Getting Started

Logistics

This project is due **Wednesday, 05/01/2024 at 11:59PM PDT (GMT-7)**. It is worth 5% of your overall grade in the class. 100% of your grade will come from the public tests released with the data set.

Like Project 1 this project **must be completed individually.** Note that while this means we expect you to write all your queries on your own, all of the following are **permitted** under our academic integrity guidelines:

- Discussion of approaches for solving a problem.
- Giving away or receiving conceptual ideas towards a problem solution.
- Discussion of specific syntax issues and bugs in your code.
- Looking at another student's code for the sole purpose of helping that student debug
- Using small snippets of code that you find online for solving tiny problems (e.g. Googling “number to string mongo” may lead you to some sample code that you copy and paste into your solution). Such code should always be cited with relevant code comments.

Prerequisites

There are no hard prerequisites for this project. The spec will walk you through writing a query in Mongo's syntax. However, you may find watching the NoSQL lectures (after they're released) to help contextualize how Mongo differs from the traditional SQL databases we've been working with for the majority of this semester.

Fetching the released code

The GitHub Classroom link for this project is in the Project 6 release post on Edstem. Once your private repo is set up clone the Project 6 skeleton code onto your local machine.

Required Software

MongoDB v4.4

We'll be exploring the document-oriented database MongoDB in this project. Check if you already have a copy installed by running `mongo --version` in a terminal. If you already have it installed you should see output similar to the following:

```
> mongo --version
MongoDB shell version v4.4.1
Build Info: ...
```

If you don't already have MongoDB on your machine, follow the instructions for your platform:

Windows

Follow the instructions here to install MongoDB on Windows. You'll also need to install Database Tools from here. For both of them, **download the 4.4 versions, not the 6.0**.

Once you have everything installed you'll want to locate the location of the mongo shell and mongoimport binaries. Confirm the location of the binaries at the following spots:

- The mongo shell binary (mongo.exe) should be located at `C:\Program Files\MongoDB\Server\4.4\bin\`.
- The mongoimport binary (mongoimport.exe) should be located at `C:\Program Files\MongoDB\Tools\100\bin\`. If you can't find it at that exact location, check other directories under `C:\Program Files\MongoDB\Tools\`. If your file has a long name like `windows-x86-64-bit-mongoimport.exe` then rename it to just `mongoimport.exe`

Add the two directories to your PATH. To edit your environment variables on Windows 10, use the following steps:

1. Open up search and type in "Edit the system environment variables"

2. Open that up and click "Environment Variables..." near the bottom
3. Click "Path" under user variables (top half of the screen) and click edit
4. Click "New" on the top right and add C:\Program Files\MongoDB\Server\4.4\bin\
5. Repeat the same process for database tools with the appropriate PATH

If everything went successfully you should be able to run `mongo --version` and `mongoimport --version` successfully in Git Bash, or `mongo.exe --version` and `mongoimport.exe --version` in other shells.

MacOS

If you don't already have it, install Homebrew. Then, in a terminal, run the following:

```
brew tap mongodb/brew
brew install mongodb-community@4.4
brew services start mongodb-community@4.4
```

If you run into a CompilerSelectionError, run `xcode-select --install` and repeat the commands above. Check that everything is installed by running `mongo --version` and `mongoimport --version`. If both of these commands work then you should be good to go.

- If your Mac is M1-based or later, and you install mongodb through homebrew, it's likely that you will get an error saying `mongo: command not found`. In the output of `brew install mongodb-community@4.4`, you should have gotten a message giving a command on how to do this. It should look similar (not necessarily the same) to

  ```
  echo 'export PATH="/opt/homebrew/opt/mongodb-community@4.4/bin:$PATH"' >> ~/.z
  ```

  Run that command, open a new terminal session, (this is important!) and your error should be gone.

If your version of Mac can't support Mongo 4.4 then we recommend you use the Docker approach to complete this assignment. Otherwise, if it doesn't support Docker either then you can use Mongo 4.2 instead. This will work the same as 4.4 except the following functions recommended in the spec will not be available, so you'll have to modify the advice slightly:

- `{$first: <array expression>}` will have to be replaced with `{ $arrayElemAt: [ <array expression>, 0 ]}`
- `$isNumber` will have to be rewritten as an or statement based on the the fields type

Linux

Follow the instructions here for your appropriate platform. If something breaks during the installation process and you can't run `mongo` and `mongoimport --version`, follow the instructions in the next section (Docker) to get a docker container with mongo and python pre-installed.

Docker

If you're on MacOS/Linux and ran into issues with installing mongo directly on your host machine, we recommend using a docker container with mongo and python pre-installed instead. To use our Docker image you'll need to install Docker Community Edition ("CE") on your machine.

- To install Docker CE on Mac open the Docker getting started page, stay on the "Developer" tab, and click the button on the right to download the installer for your OS. Follow all the instructions included.
- To install Docker CE on Linux, open the Docker docs, and click the appropriate link to find instructions for your Linux distro.

