5 Basic Operations
5.1 read queries

Question 1

How many users are there in the database?
Ref: http://docs.mongodb.org/manual/reference/command/count/.

```
 db.runCommand({count: "users"})
{ "n" : 6040, "ok" : 1 }

db.users.count()
6040
```

Question 2

How many movies are there in the database?

```
db.runCommand({count: "movies"})
{ "n" : 3883, "ok" : 1 }

db.movies.count()
3883
```

Question 3

Which is the occupation of Clifford Johnathan? In the answer show only its name and occupation.
Ref: http://docs.mongodb.org/manual/reference/method/db.collection.find/.

```
db.users.find({name:"Clifford Johnathan"},{occupation:1,name:1,_id:0});
{ "name" : "Clifford Johnathan", "occupation" : "technician/engineer" }
```

Question 4

How many users are there between 18 and 30 years old (both included)?

```
db.users.find( {age: { $gt:18, $lt:30}});
<!> Attention dangerous command <!>

db.users.count({age:{$gt:18,$lt:30}});
2004
```

Question 5

How many users are artist or scientist?

```
db.users.count({occupation:"artist"},{occupation:"scientist"});
267
db.users.count({occupation:"scientist"},{occupation:"artist"});
144
 db.users.count({occupation:"scientist"});
```

144
db.users.count({occupation:"artist"});
267
db.users.count({occupation:"scientist"}) + db.users.count({occupation:"artist"});
411
> db.users.count({occupation:"scientist" && "artist"});
267
> db.users.count({occupation:"scientist" || "artist"});
144

> db.users.findOne({occupation:"artist"},{occupation:"scientist"});
{ "_id" : 5945, "occupation" : "artist" }
> db.users.findOne({occupation:"scientist"},{occupation:"artist"});
{ "_id" : 6012, "occupation" : "scientist" }

Question6

Who are the 10 oldest woman writers?
db.users.find({"gender" : "F"},{name:1,age:1,_id:0}).sort({age:-1}).limit(10)
{ "name" : "Jestine Booker", "age" : 99 }
{ "name" : "Babara Elden", "age" : 98 }
{ "name" : "Susanna Shaun", "age" : 96 }
{ "name" : "Yaeko Hassan", "age" : 95 }
{ "name" : "Linh Tyrell", "age" : 95 }
{ "name" : "Ka Joe", "age" : 94 }
{ "name" : "Lashandra Sal", "age" : 94 }
{ "name" : "Starla Desmond", "age" : 94 }
{ "name" : "Lakeisha Wilbur", "age" : 94 }
{ "name" : "Aleshia Carol", "age" : 94 }

Question7

Show all the occupations in the database.

```
 db.users.distinct("occupation")
[
        "academic/educator",
        "sales/marketing",
        "other",
        "doctor/health care",
        "retired",
        "executive/managerial",
        "college/grad student",
        "technician/engineer",
        "scientist",
        "programmer",
        "self-employed",
        "homemaker",
        "writer",
        "customer service",
        "K-12 student",
        "clerical/admi",
        "lawyer",
        "artist",
        "unemployed",
        "tradesman/craftsman",
        "farmer"
]
```

Question 8

Insert yourself in the database but do not include the field movies. Do not worry about privacy, you are not obliged to use real information.

```
        "_id" : 6038,
        "name" : "Yaeko Hassan",
        "gender" : "F",
```

```
        "age" : 95,
        "occupation" : "academic/educator",
_

> db.users.insert({"_id" : 9999 , "name" : "ORTOLE Loic", "gender" : "M", "age" : 22, "occupation" :
"K-12 student" })
WriteResult({ "nInserted" : 1 })
> db.users.findOne({_id:9999})
{
        "_id" : 9999,
        "name" : "ORTOLE Loic",
        "gender" : "M",
        "age" : 22,
        "occupation" : "K-12 student"
}
```

