

## Assignment 1

### Find Document

1. get all documents

```
--> db.movies.find().pretty()
```

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

```
-->db.movies.find( {writer: "Quentine Tarantino"} )
```

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

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

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

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

5. get all movies released in the 90s

```
-->db.movies.find({year: {$gte: 1990, $lte: 1999 } } )
```

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

```
-->db.movies.find( { $or: [{ year: {$lte: 2000}}, {year: {$gte: 2010}}] })
```

### Update

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"

```
-->
```

```
db.movies.update({title: "The Hobbit And 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. Smaug. Bilbo Baggins is in mysterious and magical ring" session of a

-->

```
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. Smaug. Bilbo Baggins is in mysterious and magical ring"}})
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

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

-->

```
db.movies.update( {title: "pulp fiction"}, {$push: {actors: ""Samuel L Jackson""}})
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

## Text search

```
db.movies.createIndex({synopsis: "text"})
```

```
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

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

--> db.movies.find({\$text: {\$search: "Bilbo"}})

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

-->db.movies.find({\$text: {\$search: "Gandal"}})

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

--> db.movies.find({\$text: {\$search: "Bilbo -Gandalf"}})

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

-->db.movies.find({\$text: {\$search: "dwarves hobbit"}})

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

-->db.movies.find({\$text: {\$search: "gold dragon"}})

## Delete Documents

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

-->db.movies.remove({title: "Pee Vee Hirman's Big Adventure"})

2 delete the movie "Avatar"

-->db.movies.remove({title: "Avatar"})

## Relationship:

1. find all users

```
--> db.users.find({}).pretty()
```

2. find all posts

```
--> db.posts.find({}).pretty()
```

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

```
--> db.posts.find({username: "GoodGuyGreg"}).pretty()
```

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

```
-->db.posts.find({username: "ScumbagSteve"}).pretty()
```

5. find all comments

```
--> db.comments.find({}).pretty()
```

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

```
--> db.comments.find({username: "GoodGuyGreg"}).pretty()
```

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

```
--> db.comments.find({username: "ScumbagSteve"}).pretty()
```

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

```
db.comments.find({post: db.posts.findOne({title: "Reports a bug in your  
code"})._id})
```