INF 551 – Fall 2016

Homework #3

Due: October 31
100 points

In this homework, you are provided with a US presidential election data set. The data set contains an JSON document for every election year from 1900 to 2012. The document states the candidates for each party (Democratic and Republican) and the number of votes they received from each state. For example, in "2012.json", we see that Obama and Romney were the candidates for the democratic and republican parties respectively. Furthermore, Obama did not receive any electoral vote from the state of Alaska (AK) but 122,640 popular votes from the state. You can ignore the votes for other candidates (i.e., not from Democratic and Republican).

You are asked to design a database called "inf551" (with user name: "inf551" and password "inf551), load the data in JSON into tables in the database, and permit searching of election data. The details are as follows.

1. [30 points] Create tables in the database for storing the data. Create indices that help speed up the search (see below). Explain why your design is good. Submit a file "create.sql" that includes the script for creating tables and indices, and also the explanation (included as comments). Refer to this link for more details on the comment syntax).

2. [30 points] Write a Python script "load.py" that accepts a directory storing the JSON files, loading the data in the files (via "insert into …" statements) into the database. (Refer to the example script provided in the lecture for examples.)

   Execute: **python load.py <path>**
   - arg1: <path> - string, the path of data directory storing the JSON files

   Example: python load.py "data"
   It will load all files under the "data" directory into the database you designed. See the provided data set for more details, which has the data for each year stored in a JSON file under the "data" directory.

3. [40 points] Write the following search scripts.

   a. [20 points] Write a Python script "candidate.py" that accepts a year and returns the name of candidate for each party for the specified year.

   Execute: **python candidate.py <year>**

- arg1: <year> - string (NOTE: not integer!), in 4-digit format with quote marks. No need to handle other year formats.

Output: **<party>: <candidate>**
One line for each candidate, order not restricted.

Example: python candidate.py "2012"
Output:
democrat: Obama
republican: Romney

b. [20 points] Write a Python script "search.py" that accepts a year, a candidate name, and a state as the input, and output the electoral and popular votes for the candidate.

Execute: **python search.py <year> <candidate name> <state>**

- arg1: <year> - a string (NOTE: not integer!), in 4-digit format with quote marks. No need to handle other year formats.
- arg2: <candidate name> - a string. Need to accept search keywords in different cases (e.g., "obama" and "Obama" are the same), do not need to consider typo in names.
- arg3: <state> - a string. Need to accept search keywords in different cases (e.g., "AK" and "ak" are the same). You may assume that states always come in abbreviated form (that is, you do not need to recognize Alaska and so on).
- The query will take exactly 3 arguments in the order defined above, no need to consider missing arguments or reordering.

Output: **EV: <# of electoral vote>; PV: <# of popular votes>**
One line, EV comes before PV with ';' separating the two.

Example: python search "2012" "Obama" "AK"
Output:
EV: 0; PV: 122640

Note that for each of the above search scripts, you are to turn user keyword searches into appropriate SQL queries, obtain results from the database, and present them to the user.

Output format MUST be exactly the same as defined above and print to the screen. NO extra information. Do NOT write output to files.

Submission:

Please submit 4 files:

- <firstName>_<lastName>_create.sql
- <firstName>_<lastName>_load.py
- <firstName>_<lastName>_candidate.py
- <firstName>_<lastName>_search.py

DO NOT make them into zip files.