

# MongoDB Lab Assignments – Day 1

## MongoDB Exercise in mongo shell

### Inserting Documents

```
C:\Windows\System32\cmd.exe - mongo
> use mongo_practice
switched to db mongo_practice
> db.collection.insert([
... {
...   title : "Fight Club",
...   writer : "Chuck Palahniuk",
...   year : 1999,
...   actors : [
...     "Brad Pitt",
...     "Edward Norton"
...   ]},
...
... {
...   title : "Pulp Fiction",
...   writer : "Quentin Tarantino",
...   year : 1994,
...   actors : [
...     "John Travolta",
...     "Uma Thurman"
...   ]},
...
... {
...   title : "Inglorious 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"
... },
... ])
```

```
C:\Windows\System32\cmd.exe - mongo
... {
... 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 ris
ing darkness."
... },
...
... {
... title : "Pee Wee Herman's Big Adventure"
... }
... ,
...
... {
... title : "Avatar"
... }
... ]
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 8,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

## Query/Find Documents

### 1. Get all documents

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find().pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9d"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
```

```
C:\Windows\System32\cmd.exe - mongo
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aaa0"),
  "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 fallung into the hands of
a rising darkness."
}
{
  "_id" : ObjectId("61067bfb0236933439c1aaa1"),
  "title" : "Pee Wee Herman's Big Adventure"
}
{ "_id" : ObjectId("61067bfb0236933439c1aaa2"), "title" : "Avatar" }
```

## 2. Get all documents with writer set to “Quentin Tarantino”

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find({writer : "Quentin Tarantino"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9d"),
  "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.collection.find({actors : "Brad Pitt"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9d"),
  "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.collection.find({franchise : "The Hobbit"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aaa0"),
  "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."
}
```

#### 5. Get all movies released in the 90s

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find({year: {$gte:1990, $lt:2000}}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9c"),
  "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.collection.find({$or:[{year:{$gt:2010}},{year:{$lt:2000}}]}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9b"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("61067bfb0236933439c1aaa0"),
  "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 c
ombatants and keep the Lonely Mountain from falling into the hands of a rising darkness."
}
```

## Update Documents

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.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 })
> db.collection.find({title : "The Hobbit: An Unexpected Journey"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "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."
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.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, Thier homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.collection.find({title : "The Hobbit: The Desolation of Smaug"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "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, Thier homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.update({"title":"Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.collection.find({title : "Pulp Fiction"}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9c"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman",
    "Samuel L. Jackson"
  ]
}
```



## Text Search

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find({synopsis:/Bilbo./}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "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("61067bfb0236933439c1aa9f"),
  "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, Thier homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
> db.collection.find({synopsis:/Gandalf./}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "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, Thier homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find({$and: [{synopsis:/Bilbo/},{synopsis:{$not:/Gandalf/}]}]).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "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("61067bfb0236933439c1aaa0"),
  "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."
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.find({$or: [{synopsis:/dwarves/},{synopsis:/hobbit/}]}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mo
utain with a spirited group of dwarves to reclaim their mountain home- and the gol
d within it- from the dragon Smaug."
}
{
  "_id" : ObjectId("61067bfb0236933439c1aa9f"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit",
  "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the grey,
continue their quest to reclaim Erebor, Thier homeland, from Smaug. Bilbo Baggins
is in possession of a mysterious and magical ring."
}
>
>
>
>
> db.collection.find({$and: [{synopsis:/gold/},{synopsis:/dragon/}]}).pretty()
{
  "_id" : ObjectId("61067bfb0236933439c1aa9e"),
  "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 Mo
utain with a spirited group of dwarves to reclaim their mountain home- and the gol
d within it- from the dragon Smaug."
}
```

## Delete Documents

```
C:\Windows\System32\cmd.exe - mongo
> db.collection.remove({title:"Pee Wee Herman's Big Adventure"})
WriteResult({ "nRemoved" : 1 })
> db.collection.remove({title:"Avatar"})
WriteResult({ "nRemoved" : 1 })
> db.collection.find({title:"Avatar"})
> db.collection.find({title:"Pee Wee Herman's Big Adventure"})
```

## Relationships

```
C:\Windows\System32\cmd.exe - mongo
> db.users.insert([ { username : "GoodGuyGreg", first_name : "Good Guy", last_name
: "Greg" }, { username : "ScumbagSteve", full_name: {first: "Scumbag", last : "St
eve" } } ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
> db.users.find().pretty()
{
  "_id" : ObjectId("61069c5d0236933439c1aaa5"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : ObjectId("61069c5d0236933439c1aaa6"),
  "username" : "ScumbagSteve",
  "full_name" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
```

C:\Windows\System32\cmd.exe - mongo

```
> db.posts.insert([
... {
...   _id : 1,
...   username: "GoodGuyGerg",
...   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 : "GoodGuyGerg",
...   title: "Reports a bug in your code",
...   body: "Sends you a Pull Request"
... },
... {
...   _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 to private"
... }
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 6,
  "nUpserted" : 0,
  "nMatched" : 0,
```

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.insert([
... {
...   username : "GoodGuyGreg",
...   comment : "Hope you got a good deal!",
...   post : 4
... },
... {
...   username : "GoodGuyGreg"
...   , comment : "What's mine is yours!",
...   post : 5
... },
... {
...   username : "GoodGuyGreg",
...   comment : "Don't violate the licensing agreement!",
...   post: 6
... },
... {
...   username : "ScumbagSteve",
...   comment : "It still isn't clean",
...   post : 1
... },
... {
...   username : "ScumbagSteve",
...   comment : "Denied your PR cause I found a hack",
...   post : 3
... }
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 5,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

## Querying related collections

```
C:\Windows\System32\cmd.exe - mongo
> db.users.find().pretty()
{
  "_id" : ObjectId("61069c5d0236933439c1aaa5"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : ObjectId("61069c5d0236933439c1aaa6"),
  "username" : "ScumbagSteve",
  "full_name" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find().pretty()
{
  "_id" : 1,
  "username" : "GoodGuyGerg",
  "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" : "GoodGuyGerg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
{
  "_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 to private"
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({$or:[{username: "GoodGuyGreg"},{username:"GoodGuyGerg"}]}).pretty
{
  "_id" : 1,
  "username" : "GoodGuyGerg",
  "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" : "GoodGuyGerg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
```

```
C:\Windows\System32\cmd.exe - mongo
> 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 to private"
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.find().pretty()
{
  "_id" : ObjectId("6106a36a0236933439c1aaa7"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : 4
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaa8"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : 5
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaa9"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : 6
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaaa"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : 1
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaab"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : 3
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.comments.find({username: "GoodGuyGreg"}).pretty()
{
  "_id" : ObjectId("6106a36a0236933439c1aaa7"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : 4
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaa8"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : 5
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaa9"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : 6
}
> db.comments.find({username: "ScumbagSteve"}).pretty()
{
  "_id" : ObjectId("6106a36a0236933439c1aaaa"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : 1
}
{
  "_id" : ObjectId("6106a36a0236933439c1aaab"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : 3
}
```

```
C:\Windows\System32\cmd.exe - mongo
> db.posts.find({title:"Reports a bug in your code"}).pretty()
{
  "_id" : 3,
  "username" : "GoodGuyGerg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
> db.comments.find({post:3}).pretty()
{
  "_id" : ObjectId("6106a36a0236933439c1aaab"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : 3
}
```