

MongoDB Assignment 1

```
Command Prompt - mongo
> use mongo_practice
switched to db mongo_practice
> show dbs;
admin    0.000GB
config  0.000GB
local    0.000GB
> db.create
db.createCollection( db.createRole( db.createUser( db.createView(
> db.createCollection("movie")
{ "ok" : 1 }
> show dbs;
admin          0.000GB
config         0.000GB
local          0.000GB
mongo_practice 0.000GB
> show collections
movie
>
```

```
Select Command Prompt - mongo
> db.movies.insertMany([
... {
...   title:"Fight Club",
...   writer:"Chuck Palahniuko",
...   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"
... },
... {
...   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 darkne
ss."
... },
... {
...   title:"Pee Wee Herman's Big Adventure"
... },
... {
...   title:"Avatar"
... }
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6110eb8c54966d89050945f1"),
    ObjectId("6110eb8c54966d89050945f2"),
    ObjectId("6110eb8c54966d89050945f3"),
    ObjectId("6110eb8c54966d89050945f4"),
    ObjectId("6110eb8c54966d89050945f5"),
    ObjectId("6110eb8c54966d89050945f6"),
    ObjectId("6110eb8c54966d89050945f7"),
    ObjectId("6110eb8c54966d89050945f8")
  ]
}
```

```
Select Command Prompt - mongo
... 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 darkne
ss."
... },
... {
...   title:"Pee Wee Herman's Big Adventure"
... },
... {
...   title:"Avatar"
... }
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6110eb8c54966d89050945f1"),
    ObjectId("6110eb8c54966d89050945f2"),
    ObjectId("6110eb8c54966d89050945f3"),
    ObjectId("6110eb8c54966d89050945f4"),
    ObjectId("6110eb8c54966d89050945f5"),
    ObjectId("6110eb8c54966d89050945f6"),
    ObjectId("6110eb8c54966d89050945f7"),
    ObjectId("6110eb8c54966d89050945f8")
  ]
}
```

Query / Find Documents

1. Get all documents

```
> db.movies.find().pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f1"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f2"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f3"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f4"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "title" : "The Hobbit: The Desolation of Smaug",
```

```
    "_id" : ObjectId("6110eb8c54966d89050945f5"),
    "title" : "The Hobbit: The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit"
  }
  {
    "_id" : ObjectId("6110eb8c54966d89050945f6"),
    "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("6110eb8c54966d89050945f7"),
    "title" : "Pee Wee Herman's Big Adventure"
  }
  { "_id" : ObjectId("6110eb8c54966d89050945f8"), "title" : "Avatar" }
}
>
```

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

```
> db.movies.find({writer:"Quentin Tarantino"}).pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f2"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f3"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
>
```

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

```
> db.movies.find({actors:"Brad Pitt"}).pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f1"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f3"),
  "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"

```
> db.movies.find({franchise:"The Hobbit"}).pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f4"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f6"),
  "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

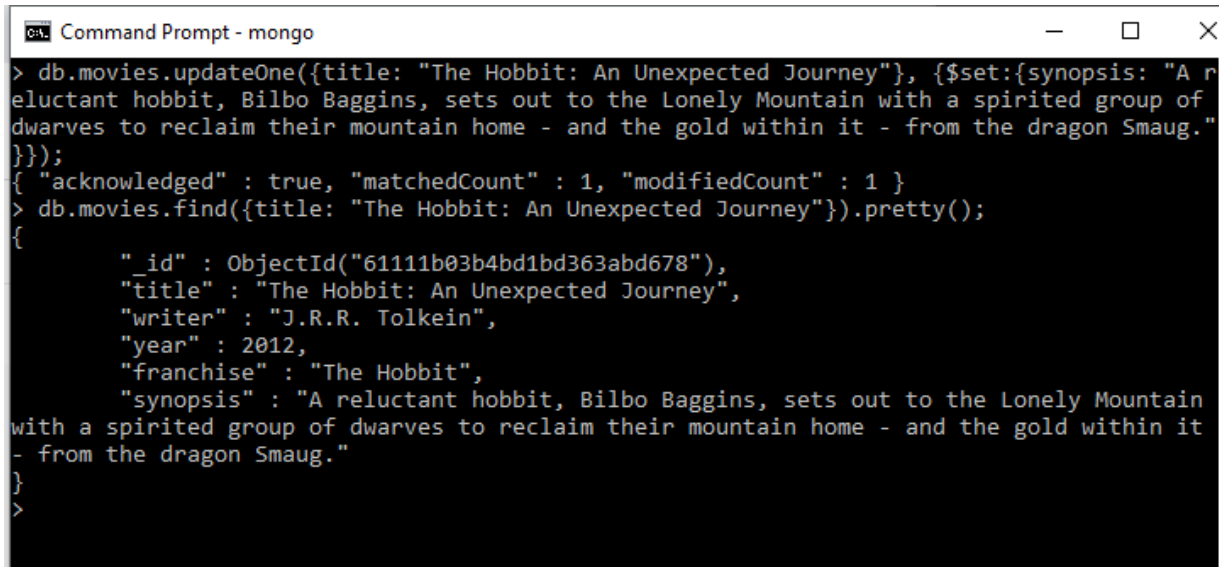
```
> db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]}).pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f1"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f2"),
  "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

