pretty()
{
  "_id" : ObjectId("60a5acc7f6127945dd426c57"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at Party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("60a5acd9f6127945dd426c58"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("60a5acf6f6127945dd426c59"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
```

4. find all posts that was authored by "ScumbagSteve"

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"username" : "ScumbagSteve"}).pretty()
{
  "_id" : ObjectId("60a5ad53f6127945dd426c5a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("60a5ad5cf6127945dd426c5b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("60a5ad6cf6127945dd426c5c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
```

5. find all comments

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.find().pretty()
{
  "_id" : ObjectId("60a5af5cf6127945dd426c5d"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("60a5ad53f6127945dd426c5a")
}
{
  "_id" : ObjectId("60a5afe4f6127945dd426c5e"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("60a5ad5cf6127945dd426c5b")
}
{
  "_id" : ObjectId("60a5b068f6127945dd426c5f"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : ObjectId("60a5ad6cf6127945dd426c5c")
}
{
  "_id" : ObjectId("60a5b131f6127945dd426c60"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : ObjectId("60a5acc7f6127945dd426c57")
}
{
  "_id" : ObjectId("60a5b1b2f6127945dd426c61"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("60a5acf6f6127945dd426c59")
}
```

6. find all comments that was authored by "GoodGuyGreg"

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.find({"username" : "GoodGuyGreg"}).pretty()
{
  "_id" : ObjectId("60a5af5cf6127945dd426c5d"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("60a5ad53f6127945dd426c5a")
}
{
  "_id" : ObjectId("60a5afe4f6127945dd426c5e"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("60a5ad5cf6127945dd426c5b")
}
{
  "_id" : ObjectId("60a5b068f6127945dd426c5f"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : ObjectId("60a5ad6cf6127945dd426c5c")
}
```

7. find all comments that was authored by "ScumbagSteve"

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.find({"username" : "ScumbagSteve"}).pretty()
{
  "_id" : ObjectId("60a5b131f6127945dd426c60"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : ObjectId("60a5acc7f6127945dd426c57")
}
{
  "_id" : ObjectId("60a5b1b2f6127945dd426c61"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("60a5acf6f6127945dd426c59")
}
```

8. find all comments belonging to the post "Reports a bug in your code"

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({"title" : "Reports a bug in your code"}).pretty()
{
  "_id" : ObjectId("60a5acf6f6127945dd426c59"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
> db.comments.find({"post" : ObjectId("60a5acf6f6127945dd426c59")}).pretty()
{
  "_id" : ObjectId("60a5b1b2f6127945dd426c61"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("60a5acf6f6127945dd426c59")
}
```