Question 9

Select one movie from the collection movies and update your entry by adding the field movies, which is an array, including a review of the film.
Hint: You can use the current timestamp : Math.round(new Date().getTime() / 1000).

```
db.movies.update(
        {"_id":1,"title":"Toy Story (1995)","genres":"Animation|Children's|Comedy"},
        {"_id":1,"title":"Toy Story (1995)","genres":"Animation|Children's|Comedy",
        "rating":3,"timestamp":Math.round(new Date().getTime()/1000)},
        { upsert: true })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

 db.movies.findOne({_id :1})
{
        "_id" : 1,
        "title" : "Toy Story (1995)",
        "genres" : "Animation|Children's|Comedy",
        "rating" : 3,
        "timestamp" : 1427116013
}
```

```
db.users.update({"_id" : 9999},{$set:{"movieid":2700,"rating":1, "timestamp":(Math.round(new Date().getTime()/1000))}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

db.users.findOne({_id:9999})
{
        "_id" : 9999,
        "name" : "ORTOLE Loic",
        "gender" : "M",
        "age" : 22,
        "occupation" : "K-12 student",
        "movieid" : 2700,
        "rating" : 1,
        "timestamp" : 1427114679
}

db.users.update({"_id" : 9999},{$set: {movies:[{"movieid":2700,"rating":1, "timestamp":(Math.round(new Date().getTime()/1000))}]}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Question 10

Remove your entry from the collection.

Do not worry if you accidentally removed other entries from the collection users. You can always recreate the original one by removing all its documents and reimporting them with mongoimport.

```
> db.users.remove({_id:9999})
WriteResult({ "nRemoved" : 1 })
> db.users.findOne({_id:9999})
null
```

Question 11

Change in all users the occupation programmer by developer.

```
db.users.findOne({occupation:"programmer"})
{
        "_id" : 6024,
        "name" : "Shawn Saul",
        "gender" : "M",
        "age" : 31,
        "occupation" : "programmer",
        "movies" : [
```

```
                {
                        "movieid" : 2058,
                        "rating" : 4,
                        "timestamp" : 956749684
                },
                ...,
                {
                        "timestamp" : 956749155,
                        "movieid" : 1097,
                        "rating" : 4
                }
        ]
}

db.users.update({occupation:"programmer"},{occupation:"developer"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

db.users.findOne({"_id" : 6024})
{ "_id" : 6024, "occupation" : "developer" }

//oulala mais ou sont les movies ??

"_id" : 6009,
```

//Pour modifier toutes les lignes

```
db.users.update({"occupation":"programmer"},{$set:{"occupation":"developer"}},{upsert:false,multi:true});
WriteResult({ "nMatched" : 386, "nUpserted" : 0, "nModified" : 386 })
```

Question 12

How many movies have been released in the 80s?

```
db.movies.count({title:{$regex:/198/}});
598
```

Question 13

How many movies have been released between 1984 and 1992?

```
db.movies.count({"title":{$in: [/1984/,/1985/,/1986/,/1987/,/1988/,/1989/,/1990/,/1991/,/1992/]}})
668
```

db.movies.count({"title":{$in: [/198[456789]/,/199[012]/]}})
668

db.movies.count({"title":{$in: [/198[4-9]/,/199[0-2]/]}})
668

Question 14

How many horror movies are there?
db.movies.count({"genres":"Horror"});
178
db.movies.count({"genres":{$regex: /Horror/}})
343

Question 15

How many movies are both Musical and Romance?
db.movies.count({"genres":"Musical|Romance"});
5
db.movies.find({"genres":"Musical|Romance"})
{ "_id" : 899, "title" : "Singin' in the Rain (1952)", "genres" : "Musical|Romance" }
{ "_id" : 900, "title" : "American in Paris, An (1951)", "genres" : "Musical|Romance" }
{ "_id" : 914, "title" : "My Fair Lady (1964)", "genres" : "Musical|Romance" }
{ "_id" : 1088, "title" : "Dirty Dancing (1987)", "genres" : "Musical|Romance" }
{ "_id" : 1947, "title" : "West Side Story (1961)", "genres" : "Musical|Romance" }

Question 16

As you have just seen, having the year integrated in the title name is not very practical.

Modify the collection movies by creating a new field called year and removing it from the title.
Ref: http://docs.mongodb.org/manual/reference/method/cursor.forEach/.
Hint: You may use the command replace to select the parts of the title that you need. Remark: MongoDB stores documents of a collection as list, one after the other. Between documents a padding is included so that changes in the size of a specific document can be done in place. However, if the document size increases more than the assigned padding, it it is relocated at the end of the list. When a normal find() is executed in MongoDB, documents are not retrieved all at once but in batches. As a consequence, if when accessing a collection, we modify a document and it has to be relocated at the end, it may be returned twice. In order to avoid this situation, MongoDB provides the command snapshot() which traverses the id index. However it makes queries slower, so it should only be used when needed.

Ref: http://docs.mongodb.org/manual/reference/method/cursor.snapshot/

