Lab 2: Starting with MongoDB

Due date: January 25, 11:59pm.

Lab Assignment

Assignment Preparation

This is an individual lab. I expect every person to complete it without consulting others.

This is a short lab to give you some familiarity with MongoDB's `db.<collection>.find` command. To complete this lab, you will need to create a simple Javascript program, which, largely will consist of your queries and the loops over the cursor printing the results.

Data

In this lab you will write a number of MongoDB queries extracting data from your MovieSurvey JSON collection.

For this lab **we are not providing you the data on which your queries are to be run**. Instead, you will build queries that return correct answers on **any** instance of a collection of a given type. You will test your queries by building specific collections out of the objects constructed by your JSON generators, running the queries on them, and ensuring that the queries produce correct answers. For this lab, there is no need to generate large collections of JSON objects. You can successfully test everything using collections of size 50-100 objects. Some queries can be successfully tested on even smaller collections.

General notes. Create a script (use any language you want), that takes as input a file containing a JSON document array, and produces as output a Javascript program that inserts the document into a database collection with a specified name (for both the database and the collection). This should include MongoDB authentication.

ThghtShre data. Generate one or more test JSON collections of ThghtShre JSON objects. Use collection named `survey` to store MovieSurvey objects in your MongoDB database. Use the script above to convert each JSON array produced by your generator into a Javascript program that creates/inserts the documents from the array into MongoDB. Use as many or as few collections as you need to debug your queries.

Queries

The main objective of this lab is for each of you to get comfortable using MongoDB's `find()` command. To that extent, you will write a few MongoDB queries.

Query preparation and submission. For each dataset, you will submit your queries in two separate ways:

1. **Text file.** Create text files `movieSurvey.mongo` and include all your queries there. Each query must be on its own lines, prefaced with a Javascript comment line specifying the query number and with at least one empty line between queries. The header of the file must contain one or more Javascript comment lines with your name and other information about the file. The expected format is something like this:

```
// CSC 369. Lab 2
// Alex Dekhtyar
// ThghtShre dataset

// Query 1
db.thought.find(...)

// Query 2
db.thought.find(...)

...

//end of queries
```

2. **Javascript program.** Create a Javascript program that connects to the MongoDBserver. For each query, the program shall print its number, the query itself, followed by the results obtained from running the query on the collection. Name your program `movieSurvey.js`.

Queries

Write MongoDB `db.<collection>.find()` queries that produce answers to each of the questions below. Each question must be answered with exactly one `find()` command.

1. Report all female respondents whose score for `"Avatar"` is 6.0 or above.
2. Report the ratrings of all denizens of North Eastern state (as defined in Lab 1-2) who have at least a BS degree, and at least $60,000 of annual income.
3. Find the five people who rated `Godfather` the highest.
4. Find the total number of people who rated `"Rocky"`, `"Saw"` and `"Suicide Squad"` below 4.0 (i.e., rated each of these three movies below 4.0).
5. Report the names and the ages of all California residents who prefer `"Princess Bride"` to `Sleepless in Seattle"`.
6. Find the six oldest fans of `"Star Wars:  A New Hope"`. A person is a fan of a movie if their rating for the movie is 8.0 or above.

Note: You can use the concordance between the movie position in the `ratings` array and the movie name in your queries. For example, to access information about the rating of `"Memento"` you can use the `ratings.2` identifier.

Submission

Submit the following artefacts:

- Your script for creating Javascript data insertion programs.
- Data insertion Javascript programs for one data collection for `MovieSurvey` dataset (The size of the dataset need not exceed 50-100 objects).
- `movieSurvey.mongo` text file with queries.
- `movieSurvey.js` Javascript program with queries.
- `README` file.

All submitted files must contain your name on them.

Submit all your code in a single archive (zip or tar.gz).

Use `handin` to submit as follows:

```
$ handin dekhtyar lab3 <FILES>
```

Good Luck!