```
Command Prompt - mongo
> db.movies.find({$or: [{year: {$lt: 2000}}, {year: {$gt: 2010}}]}).pretty()
{
  "_id" : ObjectId("6110eb8c54966d89050945f1"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f2"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f4"),
  "title" : "The Hobbit: An Unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "title" : "The Hobbit: The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6110eb8c54966d89050945f6"),
  "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."
}
```

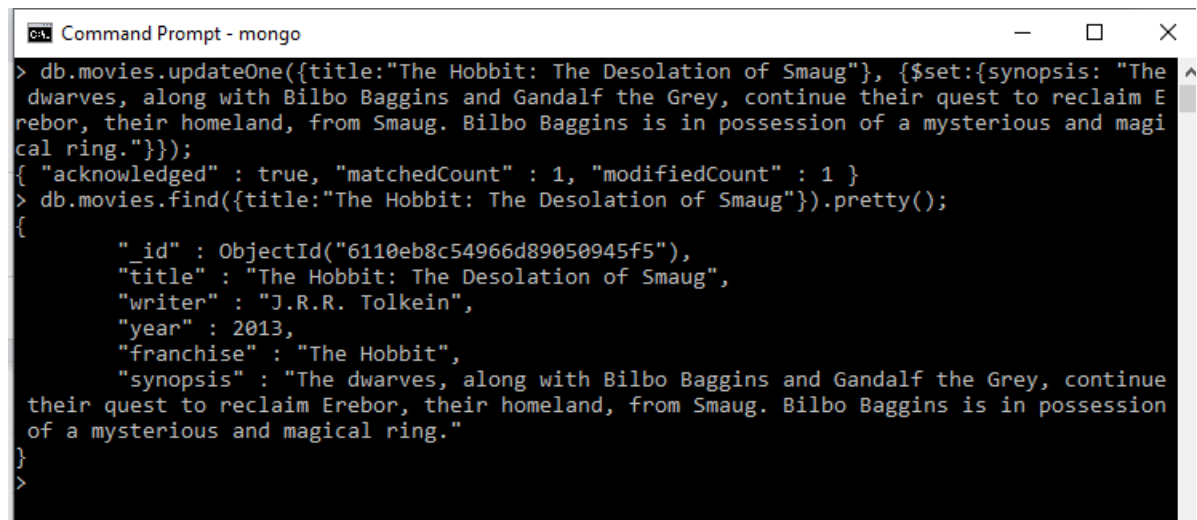
Update Documents

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."



```
Command Prompt - mongo
> db.movies.updateOne({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."}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.movies.find({title: "The Hobbit: An Unexpected Journey"}).pretty();
{
  "_id" : ObjectId("61111b03b4bd1bd363abd678"),
  "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."
}
```

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."



