CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Overview

The goal of this project is to write a console-based Python program that inputs commands from the user and outputs data from the Chicago traffic camera database. SQL should be used to retrieve and compute most of the information, while Python is used to display the results and, if the user chooses, to plot figures as well.

The Chicago Traffic Camera Database

The database provides information on all traffic cameras in Chicago.

As some may be familiar with, the city of Chicago has set up cameras all over the city to detect potential violations of red lights. The Red Light Camera Enforcement Program, created in 2003, is designed to encourage drivers to obey red light signals and thus reduce crashes at intersections. Read more about this program here: https://www.chicago.gov/city/en/depts/cdot/supp_info/red-light_cameraenforcement.html.

The city has also established cameras to detect instances of drivers exceeding the posted speed limit in areas that surround schools and parks (designated as Safety Zones). The first speed cameras in Chicago were activated in August 2013, and since then, have been a preventative measure for collisions of cars with pedestrians, children in particular. Read more about this program here: https://www.chicago.gov/city/en/depts/cdot/supp_info/children_s_safetyzoneporgramautomaticspeedenforcement.html.

This database contains information on 1) the cameras (both red light cameras and speed cameras) that are set up throughout the city, and 2) the number of daily violations for each camera. The data reflects information collected from July 1, 2014 up until and including November 28, 2024. The source of the data may be found online: https://data.cityofchicago.org/Transportation/Red-Light-Camera-Violations/spqx-js37/about_data and https://data.cityofchicago.org/Transportation/Speed-Camera-Violations/hhkd-xvj4/about_data.

The file for the database is provided for you (it is called ***chicago-traffic-cameras.db***). There are 5 tables in the database: **Intersections**, **RedCameras**, **SpeedCameras**, **RedViolations**, and **SpeedViolations**.

(The database uses *foreign keys*. A foreign key is a primary key stored in another table, typically used to join those tables. You may think of a foreign key as a pointer to the table where it is a primary key.

Example: **Intersection_ID** is the primary key of the **Intersections** table, and a foreign key in the **RedCameras** and **SpeedCameras** tables. This allows the **RedCameras** and **SpeedCameras** tables to point to the name of the intersection in case it is needed.)

The **Intersections** table gives all the intersections at which a camera is located. There could be a red light camera or a speed camera at one intersection – or even both types of cameras! An intersection may have more than one camera (in fact, this is usually the case).

- **Intersection_ID:** primary key, integer that uniquely identifies each intersection
- **Intersection:** string representing the two streets of the intersection, e.g. “WESTERN AVE AND CERMAK RD”

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

The **RedCameras** table provides information on all the red light cameras in Chicago.

- **Camera_ID:** primary key, integer that uniquely identifies each red light camera
- **Intersection_ID:** foreign key, integer corresponding to the **Intersections** table to indicate the intersection at which this camera is located
- **Address:** string representing the address of the physical camera, i.e. where it is located on the intersection
- **Latitude:** a real number representing the latitude of the physical location of the camera based on the address
- **Longitude:** a real number representing the longitude of the physical location of the camera based on the address

The **SpeedCameras** table provides information on all the speed cameras in Chicago.

- **Camera_ID:** primary key, integer that uniquely identifies each speed camera
- **Intersection_ID:** foreign key, integer corresponding to the **Intersections** table to indicate the intersection at which this camera is located. A speed camera may not be located directly at an intersection, so in some cases, this may represent the closest intersection of major streets.
- **Address:** string representing the address of the physical camera, i.e. where it is located on the intersection
- **Latitude:** a real number representing the latitude of the physical location of the camera based on the address
- **Longitude:** a real number representing the longitude of the physical location of the camera based on the address

The **RedViolations** table denotes the number of violations that occurred each day for each red light camera.

- **Camera_ID:** foreign key, integer that uniquely identifies each camera, connected to the **Camera_ID** of the **RedCameras** table
- **Violation_Date:** string representing the date corresponding to the data, in the format “YYYY-MM-DD”
- **Num_Violations:** integer representing the number of violations recorded on the given camera on the given day
- The pair (Camera_ID, Violation_Date) forms a composite primary key

The **SpeedViolations** table denotes the number of violations that occurred each day for each speed camera.