```
db.movies.find().snapshot().forEach(function(m) {
var title = m.title;
m.year = parseInt(title.replace(/.*\(/,"").replace(/\).*/,""));
m.title = title.replace( / *\([0-9]*\) */,"");
db.movies.save(m);
});

db.movies.find({"genres":"Musical|Romance"})
{ "_id" : 899, "title" : "Singin' in the Rain", "genres" : "Musical|Romance", "year" : 1952 }
{ "_id" : 900, "title" : "American in Paris, An", "genres" : "Musical|Romance", "year" : 1951 }
{ "_id" : 914, "title" : "My Fair Lady", "genres" : "Musical|Romance", "year" : 1964 }
{ "_id" : 1088, "title" : "Dirty Dancing", "genres" : "Musical|Romance", "year" : 1987 }
{ "_id" : 1947, "title" : "West Side Story", "genres" : "Musical|Romance", "year" : 1961 }
```

Question 17

Modify the collection movies by replacing the string field genres with an array that contains all the genres.

```
db.movies.find().snapshot().forEach(function(m) {
var genres = m.genres.split("|");
m.genres = genres;
db.movies.save(m);
});

db.movies.find({"year":1952})
{ "_id" : 2848, "title" : "Othello", "genres" : [ "Drama" ], "year" : 1952 }
{ "_id" : 3207, "title" : "Snows of Kilimanjaro, The", "genres" : [ "Adventure" ], "year" : 1952 }
{ "_id" : 3229, "title" : "Another Man's Poison", "genres" : [ "Crime", "Drama" ], "year" : 1952 }
{ "_id" : 3314, "title" : "Big Trees, The", "genres" : [ "Action", "Drama" ], "year" : 1952 }
{ "_id" : 3559, "title" : "Limelight", "genres" : [ "Drama" ], "year" : 1952 }
{ "_id" : 899, "title" : "Singin' in the Rain", "genres" : [ "Musical", "Romance" ], "year" : 1952 }
{ "_id" : 1070, "title" : "Macao", "genres" : [ "Adventure" ], "year" : 1952 }
{ "_id" : 1226, "title" : "Quiet Man, The", "genres" : [ "Comedy", "Romance" ], "year" : 1952 }
{ "_id" : 1283, "title" : "High Noon", "genres" : [ "Western" ], "year" : 1952 }
{ "_id" : 1943, "title" : "Greatest Show on Earth, The", "genres" : [ "Drama" ], "year" : 1952 }
{ "_id" : 3417, "title" : "Crimson Pirate, The", "genres" : [ "Adventure", "Comedy", "Sci-Fi" ],
"year" : 1952 }
```

Question 18

Modify the collection user by replacing the field timestamp with a new field called date of type Date.
Hint: Field timestamp in the original dataset is in seconds since the Unix epoch, but dates in Javascript can be created by using the number of milliseconds since the Unix epoch. Hint: You can use delete Object.attribute to delete an attribute from an object. An- other possibility is using the option $unset from the update command.

```
db.users.find().snapshot().forEach(function(u) {
for (i = 0; i < u.movies.length; i++) {
date = new Date(u.movies[i].timestamp * 1000);
u.movies[i].date = date;
delete u.movies[i].timestamp;
}
```

Question 19

How many users have seen the movie with id 1196 (Star Wars: Episode V - The Empire Strikes Back (1980))?

```
db.users.count({"movies":{$all:[{"$elemMatch":{"movieid" :1196}}]}})
2988
db.users.count({'movies.movieid':1196});
2989
```

Question 20

How many users have seen all the movies from the old Star Wars trilogy (IV,V,VI) with ids: 260, 1196, 1210?

```
db.users.count({$and :
[{"movies.movieid":260},{"movies.movieid":1196},{"movies.movieid":1210}]})
1924
```

Question 21

How many users have rated exactly 48 movies?
Ref: http://docs.mongodb.org/manual/reference/operator/query/size/.

However, $size can only match exact number. Selecting users that have rated more than a specified number of movies has to be done in two steps which will be the subject of the following questions.

```
db.users.count({movies:{$size:48}})
51
```

Question 22

For each user create a field num ratings that display the number of ratings he has given.

