

## Query / Find Documents

---

1. get all documents

```
db.getCollection('movies').find({})
```

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

```
db.movies.find({writer : 'Quentin 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: { $lt: 2000 } }, { year: { $gt: 2010 } } ] } );
```

## 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."

```
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."}
    });
```

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."

```
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." }
    });
```

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

```
db.movies.update(
    {title: 'Pulp Fiction'},
    {
        $push: {actors: 'Samuel L. Jackson'}
    });
```

## Text Search

---

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

```
db.movies.find({ synopsis: { $in: [ /.*Gandalf./ ] } });
```

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

```
db.movies.find({ synopsis: { $in: [ /.*Bilbo./ ] } });
```

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

```
db.movies.createIndex( { synopsis: "text" } );  
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: "dragon" }, $search : "gold" } });
```

## Delete Documents

---

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

```
db.movies.deleteOne({title:"Pee Wee Herman's Big Adventure"});
```

2. delete the movie "Avatar"

```
db.movies.deleteOne({title:"Avatar"});
```

## Querying related collections

---

1. find all users

```
db.getCollection('users').find({})
```

2. find all posts

```
db.getCollection('posts').find({})
```

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

```
db.posts.find({username: "GoodGuyGreg"});
```

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

```
db.posts.find({username: "ScumbagSteve"});
```

5. find all comments

```
db.comments.find({});
```

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

```
db.comments.find({username : "GoodGuyGreg"});
```

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

```
db.comments.find({username : "ScumbagSteve"});
```

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

```
var post_id = db.posts.findOne({"title":"Forks your repo on  
github"},{"_id":1});
```

```
db.comments.find({post: post_id["_id"]});
```