

Goal of the assignment:

The goal of this assignment is to further practice the use of SQL in a host programming language. This time, in addition to what you practiced in the context of Assignment #3, you will be using open data and a library to visualize the results of your query on a map.

Running the program in terminal:

```
python3 a4.py
```

```
***We access the database in main function
```

```
path="./a4.db"
```

Summary of our code:

We create a python program with multiple functions to handle all the tasks.

1. We have def connect() which connects sqlite with our code.
2. We have a function for each task :TASK 1,TASK 2,TASK 3 and TASK 4
3. Lastly our main function which runs our whole program.

We used the lab slides and sample codes provided to guide us through this assignment. We import all modules that we need for this assignment such as:

sqlite3,matplotlib.pyplot,pandas,folium.Then we write the connect(path) function which connects our code with sqlite3 and accesses a4.db. We use the main function() to execute all four tasks. For the main function, while the user keeps inputting enter, the program will not exit .According To user input,the task is executed. After a task is done, the user can press enter to perform another task. For example if the user enter "1", task 1 will be executed.Then the user may pressenter to perform another task. If there's an invalid input then print "Please enter a correct command" and asks for input again.

TASK 1:

Given a range of years and crime type, show (in a bar plot) the month-wise total count of the given crime type.

- We take input for startyear, end year and crime type.
- Then we use cursor.execute to access the query
- We add all the incidents_count from the year range and crime given by the user
- Then we create a DataFrame for the data
- Then we plot the data and save the output

TASK 2:

Given an integer N, show (in a map) the N-most populous and N-least populous neighborhoods with their population count.

- We take the input of number of locations
- Then we create a query for the top N- most populous neighborhoods
- Then we create a query for the N-least populous neighborhoods
- We use folium.Map to create the base map of Edmonton
- We use a for loop to access both the top and least populous areas

- We also deal with ties
- Then we save the map in a file

TASK 3:

Given a range of years, a crime type and an integer N, show (in a map) the Top-N neighborhoods and their crime count where the given crime type occurred most within the given range.

- Ask the user for a year range, type of crime, and number of locations
- Create a query to find the top N locations where the specific crime occurred the most
- We use folum.Map to create the base map of Edmonton
- Then we use a for loop to plot the top N points on the map including their neighborhood names and how many time the specific crime occurred.
- We also deal with ties
- Save the updated map

TASK 4:

Given a range of years and an integer N, show (in a map) the Top-N neighborhoods with the highest crimes to population ratio within the provided range. Also, show the most frequent crime type in each of these neighborhoods.

- Ask the user for startyear, end year and number of locations
- Use SQL to find the neighborhood name
- Add markers on the map to show the highest crimes to population ratio
- Then we deal with ties: for example if 2 areas have the same ratio, we show them both according to N

Main Function:

- Where we accesses the database
- We create a boolean value to keep track of Exit
- We create a while loop to get the user input and according to the user input, we run the task
- We update the count in our main
- Count is a variable for saving the files
- Everytime we access a function count adds one
- After the task is done, the user presses enter and has a choice of running more tasks.
- We end the program when the user presses E

Our Testing Policy:

We tested our assignment by sorting the data in the database and making sure the output we are getting is the same as the one in the database. Our testing process was lengthy however it was really helpful as it helped us figure out our errors

Our group work:

We worked together on brainstorming the ideas and all the tasks. We created a gitHub repository to upload our code. We met 3 times to stay on top of our assignment. We also met during lab times to figure unresolved issues and ask TAs for help.