```
db.users.find().snapshot().forEach(function(u){u.numRatings = u.movies.length;
db.users.save(u);});
> db.users.findOne()
{
        "_id" : 6038,
        "name" : "Yaeko Hassan",
        "gender" : "F",
        "age" : 95,
```

```
"occupation" : "academic/educator",
"movies" : [
        {
                "movieid" : 1419,
                "rating" : 4,
                "timestamp" : 956714815
        },
        {
                "timestamp" : 956706827,
                "movieid" : 920,
                "rating" : 3
        },
        {
                "movieid" : 3088,
                "rating" : 5,
                "timestamp" : 956707640
        },
        {
                "movieid" : 232,
                "rating" : 4,
                "timestamp" : 956707640
        },
        {
                "rating" : 4,
                "timestamp" : 956707708,
                "movieid" : 1136
        },
        {
                "timestamp" : 956707604,
                "movieid" : 1148,
                "rating" : 5
        },
        {
                "movieid" : 1183,
                "rating" : 5,
                "timestamp" : 956717204
        },
        {
                "movieid" : 2146,
                "rating" : 4,
                "timestamp" : 956706909
        },
        {
                "movieid" : 3548,
                "rating" : 4,
                "timestamp" : 956707604
        },
        {
                "movieid" : 356,
                "rating" : 4,
                "timestamp" : 956707005
        },
        {
                "movieid" : 1210,
                "rating" : 4,
                "timestamp" : 956706876
        },
```

```
                {
                        "rating" : 5,
                        "timestamp" : 956707734,
                        "movieid" : 1223
                },
                {
                        "movieid" : 1276,
                        "rating" : 3,
                        "timestamp" : 956707604
                },
                {
                        "movieid" : 1296,
                        "rating" : 5,
                        "timestamp" : 956714684
                },
                {
                        "movieid" : 1354,
                        "rating" : 3,
                        "timestamp" : 956714725
                },
                {
                        "movieid" : 1387,
                        "rating" : 2,
                        "timestamp" : 956707005
                },
                {
                        "movieid" : 2700,
                        "rating" : 1,
                        "timestamp" : 956715051
                },
                {
                        "timestamp" : 956707604,
                        "movieid" : 2716,
                        "rating" : 3
                },
                {
                        "movieid" : 3396,
                        "rating" : 3,
                        "timestamp" : 956706827
                },
                {
                        "movieid" : 1079,
                        "rating" : 5,
                        "timestamp" : 956707547
                }
        ],
        "numRatings" : 20
}
```

Question 23

How many users have rated more than 90 movies?

```
db.users.find({"numRatings": { $gt:90}}).count()
3114

db.users.count({"numRatings":{$in: [/9[0-9]/,/[1-9][1-9][1-9]/]}})
0
```

Question 24

How many ratings have been submitted after 1st January 2001?

```
> new Date()
ISODate("2015-03-26T10:45:08.263Z")

> Math.round(ISODate("2001-01-01T00:00:01.263Z").getTime()/1000)
978307201

>db.users.find({"movies.timestamp": { $gt:
Math.round(ISODate("2001-01-01T00:00:01.263Z").getTime()/1000)}}).count()
1177
```

Question 25

Which are the last 5 films rated by Jayson Brad. Do not use the date but the order in which they have been added to the array movies (assume that ratings are added to the end).

```
db.users.find({"name" :"Jayson Brad"},{"movies":{$slice:-5}})
{ "_id" : 6016, "name" : "Jayson Brad", "gender" : "M", "age" : 47, "occupation" :
"academic/educator", "movies" : [ { "rating" : 4, "timestamp" : 956780537, "movieid" : 562 }, {
"timestamp" : 956957974, "movieid" : 1096, "rating" : 4 }, { "movieid" : 1097, "rating" : 3,
"timestamp" : 956776340 }, { "movieid" : 2043, "rating" : 2, "timestamp" : 956778658 }, {
"movieid" : 3783, "rating" : 4, "timestamp" : 963610480 } ], "numRatings" : 909 }

