

Pill MongoDB Solutions

1. Introduction

To solve this exercise, the MongoDB 'mongo shell' will be used.

You must connect to an active instance of mongodb, you can use a database called `mongo_exercise`.

Document all your queries in a javascript file to use as a reference.

2. Documents to Insert

Insert the following documents into a collection called `movies`.

```
...
```

```
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. Tolkien

year : 2012

franchise : The Hobbit

~ ~ ~

~ ~ ~

title : The Hobbit: The Desolation of Smaug

writer : J.R.R. Tolkien

year : 2013

franchise : The Hobbit

~ ~ ~

~ ~ ~

title : The Hobbit: The Battle of the Five Armies

writer : J.R.R. Tolkien

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.

~ ~ ~

```

title : Pee Wee Herman's Big Adventure

```

```

title : Avatar

```

3. Queries / Search documents

Perform the following queries on the `movies` collection:

1. Get all the documents
2. Get documents with `writer` equal to "Quentin Tarantino"
3. Get documents with `actors` that include "Brad Pitt"
4. Get documents with `franchise` equal to "The Hobbit"
5. Get all the 90s movies
6. Obtain the films released between the year 2000 and 2010

Solutions:

1. `db.movies.find()`
2. `db.movies.find({"writer":"Quentin Tarantino"})`
3. `db.movies.find({actors:"Brad Pitt"})`
4. `db.movies.find({franchise:"The Hobbit"})`
5. `db.movies.find({ $and: [{year: { $gt:1990} } , {year:{ $lt:2000} }] })`
6. `db.movies.find({$and: [{year: { $gt: 2000}}, {year: { $lt: 2010}}]})`

4. Update Documents

1. Add 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 ."

2. Add 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."

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

Solutions:

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

5. Search by Text / Text Search

1. Find the movies that contain the word "Bilbo" in the synopsis

2. Find the movies that contain the word "Gandalf" in the synopsis

3. Find the movies that in the synopsis contain the word "Bilbo" and not the word "Gandalf"

4. Find the movies that contain the word "dwarves" or "hobbit" in the synopsis

5. Find the movies that contain the words "gold" and "dragon" in the synopsis

Solutions:

1. `db.movies.find({synopsis: /Bilbo/})`

2. `db.movies.find({synopsis:/Gandalf/})`

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

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

5. `db.movies.find({$and:[{synopsis:/gold/},{synopsis:/dragon/}]})`

6. Delete Documents

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

Solutions

- 1.. `db.movies.deleteOne({title:"Pee Wee Herman's Big Adventure"})`
2. `db.movies.deleteOne({"title": "Avatar"})`