Homework Assignment #2 Setup

(Apache Cassandra)

For this assignment, we will be using an online service called DataStax Astra which provides a cloud-based implementation of Apache Cassandra. We will also be using your PostgreSQL database from Assignment #1 to export CSV files, so make sure you keep that on hand.

1. Go to http://astra.datastax.com/ and sign up for a free Astra account. (Select **"Learning Cassandra"** as the reason for using DataStax Astra.)
2. After verifying your email, you will be prompted to **"Try Serverless Today"**. The pricing plan page will appear, where you will choose the "Pay as you go" option on the top right by clicking on **"Get Started"**. You do not have to pay for this service, so do not worry. You are allowed up to $25 for free for Astra service use, but you should not reach that limit while doing this assignment!
3. The "Create a Database" page will render. Go ahead and create a database called **cs122d-spring** with a keyspace called **ShopALot**. For the "Provider and Region" section, choose **North America** and **AWS**.
   Note: your database might take a few minutes to be successfully created. While it is being created, you will see "pending" next to the database name.
4. Once you receive confirmation that your database is created, head over to the dashboard by clicking on your database name and open the CQL console. This is where you will be running your queries. To test out your setup, let us create a keyspace called "**Hoofers**", which we will load data into later. Use the **CREATE KEYSPACE** CQL statement to do so.
5. For purposes of loading data into Datastax Astra, we will be using a software tool called the **DataStax Bulk Loader** instead of using the "Upload Data" button on the dashboard. This is to make sure that you learn how to manually create tables in CQL before loading data, and also to experience the task of uploading bulk data into a database service in the cloud (a skill whose relevance is increasing over time).
6. Download DSBulk from the following link: https://downloads.datastax.com/#bulk-loader and unpack the downloaded distribution.
   Note: DSBulk requires a Java Runtime Environment. (If you do not have a JRE, make sure to download that as well, but most of you probably have Java set up already.)
7. Add the downloaded file to your PATH so that you are able to use the dsbulk command. This process can differ depending on your OS.
   a. For MAC/Linux users: on the command line run the following:
      export PATH=path-to-unpacked-location/dsbulk-1.8.0/bin:$PATH
      (You might want to add this to your favorite shell's .xxxrc file.)

   b. For Windows users: this link might help https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/

8. Ensure that dsbulk is working properly by running the command:

```
$ dsbulk --version

DataStax Bulk Loader v1.8.0
```

Let’s test out the workflow of loading data into Astra by first exporting data from PostgreSQL then importing it with DSBulk by using the ‘Hoofers’ data previously given.

9. If you haven’t already done so, create a database called “**Hoofers**” in PostgreSQL. Run the following SQL script in PostgreSQL to create the tables and insert the data: https://grape.ics.uci.edu/wiki/asterix/attachment/wiki/cs122d-2021-spring/HoofersDB.txt.
10. Export the “**Boats**” table into a CSV file; here are some useful hints on how to do so in PostgreSQL:

```
-- Suppose you want to export a table named Person into a csv file called
persons.csv, this file will be created at runtime --
\copy Person to '/desired-path/persons.csv' DELIMITER ',' CSV HEADER;

-- Now suppose you want to export the results of a query on the Person
table into a csv file called results.csv, this file will be created at
runtime --
\copy (SELECT * FROM Person P WHERE P.person_id = ‘1234’) to
'/desired-path/persons.csv' DELIMITER ',' CSV HEADER;
-- The process would be the same for exporting the results of a JOIN query
--
\copy (SELECT * FROM Person P, Username U WHERE P.uid = U.uid) to
'/desired-path/persons.csv' DELIMITER ',' CSV HEADER;
```

Now that we have the CSV file, we can begin the process of loading data into Astra.

11. In the Astra CQL console, create a table called “Boats” by translating the PostgreSQL CREATE TABLE statement given in the script.
12. Follow the instructions in the answer to this post https://community.datastax.com/questions/10901/need-dsbulk-astra-examples-concrete-ones.html?childToView=10951

13. You should be armed and ready to start this assignment! Good luck! Navigate to the course website to find the Cassandra documentation.