Confirm that Docker is installed by running `docker --version` on the command line. If it works, you should be good to go. From the root of your project directory, run `pwd` to get the path to your present working directory. Then carefully run the following command (be sure to replace `/path/to/project/directory` with the path from the previous step):

```
docker run --name mongo186 -v "/path/to/project/directory:/proj6" -it
chriskw/mongo186
```

This will download an image of a container with mongo and python preinstalled. You should see output like the following:

```
$ docker run --name mongo186 -v "/replace/this/accordingly:/proj6" -it chriskw/m
Unable to find image 'chriskw/mongo186:latest' locally
latest: Pulling from chriskw/mongo186

(a bunch of downloads should happen here)

about to fork child process, waiting until server is ready for connections.
forked process: 10
child process started successfully, parent exiting
student@a2ba045477a4:/$
```

This should bring you into a container with the necessary requirements. Run `cd /proj6` to enter the proj6 directory. All of the files on your host machine should be present. Changes on your host machine (for example, using a text editor like VSCode/Sublime) should be visible from within this container as you work through the project. If everything went smoothly, you should be able to go to the next section Extract the data set.

If you exit the container and wish to access it again, you can run the command `docker start -ai mongo186` to re-enter the container.

If your proj6 directory is empty, you most likely provided an invalid path when you ran `docker run`. In this case, run the following two commands: `docker kill mongo186` followed by `docker rm mongo186`. Afterwards, rerun the `docker run` command from above, making sure to replace the string after the `-v` accordingly. Followup on Edstem if you run into trouble with this step.

Python

You'll need a copy of Python 3.5 or higher to run the tests for this project locally (the same as used in Project 1). You can check if you already have an existing copy by running `python3 --version` in a terminal. If you don't already have a working copy download and install one for your appropriate platform from here.

Extract the data set

Download the data set from the class drive here.

Unzip the `data.zip` file inside your `sp24-proj6-yourname` directory. You should now have a `data/` directory in your `sp24-proj6-yourname` directory.

Afterwards, try running `python3 load.py`. You should see output like the following:

```
> python3 load.py
2021-04-11T08:30:08.567-0700    connected to: mongodb://localhost:27017/
2021-04-11T08:30:08.567-0700    dropping: movies.credits
2021-04-11T08:30:14.590-0700    45475 document(s) imported successfully. 0 docum
2021-04-11T08:30:14.609-0700    connected to: mongodb://localhost:27017/
2021-04-11T08:30:14.610-0700    dropping: movies.movies_metadata
2021-04-11T08:30:15.551-0700    45406 document(s) imported successfully. 0 docum
2021-04-11T08:30:15.562-0700    connected to: mongodb://localhost:27017/
2021-04-11T08:30:15.562-0700    dropping: movies.keywords
2021-04-11T08:30:16.569-0700    43986 document(s) imported successfully. 0 docum
2021-04-11T08:30:16.580-0700    connected to: mongodb://localhost:27017/
2021-04-11T08:30:16.581-0700    dropping: movies.ratings
2021-04-11T08:30:17.968-0700    99958 document(s) imported successfully. 0 docum
```

Unfiltered Dataset

The dataset used for this project is a subset of the original dataset -- we've filtered out overtly offensive and inappropriate keywords and movie descriptions, as well as any keywords that appear less than 15 times. If you want to access the original dataset without these filters applied, the unfiltered version is here. Note that the expected output of the project is based on the filtered version, so you'll need to use the filtered version to complete the project.

Running the tests

If you followed the instructions above you should now be able to test your code. Navigate to your project directory and try using `python3 test.py`. You should get output similar to the following:

```
> python3 test.py
q0    FAIL: Empty output
q1i   FAIL: Empty output
q1ii  FAIL: Empty output
q1iii FAIL: Empty output
q1iv  FAIL: Empty output
q2i   FAIL: Empty output
q2ii  FAIL: Empty output
q2iii FAIL: Empty output
q3i   FAIL: Empty output
q3ii  FAIL: Empty output
```

If so, move on to the next section to start the project. If you see `ERROR` instead of `FAIL` create a followup on Edstem with details from your `your_output/` folder.

Debugging Issues with GitHub Classroom

Feel free to skip this section if you don't have any issues with GitHub Classroom. If you are having issues (i.e. the page froze or some error message appeared), first check if you have access to your repo at `https://github.com/cs186-student/sp24-proj6-username`, replacing `username` with your GitHub username. If you have access to your repo and the starter code is there, then you can proceed as usual.

404 Not Found

If you're getting a 404 not found page when trying to access your repo, make sure you've set up your repo using the GitHub Classroom link in the Project 6 release post on Edstem.

If you don't have access to your repo at all after following these steps, feel free to contact the course staff on Edstem.

Previous
Project 6: NoSQL

Next
Your Tasks

Last updated 3 months ago