BIT2008A/ITEC2000A: Multimedia Data Mangement
Fall 2022

Lab 9

MongoDB

In this Session we practice supplementary MongoDB syntax. Please follow the step-by-step instructions.

i) First, we want to import two collections into our database. There are two main formats namely *BSON* and *JSON*, to import/export data in MongoDB and many other technologies. You can learn more about those formats here. [**not** included in the course evaluation.] We use *users* and *movies* collections for this Lab, and you can find them under "Lab 10/data" on Brightspace. You may use a cloud cluster or a Docker container to set up your mongoshell. If you need a recap, you can find the instructions under "Week 12".

- If you are using the cluster, you have probably used a command like below to initiate a connection:

```
.\mongosh.exe "mongodb+srv://cluster0.frvrb0o.mongodb.net/<<db name>>" --
    apiVersion 1 --username <<user name>>
```

Now you can go to this download page and scroll down to *MongoDB Command Line Database Tools*. Please consider Fig. 1 showing the correct tool to download. Afterward, you should

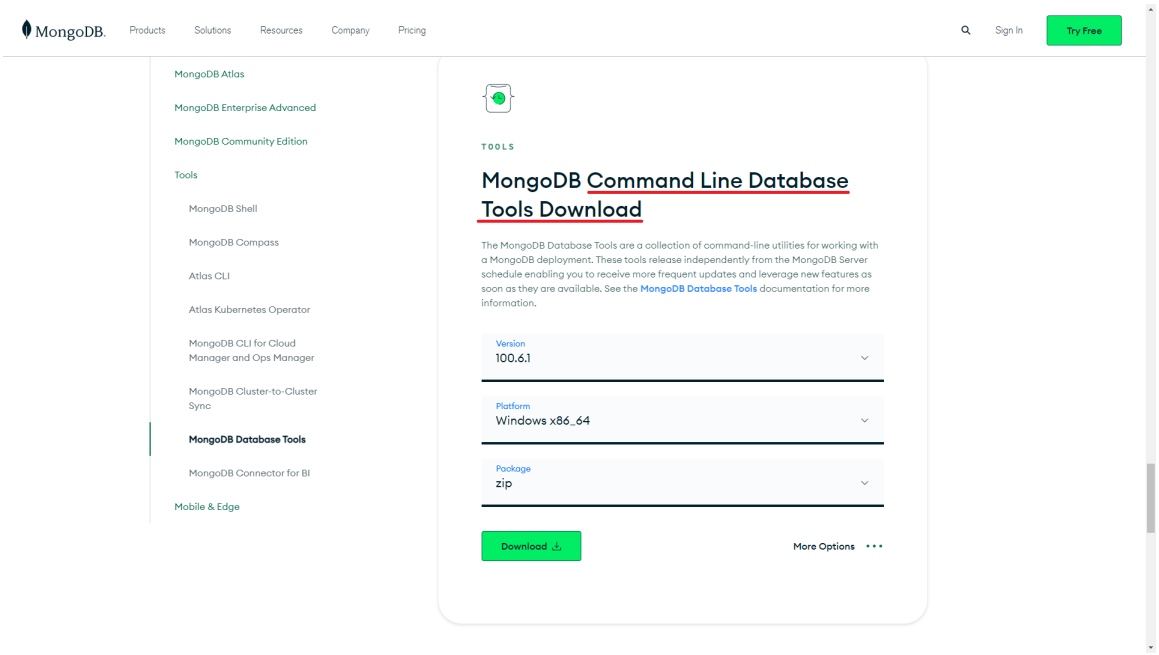

FIGURE 1. Download command line tools.

extract the zip file in your project folder the same way you did for *mongosh.exe*. Consequently, your project folder that you open your terminal in, should look like Fig. 2. The files that we actually care about are underlined. Now, we are ready to import our collections into the database. You can use the command below to import *users* and *movies* collections. Please remember that it only shows the pattern of that command and you should modify it to fit to your URI (the connection link), database, username, etc.

```
.\mongorestore.exe "mongodb+srv://cluster0.frvrb0o.mongodb.net" --
    username admin --drop -d "Bit" -c "movies" .\movies.bson
```

