Lab Assignment 4: SQLAlchemy Core

In this assignment we will carry out work similar to that of assignment 2, but working with SQLAlchemy instead, and writing what amounts to SQL queries. You should download this Python script from GitHub[1] and store it in the same location where your keys.json file is. You will need to add a key called vault with value an object with keys username, password, server and schema. It would look something like this:

```
{
   "twitter": {
      "key": "....",
      "secret": "...."
   },
   "vault": {
      "username": "skiadas",
      "password": "....",
      "server": "vault.hanover.edu",
      "schema": "skiadas"
   }
}
```

You should use your own login for username and schema name, and type in your own password. Keep the server value at vault.hanover.edu as above.

You will be submitting two files. The one is an SQL script you should start. The other is the Python script you just downloaded, with your additions at the end. You should provide two solutions for each question:

- You should first work out the problem in the SQL script, working with MySQL-Workbench. All problems will be appropriate SELECT queries. You can use JOINs when appropriate, or you can do everything exclusively with WHERE.
- Once you have that working, you should transport that script into SQLAlchemy terms.
- Throughout, you should only need to work with the SQL tables. When working on the SQL side, they are called tw_users, tw_tweets, tw_hashtags and tw_mentions. When working on the Python side, they are called dbusers, dbtweets and so on.

Here are the questions.

1. Create a table with three columns: The first is the tweet id, the second is the tweet's user's name and the third is a count of the number of hashtags in that tweet (called no_hashtags).
2. Create a table that has two columns: The first is the user name (Clinton or Trump) and the other is the average number of hashtags per tweet.
3. Create a table that has three columns: The first is a day of the week (Monday, Tuesday etc), the second is a candidate name, and the third is the number of tweets the candidates sent on that weekday. There should be a total of at most 14 rows, one for each combination of a day of the week and a candidate. It is OK

[1] https://github.com/skiadas/DataWranglingCourse/blob/gh-pages/assignments/assignment4.py

if some rows are missing if there were no tweets sent on that day. The function DAYNAME()[2] will come in handy. Also use the function DAYOFWEEK() from the same documentation page to order the resulting data by day of the week.

4. Produce a table with two columns: The first has a user's name, and the second has the number of times that user was mentioned. Order the results by descending order on this number of times.
5. Produce a similar table as in the previous problem, but now with an extra column for the candidates. Each row would then contain a user's name, a candidate's name, and the number of tweets that the candidate did that mentioned that user. Order the results alphabetically by user name and then candidate name. You will need to have two copies of the user's table, along with other tables, in your joins. One copy for the user mentioned, and the other for the writer of the tweet. Make sure to give the columns of the result table distinct names.

You should submit your completed SQL and Python scripts as an email attachment to me. The name of the files should include your first and last name, in addition to the assignment's number. It should contain no whitespaces.

[2] http://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_dayname