- **Camera_ID:** foreign key, integer that uniquely identifies each camera, connected to the **Camera_ID** of the **SpeedCameras** table
- **Violation_Date:** string representing the date corresponding to the data, in the format “YYYY-MM-DD”
- **Num_Violations:** integer representing the number of violations recorded on the given camera on the given day
- The pair (Camera_ID, Violation_Date) forms a composite primary key

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Starting The Program

The program starts by outputting some basic statistics retrieved from the database:

```
Project 1: Chicago Traffic Camera Analysis
CS 341, Spring 2025

This application allows you to analyze various
aspects of the Chicago traffic camera database.

General Statistics:
  Number of Red Light Cameras: 365
  Number of Speed Cameras: 184
  Number of Red Light Camera Violation Entries: 998,470
  Number of Speed Camera Violation Entries: 408,256
  Range of Dates in the Database: 2014-07-01 - 2024-11-28
  Total Number of Red Light Camera Violations: 6,216,244
  Total Number of Speed Camera Violations: 16,767,367
```

You will need to match the output exactly. Note that these values may change based on the database tested against. When you submit to Gradescope for testing, we reserve the right to change the contents of the database to confirm you are writing a general-purpose program (same schema, but the underlying data may be different).

After the statistics, the program allows the user to repeatedly enter string-based commands “1” - “9” or “x” to exit. All other inputs should yield an error message.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 10
Error, unknown command, try again...

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
```


```
Your choice --> hello
Error, unknown command, try again...

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> X
Error, unknown command, try again...

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> x
Exiting program.
```

The program should continue running until the user inputs “x” to exit. The user can input commands in any order, and may repeat commands as often as they want.

Details on each command may be found in the sections that follow.

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 1

Find all the intersections that match the user input. The user will be asked to input an intersection. SQL wildcards _ and % are allowed. Output the ID and name of each intersection, in order by the names, from A to Z. If no intersections are found, print a message indicating that.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): LaSalle
No intersections matching that name were found.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): western
No intersections matching that name were found.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): %Western%
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
5073 : WESTERN AVE AND 103RD ST
5102 : WESTERN AVE AND 107TH PL
5193 : WESTERN AVE AND 35TH ST
5197 : WESTERN AVE AND 47TH ST
5046 : WESTERN AVE AND 51ST ST
5032 : WESTERN AVE AND 55TH ST
5212 : WESTERN AVE AND 63RD ST
5232 : WESTERN AVE AND 71ST ST
5031 : WESTERN AVE AND 79TH ST
5014 : WESTERN AVE AND ADDISON ST
5174 : WESTERN AVE AND ARMITAGE AVE
5052 : WESTERN AVE AND CERMAK RD
5182 : WESTERN AVE AND CHICAGO AVE
5025 : WESTERN AVE AND DEVON AVE
5138 : WESTERN AVE AND DIVERSEY AVE
5061 : WESTERN AVE AND DIVISION ST
5108 : WESTERN AVE AND FOSTER AVE
5155 : WESTERN AVE AND FULLERTON AVE
5100 : WESTERN AVE AND IRVING PARK RD
5118 : WESTERN AVE AND LAWRENCE AVE
5181 : WESTERN AVE AND MADISON ST
5238 : WESTERN AVE AND MARQUETTE RD
5026 : WESTERN AVE AND MONTROSE AVE
5173 : WESTERN AVE AND NORTH AVE
5105 : WESTERN AVE AND PETERSON AVE
5125 : WESTERN AVE AND PRATT BLVD
5120 : WESTERN AVE AND TOUHY AVE
5189 : WESTERN AVE AND VAN BUREN ST
5081 : WESTERN BLVD AND 51ST ST
5002 : WESTERN BLVD AND PERSHING RD

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): %lakeshore%
No intersections matching that name were found.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): %lake%
5034 : ASHLAND AVE AND LAKE ST
5162 : CENTRAL AVE AND LAKE ST
5176 : HAMLIN AVE AND LAKE ST
5135 : LAKE SHORE DR AND BELMONT AVE
5170 : SACRAMENTO BLVD AND LAKE ST
5249 : UPPER WACKER DR AND LAKE ST

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 1

Enter the name of the intersection to find (wildcards _ and % allowed): %55TH___
5007 : COTTAGE GROVE AVE AND 55TH ST
5078 : KEDZIE AVE AND 55TH ST
5191 : NARRAGANSETT AVE AND ARCHER AVE/55TH ST
5029 : PULASKI RD AND 55TH ST
5032 : WESTERN AVE AND 55TH ST
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 2

Given the name of an intersection, find all the cameras located at that intersection. Both red light and speed cameras should be shown, each in ascending order by camera ID. Also display the address of each camera. The name of the intersection entered by the user should be an exact match. If no red light cameras and/or speed cameras are located at that intersection, print a corresponding message to indicate that.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 2

Enter the name of the intersection (no wildcards allowed): PULASKI RD AND NORTH AVE

Red Light Cameras:
   1992 : 1600 N PULASKI AVENUE
   1994 : 4000 W NORTH AVENUE
   8304 : 4000 W NORTH AVENUE
   8313 : 1600 N PULASKI AVENUE

Speed Cameras:
   9157 : 1754 N PULASKI RD
   9159 : 4053 W NORTH AVE
   9161 : 4042 W NORTH AVE

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 2

Enter the name of the intersection (no wildcards allowed): ARCHER AND ASHLAND

No red light cameras found at that intersection.

No speed cameras found at that intersection.
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 2

Enter the name of the intersection (no wildcards allowed): ASHLAND AVE AND ARCHER AVE

Red Light Cameras:
   2301 : 3100 S ASHLAND AVENUE
   2302 : 3100 S ARCHER AVENUE

Speed Cameras:
   9144 : 3200 S ARCHER AVE

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 2

Enter the name of the intersection (no wildcards allowed): DAMEN AVE AND ELSTON AVE

Red Light Cameras:
   3072 : 2426 N DAMEN AVE

No speed cameras found at that intersection.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
```

```
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 2

Enter the name of the intersection (no wildcards allowed): KEDZIE AVE AND AUGUSTA BLVD

No red light cameras found at that intersection.

Speed Cameras:
   9011 : 3100 W AUGUSTA
```