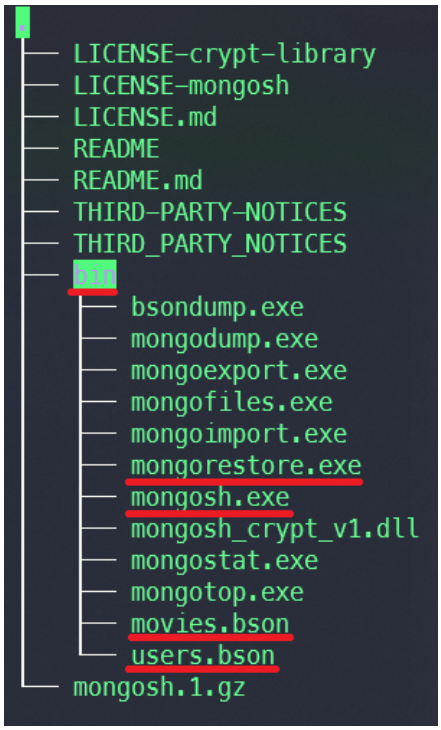

FIGURE 2. Project folder.

When you successfully import those collections, you can find them as shown on Fig. 3. *users* and *movies* collections should contain 183 and 45993 documents, respectively.

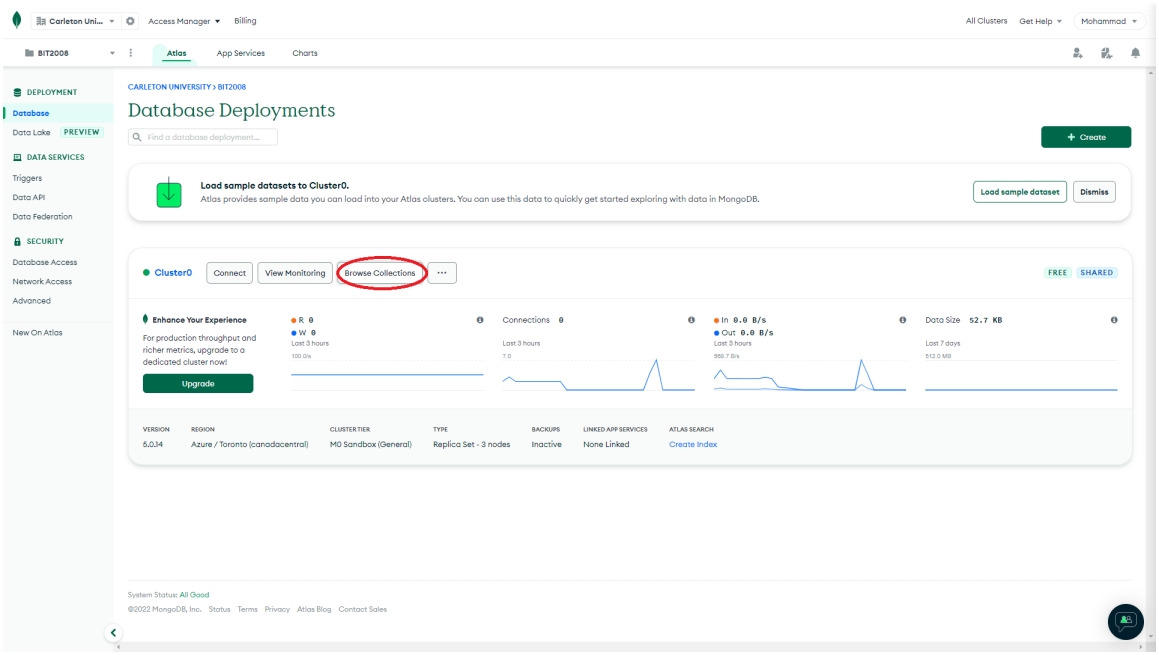

FIGURE 3. Browsing your collections.

