Assignment #4

CS106E Spring 2022, Young

We will be giving you two weeks for this assignment, as its due date would otherwise fall on the same day as the midterm. It is one of the more involved assignments.

In this assignment we begin by working with SQLite on the Stanford servers using a ***Command Line Interface***. This is a common type of program interface that programmers and computer administrators use. Next, you'll experiment with client-side and server-side programming. In order to do this part of the assignment you will need to first request CGI access as described in the "h06 Hands on with PHP" handout. You may also want to review how to get files onto the Stanford server and how to access them on the web (also described in that handout).

This assignment is due before the start of class Wednesday May 11th – submit it on Gradescope (Entry Code: 2R7JKV) before 1:30pm. Additional Administrative Details can be found at the end of this document.

Honor Code Reminder

For the purposes of this class, if someone (other than your partner, a TA, or the instructor) looks at your HTML, CSS, or SQL code, or if you look at another student's HTML, CSS or SQL code, you have received too much assistance.

If in doubt, or if you're concerned you have received too much assistance, explicitly document the help that you received. If you have specified in writing the help which you've received on an assignment, you will never be considered as having violated the CS106E honor code policy (although we may reduce the number of points you receive on the assignment).

Unix and SQL

The following questions will involve using a *Command Line Interface* to write SQL queries to perform the requested lookups on a database that we have provided. Command Line Interfaces are often used by programmers and computer administrators.

First, copy the provided hw4.db file to your home directory on the Stanford servers, using either SecureFX (for Windows) or Fetch (for Mac). Then, connect to myth.stanford.edu using either SecureCRT (for Windows) or the ssh command in Terminal (for Mac) and open the hw4.db file in sqlite3. For instructions on how to connect to Stanford servers and run SQLite, refer to the provided handout (h08 SQL.pdf) on Canvas.

hw4.db contains a table named `cities` with columns `city`, `continent`, and `population`. For each lookup question, **provide the SQL query that you use, along with the data returned by your query.**

1. What is the CREATE TABLE query you would use to create `cities`? You can run "SELECT * FROM cities;" to see what the table looks like (and the default order of the columns). See the handout for SQLite commands to better format the output.
2. What INSERT query would you use to add a row for the city Seattle in North America with population 3,700,000?
3. Lookup the name, continent, and population of cities on continents beginning with an "A".

4. Lookup only the name and population of cities with a population between 1,000,000 and 5,000,000 (inclusive).
5. Lookup the name, continent, and population of cities on any continents ending with "America" and with a population of at least 5,000,000. Sort the output by the city's name in alphabetical order.

Client-Side vs. Server-Side Processing

1) Try running the server-side version of the Compound Interest program. If you've downloaded the file directly into the top level of your cgi-bin directory, the url should be:

   `http://cgi.stanford.edu/~yourSUNetID/compounding.html`

   where the yourSUNetID is replaced with your actual SUNetID. You do need to leave the ~ tilde in, as that's part of the UNIX syntax.

   Try running the client-side version of the same program at:

   `http://cgi.stanford.edu/~yourSUNetID/client-compounding.html`

   Do you find either of these easier to use?

2) Go to the initial webpage of the server-side version of the program:

   `http://cgi.stanford.edu/~yourSUNetID/compounding.html`

   Disconnect from the Internet. Try hitting the submit button. What happened? Reconnect the internet. Go to the initial webpage of the client-side version of the program:

   `http://cgi.stanford.edu/~yourSUNetID/client-compounding.html`

   Disconnect from the Internet. Try hitting the calculate button. What happened?

   What are the implications here?

Server-Side with SQL Experiments

Go to the webpage:

`http://cgi.stanford.edu/~yourSUNetID/party-invite.html`

again replacing yourSUNetID with your actual ID. Go ahead and tell the webpage what you're bringing to the party. Go to the webpage:

`http://cgi.stanford.edu/~yourSUNetID/party-summary.php`

You should see a list of everything everyone is bringing.

Now, using the SQLite skills you learned in the last HW, log into the Stanford server and access the database file party.db in your CGI directory. Run

```
sqlite3 party.db
```

then in the SQLite monitor do a

```
SELECT * FROM party;
```

Write down the results.

Website Observations

Go to the following websites. For each website:

- List three different aspects or features of the homepage that require client-side processing. If you can't find three, go ahead and write down that there are fewer than three and list any that you do find.

- Briefly describe one aspect of the website which requires server-side programming (if any exist). Here we are talking about the website as a whole, not just the homepage. Note that simply serving up static webpages is not generally considered server-side programming.

1) Visit Amazon.com

2) Visit Wikpedia.com

Choose another two websites that you visit regularly, and do the same analysis on them.

Administrative Details

You will need to turn in the following files for this assignment:

- A typed document with
  - the SQL queries and results for the lookup questions,
  - observations on Client-Side vs. Server-Side Processing,
  - results from Server-Side with SQL experiment,
  - Website Observations
- coversheet.txt with your name and the name of your partner (put None if no partner)