Lab 2: Starting with MongoDB

Due date: January 16, 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 CRUD operations and with the `db.<collection>.find` command.

Assignment

Each of you have access to the MongoDB server running on `ambari-head.csc.calpoly.edu`. On this server, each of you is given a database under your own name (i.e., the name of the database coincides with your MongoDB account name) for which you are granted the dbOwner role - i.e., full control over the content. As part of this lab assignment you will create a number of collections in your database, and will demonstrate the contents of these collections.

Part 1. Data Upload. Create nine collections in your database as described above. One of the collections will store a small slice of data from the Lab 1 Iowa liquor sales dataset. Another collection will store the Iowa county population data. The remaining seven collections will store the outputs of your seven Lab 1 programs. Specifically, create the following collections in your database:

Collection name	Content
`sales`	First 10 JSON objects from `iowa.json` (Lab 1 file)
`counties`	The entirety of `counties.json` file
`summary`	The full output of your `liquorAggregate.py` program
`stores`	The full output of your `storeAccounting.py` program
`network`	The full output of your `distribution.py` program
`vodka`	The full output of your `vodkas.py` program
`liquor`	The full output of your `categorySales.py` program
`leader`	The full output of your `leaderboard.java` program
`stats`	The full output of your `countyStats.java` program

You can conduct data upload in any way you want, as long as you are using actual MongoDB `db.<collection>.insert()` (or similar) commands.

Part 2. Dataset display. Using Linux's `script` command (see `man` page for it for instructions on how to use it - it is not too complicated, and we will have in-class demos that use this command), create a log of a MongoDB session during which you do the following[1]:

- Authenticate your account against the `admin` database.
- Switch to the database carrying your account name.
- Report the full contents of each of the nine repositories you created.
- exit MongoDB.

Call your log file `mongo.<loginId>.log`, where `<loginid>` is your MongoDB account name.

Part 3. Four simple queries. Using the Linux's `script` command, create a log file a MongoDB session during which you do the following[2]:

- Authenticate your account against the `admin` database.
- Switch to the database carrying your account name.
- Write a MongoDB `find()` query that outputs the list of counties (as unmodified JSON objects from the `counties` repository) with populations greater than 100,000 people[3].
- Write a MongoDB `find()` query that outputs the total number of liquor vendors that provide wholesale services to Iowa's liquor stores (this is best done using `network` collection).

[1] You are allowed to access the log file after it is complete, and remove your password from the authentication command(s) you used in the session.

[2] You are allowed to access the log file after it is complete, and remove your password from the authentication command(s) you used in the session.

[3] You may need to do some data conversion prior to creating the `counties` repository to make this work.

- Write a MongoDB `find()` query that reports sales statistics from the `stats` collection for Polk County.
- Write a MongoDB `find()` query that prints all liquor purchases of 10 bottles or more that were made from the vendor `"Diageo Americas"`. Run this query against the `sales` collection. Report only the following attributes: store name, item description, bottles sold, sale in dollars.

Call your log file `queries.<loginId>.log`, where `<loginId>` is your mongoDB account name.

Submission

Submit the following artefacts:

- `README` file containing your name and a detailed description of the process which you used to insert the data into your database.
- `mongo.<loginId>.log` file.
- `queries.<loginId>.log` file.

Use the following `handin` command.

```
$ handin dekhtyar lab2 <files>
```

Note, `ambari-head` does not have `handin` installed. Therefore, you will need to transfer your log files from the `ambari-head` server to one of the `unixN.csc.calpoly.edu` machines.

Good Luck!