- If you are using the Docker container solution, you should create a new folder for your project, then get *docker-compose.yml* and *Dockerfile* from Brightspace under "Week 12". Then put your BSON files in the same folder and do *docker compose up* like before. After the containers run, execute the command below in a docker terminal from *mongo* container:

```
cd code
mongorestore --authenticationDatabase admin -u root --drop -d <<db name>>
     -c <<collection>> <<.bson file>>
```

You should load *movies* and *users containers*. As you know, to run a mongoshell, you need to execute this command:

```
mongosh -u root -p example
```

You can open your browser to *http://localhost:8081/* to investigate the documents visually via MongoExpress as Fig. 4 shows.

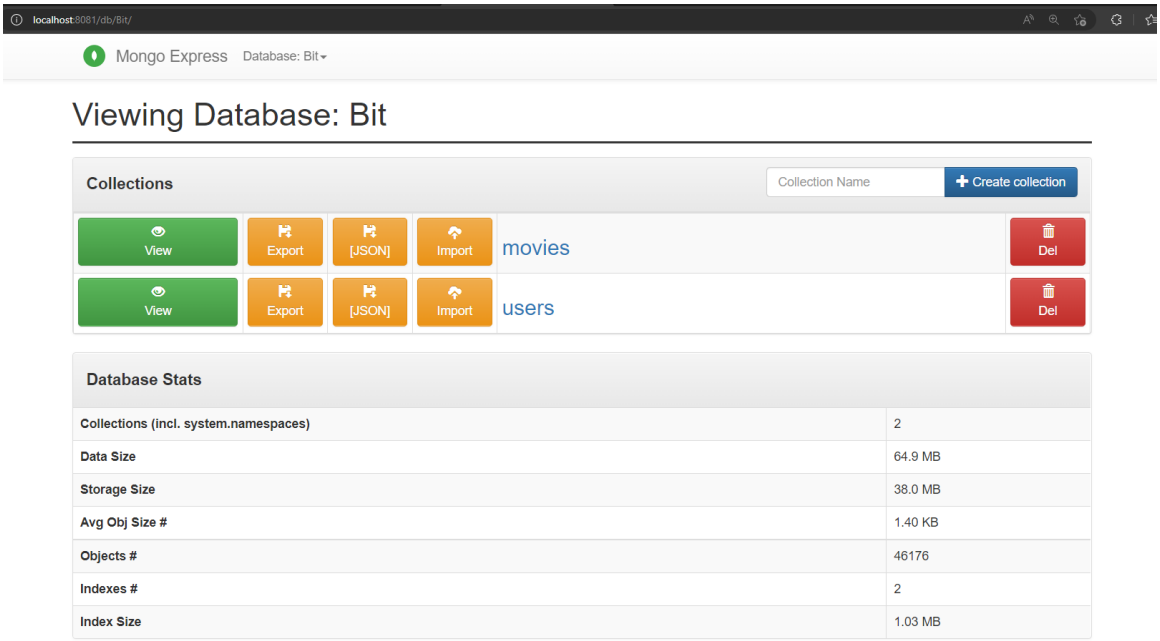

FIGURE 4. Mongo Express showing the collections.

ii) You can limit the number of returned documents by calling *aggregate* on your collection and passing *$limit* as an argument. Considering that, write a query to get 10 documents from the *users* collection.

iii) You can define patterns to look up as query filters using *$regex* argument. for instance *$regex:"com$"* returns all the attribute values ending with "com". Having that in mind, write a query that counts the number of users whose email address is in *@gameofthron.es* domain.

iv) The *movies* collection has a array attribute for each document called **genres**. We can apply aggregation on the elements of an array attributes via *$unwind*. Using that operator, write a query that shows the name of those genres along with their occurrence descendingly ordered.

v) Find the title of the movies that are *released* in from 1998 until 2015.

vi) You can use dot operator to move to different inner attributes of a nested document. To test that, find the movies with an IMDB score above 9 and show their casts.

vii) Remove *plot* and *fullplot* attributes from movie documents. This will cut the size of your collection by half.