

MONGODB ASSIGNMENT 1

QUERY/FIND DOCCUMENTS

1. get all documents

sol: `db.movies.find()`

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

sol: `db.movies.find({writer: "Quentin Tarantino"})`

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

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

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

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

5. get all movies released in the 90s

sol: `db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]})`

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

Sol: `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."

Sol: `db.movies.update({title: "The Hobbit: An Unexpected Journey"}, {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."

Sol: `db.movies.update({title: "The Hobbit: The Desolation of Smaug"}, {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"

sol: `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 "Bilbo"

```
sol: db.movies.find({'$text': {'$search': "Bilbo"}})
```

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

```
sol: db.movies.find({'$text': {'$search': "Gandalf"}})
```

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

```
sol: db.movies.find({'$and': [{synopsis: /Bilbo/}, {synopsis: {'$not': /Gandalf/}}]}).pretty()
```

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

```
sol: db.movies.find({'$or': [{synopsis: /dwarves/}, {synopsis: /hobbit/}]}).pretty()
```

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

```
sol: db.movies.find({'$and': [{synopsis: /.gold./}, {synopsis: /\.^Dragon./}]}).pretty()
```

DELETE DOCCUMENTS

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

sol: db.movies.remove({title: "Pee Wee Herman's Big Adventure"})

2. delete the movie "Avatar"

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

QUERYING RELATED COLLECTIONS (RELATIONSHIPS)

1. find all users

sol: `db.users.find()`

2. find all posts

Sol: `db.posts.find()`

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

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

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

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

5. find all comments

Sol: `db.comments.find()`

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

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

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

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

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

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