```
Command Prompt - mongo
> db.movies.updateOne({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."}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.movies.find({title:"The Hobbit: The Desolation of Smaug"}).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "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."
}
```

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

```
> db.movies.updateOne({title:"Pulp Fiction"}, {$set:{actor:"Samuel L. Jackson"}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.movies.find({title:"Pulp Fiction"}).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f2"),
  "title" : "Pulp Fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ],
  "actor" : "Samuel L. Jackson"
}
```

Text Search

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

```
@3:11/11:30
> db.movies.find({ synopsis: /Bilbo/g }).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "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("6110eb8c54966d89050945f6"),
  "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("61111b03b4bd1bd363abd678"),
  "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."
}
>
```

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

```
> db.movies.find({ synopsis: /Gandalf/g }).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "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."
}
```

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

```
> db.movies.find({ $and: [{ synopsis: /Bilbo/g }, { synopsis: { $not: /Gandalf/g } }] }).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f6"),
  "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 combatant
s and keep the Lonely Mountain from falling into the hands of a rising darkness."
}
{
  "_id" : ObjectId("61111b03b4bd1bd363abd678"),
  "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 spi
rited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Sm
aug."
}
```

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

```
> db.movies.find({ synopsis: /(dwarves|hobbit)/g }).pretty();
{
  "_id" : ObjectId("6110eb8c54966d89050945f5"),
  "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 qu
est to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious a
nd magical ring."
}
{
  "_id" : ObjectId("61111b03b4bd1bd363abd678"),
  "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 spi
rited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Sm
aug."
}
```


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

```
> db.movies.find({ synopsis: /(gold.*dragon|dragon.*gold)/g }).pretty();
{
  "_id" : ObjectId("61111b03b4bd1bd363abd678"),
  "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."
}
```

```
> db.movies.find({ $and: [{ synopsis: /gold/g }, { synopsis: /dragon/g }] }).pretty();
{
  "_id" : ObjectId("61111b03b4bd1bd363abd678"),
  "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."
}
```

Delete Documents

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

```
> db.movies.deleteMany({ title: "Pee Wee Herman's Big Adventure" });
{ "acknowledged" : true, "deletedCount" : 1 }
```

2. Delete the movie "Avatar"

```
> db.movies.deleteMany({ title: "Avatar" });
{ "acknowledged" : true, "deletedCount" : 1 }
> _
```

Relationships

Insert the following documents into a users collection

```
Command Prompt - mongo
> db.users.insertMany([
... {
...   username: "GoodGuyGreg",
...   first_name: "Good Guy",
...   last_name: "Greg"
... },
... {
...   username: "ScumbagSteve",
...   full_name: {
...     first: "Scumbag",
...     last: "Steve"
...   }
... }
... ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61113d75d527fae44826ad45"),
    ObjectId("61113d75d527fae44826ad46")
  ]
}
```

Insert the following documents into a posts collection

```
Command Prompt - mongo
> db.posts.insertMany([
... {
...   username: "GoodGuyGreg",
...   title: "Passes out at party",
...   body: "Wakes up early and cleans house"
... },
... {
...   username: "GoodGuyGreg",
...   title: "Steals your identity",
...   body: "Raises your credit score"
... },
... {
...   username: "GoodGuyGreg",
...   title: "Reports a bug in your code",
...   body: "Sends you a Pull Request"
... },
... {
...   username: "ScumbagSteve",
...   title: "Borrows something",
...   body: "Sells it"
... },
... {
...   username: "ScumbagSteve",
...   title: "Borrows everything",
...   body: "The end"
... },
... {
...   username: "ScumbagSteve",
...   title: "Forks your repo on github",
...   body: "Sets to private"
... }
... ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("611142c5d527fae44826ad47"),
    ObjectId("611142c5d527fae44826ad48"),
    ObjectId("611142c5d527fae44826ad49"),
    ObjectId("611142c5d527fae44826ad4a"),
    ObjectId("611142c5d527fae44826ad4b"),
    ObjectId("611142c5d527fae44826ad4c")
  ]
}
```

Insert the following documents into a comments collection

```
CA Command Prompt - mongo
> db.posts.find().pretty();
{
  "_id" : ObjectId("611142c5d527fae44826ad47"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad48"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad49"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
```

```
CA Command Prompt - mongo
> db.comments.insertMany([
... {
...   username: "GoodGuyGreg",
...   comment: "Hope you got a good deal!",
...   post: ObjectId("611142c5d527fae44826ad4a")
... },
... {
...   username: "GoodGuyGreg",
...   comment: "What's mine is yours!",
...   post: ObjectId("611142c5d527fae44826ad4b")
... },
... {
...   username: "GoodGuyGreg",
...   comment: "Don't violate the licensing agreement!",
...   post: ObjectId("611142c5d527fae44826ad4c")
... },
... {
...   username: "ScumbagSteve",
...   comment: "It still isn't clean",
...   post: ObjectId("611142c5d527fae44826ad47")
... },
... {
...   username: "ScumbagSteve",
...   comment: "Denied your PR cause I found a hack",
...   post: ObjectId("611142c5d527fae44826ad49")
... }
... ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("611147acd527fae44826ad4d"),
    ObjectId("611147acd527fae44826ad4e"),
    ObjectId("611147acd527fae44826ad4f"),
    ObjectId("611147acd527fae44826ad50"),
    ObjectId("611147acd527fae44826ad51")
  ]
}
```

Querying related collections

1. Find all users

