MongoDB Lab Assignments -Day 1

MongoDB Exercise in mongo shell

Connect to a running mongo instance, use a database named **mongo_practice**. Document all your queries in a javascript file to use as a reference.

```
Atlas atlas-ca07dn-shard-0 [primary] test> use mongo_practice

switched to db mongo_practice

Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
db.createCollection('movies');

{ ok: 1 }

Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> show dbs

mongo_practice  8.19 kB

admin           336 kB
```

Insert Documents

Insert the following documents into a **movies** collection.

```
title : Fight Club
writer : Chuck Palahniuko
year : 1999
actors : [
 Brad Pitt
 Edward Norton
]
```

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.insert({"title":"fight
club","writer":"chuck palahniuko","year":"1999","actors":[ "Brad pitt","Edward
Norton"]});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("61d94e541aed68f981724c4d") }
}
```

```
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
```

darkness.

title : Pee Wee Herman's Big Adventure

title : Avatar

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.insert({"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","title":"Pee wee herman's big adventure","title":"avatar"});
```

Reference
https://www.tutorialspoint.com/mongodb/mongodb_insert_document.htm

Query / Find Documents

query the **movies** collection to

1. get all documents

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.find()

[

  {

    _id: ObjectId("61d910051aed68f981724c4c"),

    title: 'fight club',

    writer: 'chuck palahniuko',

    year: '1999'

  },

  {

    _id: ObjectId("61d94e541aed68f981724c4d"),
    title: 'fight club',
    writer: 'chuck palahniuko',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("61d950dc1aed68f981724c4e"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: [ 'John Travolta', 'Uma Thruman' ]
  },
  {
    _id: ObjectId("61d99636e0f4b650f37cd0ab"),
    title: { 'the hobbit': 'An unexpected journey' },
    writer: 'J.R.R.Tolkein',
    year: '2012',
    franchise: 'the hobbit'
  },
  {
    _id: ObjectId("61d998d9e0f4b650f37cd0ac"),
    title: 'avatar',
    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'

  }

]
```

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

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
db.movies.findOne({"writer":"Quentin Tarantino"});
{
  _id: ObjectId("61d950dc1aed68f981724c4e"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '1994',
  actors: [ 'John Travolta', 'Uma Thruman' ]
}
```

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

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
db.movies.findOne({"actors":"Brad pitt"});
{
  _id: ObjectId("61d94e541aed68f981724c4d"),
  title: 'fight club',
  writer: 'chuck palahniuko',
  year: '1999',
  actors: [ 'Brad pitt', 'Edward Norton' ]
}
```

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

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
 db.movies.findone{("franchise":"the hobbit"});
```

```
_id: ObjectId("61d99636e0f4b650f37cd0ab"),
  title: { 'the hobbit': 'An unexpected journey' },
  writer: 'J.R.R.Tolkein',
  year: '2012',uck palahniuko',
  franchise: 'the hobbit'
}   actors: [ 'Brad pitt', 'Edward Norton' ]
  }
```

5. get all movies released in the 90s

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.find({"year":{
$not:{ $gt:"2000"}}})
[
  {
    _id: ObjectId("61d910051aed68f981724c4c"),
    title: 'fight club',
    writer: 'chuck palahniuko',
    year: '1999'
  },
  {
    _id: ObjectId("61d94e541aed68f981724c4d"),
    title: 'fight club',
    writer: 'chuck palahniuko',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("61d950dc1aed68f981724c4e"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: [ 'John Travolta', 'Uma Thruman' ]
  }
]
```

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

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.find({$and:
```

```
[{"year": {$gt:"1900"}},{"year": {$lt:"2000"}}]})
[
  {
    _id: ObjectId("61d910051aed68f981724c4c"),
    title: 'fight club',
    writer: 'chuck palahniuko',
    year: '1999'
  },
  {
    _id: ObjectId("61d94e541aed68f981724c4d"),
    title: 'fight club',
    writer: 'chuck palahniuko',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("61d950dc1aed68f981724c4e"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: [ 'John Travolta', 'Uma Thruman' ]
  }
]
```

Reference:
https://www.tutorialspoint.com/mongodb/mongodb_query_document.htm

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."

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
    db.movies.update({"title":{"the hobbit":"An unexpected journey"}},{
    $push:{"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 swaug"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne,
    updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
    db.movies.find({"title":{"the hobbit":"An unexpected journey"}}).pretty()
[
  {
    _id: ObjectId("61d99636e0f4b650f37cd0ab"),
    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 swaug'
    ]
  }
]
```

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."

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
```

```
    db.movies.update({"title":{"the hobbit":"The desolation of
    smaug"}},{$push:{"synopsis":"The dwares,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,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice> db.movies.find({"title":{"the
    hobbit":"The desolation of smaug"}}).pretty()
[
  {
    _id: ObjectId("61d9c655e0f4b650f37cd0ad"),
    title: { 'the hobbit': 'The desolation of smaug' },
    writer: 'J.R.R.Tolkein',
    year: '2012',
    franchise: 'The hobbit',
    synopsis: [
      'The dwares,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"

Reference:
https://www.tutorialspoint.com/mongodb/mongodb_update_document.htm

Text Search

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

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

db.movies.find({$text:{$search:”Bilbi”}}).pretty()
```

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

```
db.movies.find({$text:{$search:”Gandalf”}}).pretty()
```

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()
```

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

```
db.movies.find({synopsis:/(dwarvws/hobbit)/g}.pretty()
```

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

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

Reference: https://www.tutorialspoint.com/mongodb/mongodb_text_search.htm

Delete Documents

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

```
Atlas atlas-ca07dn-shard-0 [primary] mongo_practice>
db.movies.remove({"title":"Pee wee herman's big adventure"})
{ acknowledged: true, deletedCount: 1 }
```

2. delete the movie "Avatar"

Reference:
https://www.tutorialspoint.com/mongodb/mongodb_delete_document.htm

Relationships

Insert the following documents into a **users** collection

username : GoodGuyGreg
first_name : "Good Guy"
last_name : "Greg"

username : ScumbagSteve
full_name :
 first : "Scumbag"
 last : "Steve"

Insert the following documents into a **posts** collection

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

Insert the following documents into a **comments** collection

username : GoodGuyGreg
comment : Hope you got a good deal!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Borrows something"

username : GoodGuyGreg
comment : What's mine is yours!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Borrows everything"

username : GoodGuyGreg
comment : Don't violate the licensing agreement!
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Forks your repo on github

username : ScumbagSteve
comment : It still isn't clean
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Passes out at party"
username : ScumbagSteve
comment : Denied your PR cause I found a hack
post : [post_obj_id]
where [post_obj_id] is the ObjectId of the posts document: "Reports a bug in your code"

Querying related collections

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”}).
```

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"

```
db.posts.aggregate([
{
$match:{title:”Reports a bug in your code”}
},{
$lookup:{
from:”comments”,localfield:”_id”,foreignField:”post”,as:”comments”}}]).pretty();
```

References:

https://docs.mongodb.com/manual/reference/method/db.collection.find/

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@@@