db.users.find({"name" :"Jayson Brad"},{"movies.movieid":1,"movies":{$slice:-5}})
```

{ "_id" : 6016, "movies" : [{ "movieid" : 562 }, { "movieid" : 1096 }, { "movieid" : 1097 }, { "movieid" : 2043 }, { "movieid" : 3783 }] }

Question 26

Return only the information about Tracy Edward's information about the rating of the film Star Wars: Episode VI - Return of the Jedi (whose id is 1210).

db.users.find({"name" :"Tracy Edward","movies.movieid":1210},{"movies.rating":1})

db.users.find({"name" :"Tracy Edward")

{ "_id" : 5951, "name" : "Tracy Edward", "gender" : "M", "age" : 24, "occupation" : "college/grad student", "movies" : [{ "movieid" : 593, "rating" : 4, "timestamp" : 957159935 }, { "movieid" : 595, "rating" : 4, "timestamp" : 957160283 }, { "movieid" : 912, "rating" : 3, "timestamp" : 957160264 }, { "movieid" : 924, "rating" : 2, "timestamp" : 957160264 }, { "movieid" : 953, "rating" : 4, "timestamp" : 957160283 }, { "movieid" : 3421, "rating" : 4, "timestamp" : 957160240 }, { "movieid" : 2490, "rating" : 1, "timestamp" : 957160366 }, { "movieid" : 223, "rating" : 5, "timestamp" : 957160463 }, { "movieid" : 260, "rating" : 5, "timestamp" : 957160127 }, { "movieid" : 293, "rating" : 5, "timestamp" : 957160405 }, { "movieid" : 1136, "rating" : 5, "timestamp" : 957160184 }, { "rating" : 4, "timestamp" : 957160202, "movieid" : 608 }, { "timestamp" : 957160167, "movieid" : 1188, "rating" : 4 }, { "movieid" : 1196, "rating" : 5, "timestamp" : 957159915 }, { "rating" : 5, "timestamp" : 957159935, "movieid" : 1197 }, { "timestamp" : 957160418, "movieid" : 1500, "rating" : 4 }, { "movieid" : 356, "rating" : 4, "timestamp" : 957160348 }, { "movieid" : 2918, "rating" : 5, "timestamp" : 957160315 }, { "movieid" : 364, "rating" : 3, "timestamp" : 957159914 }, { "movieid" : 50, "rating" : 5, "timestamp" : 957160150 }, { "movieid" : 1200, "rating" : 5, "timestamp" : 957160382 }, { "rating" : 5, "timestamp" : 957160476, "movieid" : 1210 }, { "movieid" : 1221, "rating" : 2, "timestamp" : 957160092 }, { "timestamp" : 957160202, "movieid" : 1223, "rating" : 4 }, { "movieid" : 1242, "rating" : 5, "timestamp" : 957160405 }, { "rating" : 4, "timestamp" : 957160526, "movieid" : 1246 }, { "movieid" : 2987, "rating" : 4, "timestamp" : 957160488 }, { "movieid" : 720, "rating" : 5, "timestamp" : 957159935 }, { "timestamp" : 957160510, "movieid" : 1258, "rating" : 4 }, { "movieid" : 1259, "rating" : 5, "timestamp" : 957160451 }, { "movieid" : 1270, "rating" : 5, "timestamp" : 957160167 }, { "movieid" : 1278, "rating" : 4, "timestamp" : 957160283 }, { "movieid" : 750, "rating" : 2, "timestamp" : 957160079 }, { "movieid" : 1291, "rating" : 4, "timestamp" : 957160264 }, { "movieid" : 1606, "rating" : 3, "timestamp" : 957159915 }, { "movieid" : 2291, "rating" : 2, "timestamp" : 957160526 }, { "movieid" : 2692, "rating" : 5, "timestamp" : 957160301 }, { "rating" : 3, "timestamp" : 957160441, "movieid" : 457 }, { "movieid" : 480, "rating" : 1, "timestamp" : 957160561 }, { "movieid" : 1358, "rating" : 3, "timestamp" : 957160184 }, { "movieid" : 1374, "rating" : 5, "timestamp" : 957160348 }, { "movieid" : 1376, "rating" : 3, "timestamp" : 957160382 }, { "timestamp" : 957160463, "movieid" : 1387, "rating" : 5 }, { "movieid" : 858, "rating" : 3, "timestamp" : 957160382 }, { "movieid" : 2355, "rating" : 4, "timestamp" : 957160538 }, { "movieid" : 1721, "rating" : 3, "timestamp" : 957159896 }, { "rating" : 5, "timestamp" : 957160333, "movieid" : 3347 }, { "movieid" : 3360, "rating" : 3, "timestamp" : 957160333 }, { "movieid" : 3361, "rating" : 4, "timestamp" : 957160150 }, { "movieid" : 2716, "rating" : 4, "timestamp" : 957160213 }, { "movieid" : 1784, "rating" : 5, "timestamp" : 957160548 }, { "movieid" : 1036, "rating" : 5, "timestamp" : 957160184 }, { "movieid" : 2797, "rating" : 5, "timestamp" : 957159935 }], "numRatings" : 53 }

db.users.find({"name" :"Tracy Edward","movies.movieid":1210},{movies$:1})

{ "_id" : 5951 }

Question 27

Find how many users have rated the movie Untouchables, The with a note of 5.

db.movies.find({})

db.movies.find({"title":"Untouchables, The"})

{ "_id" : 2194, "title" : "Untouchables, The", "genres" : "Action|Crime|Drama", "year" : 1987 }