```
Command Prompt - mongo
> db.users.find().pretty();
{
  "_id" : ObjectId("61113d75d527fae44826ad45"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"
}
{
  "_id" : ObjectId("61113d75d527fae44826ad46"),
  "username" : "ScumbagSteve",
  "full_name" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
```

2. Find all posts

```
Command Prompt - mongo
> db.posts.find().pretty();
{
  "_id" : ObjectId("611142c5d527fae44826ad47"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad48"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad49"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
>
```

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

```
Command Prompt - mongo
> db.posts.find({ username: 'GoodGuyGreg' }).pretty();
{
  "_id" : ObjectId("611142c5d527fae44826ad47"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad48"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad49"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request"
}
>
```

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

```
Command Prompt - mongo
> db.posts.find({ username: 'ScumbagSteve' }).pretty();
{
  "_id" : ObjectId("611142c5d527fae44826ad4a"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4b"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("611142c5d527fae44826ad4c"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
>
```

5. Find all comments

```
Command Prompt - mongo
> db.comments.find().pretty();
{
  "_id" : ObjectId("611147acd527fae44826ad4d"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("611142c5d527fae44826ad4a")
}
{
  "_id" : ObjectId("611147acd527fae44826ad4e"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("611142c5d527fae44826ad4b")
}
{
  "_id" : ObjectId("611147acd527fae44826ad4f"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : ObjectId("611142c5d527fae44826ad4c")
}
{
  "_id" : ObjectId("611147acd527fae44826ad50"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : ObjectId("611142c5d527fae44826ad47")
}
{
  "_id" : ObjectId("611147acd527fae44826ad51"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("611142c5d527fae44826ad49")
}
>
```

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

```
Command Prompt - mongo
> db.comments.find({ username: 'GoodGuyGreg' }).pretty();
{
  "_id" : ObjectId("611147acd527fae44826ad4d"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you got a good deal!",
  "post" : ObjectId("611142c5d527fae44826ad4a")
}
{
  "_id" : ObjectId("611147acd527fae44826ad4e"),
  "username" : "GoodGuyGreg",
  "comment" : "What's mine is yours!",
  "post" : ObjectId("611142c5d527fae44826ad4b")
}
{
  "_id" : ObjectId("611147acd527fae44826ad4f"),
  "username" : "GoodGuyGreg",
  "comment" : "Don't violate the licensing agreement!",
  "post" : ObjectId("611142c5d527fae44826ad4c")
}
>
```

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

```
CA: Command Prompt - mongo
> db.comments.find({ username: 'ScumbagSteve' }).pretty()
{
  "_id" : ObjectId("611147acd527fae44826ad50"),
  "username" : "ScumbagSteve",
  "comment" : "It still isn't clean",
  "post" : ObjectId("611142c5d527fae44826ad47")
}
{
  "_id" : ObjectId("611147acd527fae44826ad51"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("611142c5d527fae44826ad49")
}
>
```

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

```
CA: Select Command Prompt - mongo
> db.posts.aggregate([
... {
... $match: { title: 'Reports a bug in your code' }
... },
... {
... $lookup: {
... from: 'comments',
... localField: '_id',
... foreignField: 'post',
... as: 'comments'
... }
... }
... ]).pretty();
{
  "_id" : ObjectId("611142c5d527fae44826ad49"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sends you a Pull Request",
  "comments" : [
    {
      "_id" : ObjectId("611147acd527fae44826ad51"),
      "username" : "ScumbagSteve",
      "comment" : "Denied your PR cause I found a hack",
      "post" : ObjectId("611142c5d527fae44826ad49")
    }
  ]
}
>

> db.comments.find({ post:ObjectId("611142c5d527fae44826ad49")}).pretty();
{
  "_id" : ObjectId("611147acd527fae44826ad51"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("611142c5d527fae44826ad49")
}
>
```

