CMPUT 291

Assignment 4 Report

Ye Wang ye7

Xiaoyu Liu xliu2

Overview

The purpose of this project is to practice the use of SQL in a host programming language. Our task is to build a system that keeps the enterprise data in a database and to provide services to users. We should store data in a SQLite database and write code in Python. The database schema for this assignment is:

Coordinates(Neighbourhood_Name, Latitude, Longitude);

Crime_Incidents(Neighbourhood_Name, Crime_Type, Year, Quarter, Month, Incidents_Count);

Population(Ward, Neighbourhood_Name, Neighbourhood_Number, CANADIAN_CITIZEN, NON_CANADIAN_CITIZEN, NO_RESPONSE)

For users to use our program, the first thing he/she needs to do is simply that open the terminal, go the directory of the file and type "python3 prjcode.py". The the program will be executed. And the main menu will be shown in the terminal. There are totally five options for the user to choose. The first four options are some rough description about the respective questions. And the last one is for user to exit the program. The user should type number 1 to 4 and letter "E" to choose the questions he/she wants to know or quit the program.

If the user types "1", then select desired range of years and crime type, the program will display a bar chart showing the month-wise total count of the given crime type. When the user close the bar chart, a corresponding file will be saved.

If the user types "2", followed by an integer N, the program will generate a map showing the 4 most/least populous neighborhoods, along with their population count. A html file will be saved in the disk as well.

If the user types ”3", then select a range of years, a crime type and an integer N, the program will generate a map showing Top-N neighborhoods and their crime count where the given crime type occurred most within the given range. The map will be saved as html file in the disk.

If the user types "4", a range of years and an integer N, the program will generate a map showing the Top-N neighborhoods with the highest crimes to population ratio within the provided range, along with the most frequent crime type in each of these neighborhoods.

If the user types "E", the program stops.

Design

MAIN function: This function will first connect the database using CONNECT function. Then there is a while loop to let the program execute. It will print the main menu first. And then let users to input a number between 1 and 4 to select the question they desire or input "E" to exit the program. Then, there is a long "if-elif-else" statement to distinguish which question should be implemented.

CONNECT function: This function is copied from the lab sample code “sqlite3-example1.py” to connect a SQL database and run SQL commands and conn.commit() is used to commits the current transaction.

QUESTION1 function: This function is used for question 1. Firstly, read 3 inputs(startyear, endyear and crimetype) from user. If the typed is invalid, for example startyear > endyear, the program will ask user to type again. Then put these 3 values into the sql query to obtain the desired crime type and corresponding month in the range of years. Next, plot the bar plot with month on x-axis and crime count on y-axis and save the plot in the disk. If we quit the program and we reopen the program and save the file, the new file will not replace the previous file we created.

QUESTION2 function: This function is used for question 2. Firstly, read the value N from user. This will be used to find the number of neighbourhoods. For this question, we created two sql query to get both N-most populous neighbourhoods and N-least populous neighbourhoods. The two sql queries are basically the same, other than DESC was used in obtaining N-most populous neighbourhoods. After this, we generate a map and circle out the areas where are most/least populous neighbourhoods. Crimson circles shows the most populous neighbourhoods whereas blue circle shows the least populous neighbourhoods. An html file will then be saved. If we quit the program and we reopen the program and save the file, the new file will not replace the previous file we created.

QUESTION3 function: This function is used for question 3. Firstly, read the total 3 inputs(startYear, endYear and N) from user. If the typed is invalid, for example, startYear > endYear, the program will ask user to retry it. Then we put these three inputs into the query. We use SQL command to show the Top-N neighbourhoods and their crime count where the given crime type occurred most within the given range. After this, we generate a map and circle out the areas based on the ranking of crime count. Different sum of crime count will have different size of circle. An html file will be saved. If we quit the program and we reopen the program and save the file, the new file will not replace the previous file we created.

QUESTION4 function: This function is used for question 4. Firstly, read the total 3 inputs(startYear, endYear and N) from user. If the typed is invalid, the program will ask user to retry it. Then we put these three inputs into the query. We use SQL command to show the top-N neighbourhoods with the highest crimes to population ratio within the provided range. We also show the most frequent crime type in each of these neighbourhoods. We use two separate queries to solve these problems. In the first query, we only try to select neighbourhood name, rime count, the population, ratio, and location. In the second query, we use the result from the first query and select the most frequent crime type of each neighbourhoods. After this, we generate a map and circle out the areas based on the ranking of ratio. Different ratio will have different size of circle. An html file will be saved. If we quit the program and we reopen the program and save the file, the new file will not replace the previous file we created. If we quit the program and we reopen the program and save the file, the new file will not replace the previous file we created.

Testing Strategy

To test our code, we mainly used “a4.db” as our database. We first had a discussion on the result of each question. We came up with results that we believed they were correct. Then we ran our code to see if the results are matched or not. The results generated from our code, thankfully, were matched with what we had discussed. To double check, we modified some of our database to see if the results are still satisfied. The outputs of the functions did change, and they were within our expectation. For example, in order to test Task 2, we changed number of population(canadian_citizen, non_canadian_citizen, and no_response) in the database, and the new population affect the result neighbourhoods. Similarly for Task 5, we modified the crime count with related data in database to check if the map shows the correct result neighbourhoods. We did similar tests for all of the four tasks.

Work Distribution

The interface we used was similar to the one from last assignment, with only few modifications. Therefore we did not put much time on it. For the coding part, we tried to make each of us have the same amount of work, so Xiaoyu Liu did Question 1 and 2 as well as the file saving part.Ye Wang did Question 3 and 4. We finished the report together via Google Docs. Both of us spent about a week working on our own work and the report. We started the project on Thursday March 21 and finished it on Saturday March 30. To keep the project on track, we met each other almost every day in the past week.