Command 3

The user enters a specific date, and the program outputs the number of red light violations for that date across all cameras and the number of speed violations for that date across all cameras. Also show the percentages of each, taken out of the total number of violations.

The date entered by the user should be an exact match, including formatting. If the date does not exist in the database (it was formatted incorrectly, it is out of range of the dates included in the data, etc), print a message to indicate that.

The number of red light violations and the number of speed violations **must** be computed using SQL. You may use Python to calculate percentages and to calculate the total number of violations.

Here is one way that you can format the output in Python:

```
print("Number of Red Light Violations:", f"{num:,}", f"({percent:.3f}%)")
```

In the print statement, the “f” stands for formatted output. The “:,” after the count variable means that the value should be formatted with “,” separators. The “:.3f” after the percentage variable means that it should output the value with 3 digits following the decimal point.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 3

Enter the date that you would like to look at (format should be YYYY-MM-DD): 07/01/2014
No violations on record for that date.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 3

Enter the date that you would like to look at (format should be YYYY-MM-DD): 2014-07-01
Number of Red Light Violations: 1,673 (25.238%)
Number of Speed Violations: 4,956 (74.762%)
Total Number of Violations: 6,629

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 3

Enter the date that you would like to look at (format should be YYYY-MM-DD): 2025-01-01
No violations on record for that date.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 3

Enter the date that you would like to look at (format should be YYYY-MM-DD): 2018-12-31
Number of Red Light Violations: 1,097 (43.189%)
Number of Speed Violations: 1,443 (56.811%)
Total Number of Violations: 2,540
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 4

Output the number of red light cameras at each intersection, along with the percentages (taken out of the total number of red light cameras in Chicago). Then output the number of speed cameras at each intersection, along with the percentages (taken out of the total number of speed cameras in Chicago). Each list should be ordered from the most number of cameras to the least number of cameras, and should include the ID and name of each intersection as well.

The number of red light cameras at each intersection, the number of speed cameras at each intersection, the total number of red light cameras in Chicago, and the total number of speed cameras in Chicago **must all** be computed using SQL. You may use Python to calculate percentages.

Partial output is shown below, as it is quite long.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 4

Number of Red Light Cameras at Each Intersection
  STONY ISLAND AVE AND 79TH ST (5221) : 4 (1.096%)
  PULASKI RD AND NORTH AVE (5075) : 4 (1.096%)
  KOSTNER AVE AND NORTH AVE (5185) : 3 (0.822%)
  CALIFORNIA AVE AND DIVERSEY AVE (5143) : 3 (0.822%)
  DAMEN AVE AND DIVERSEY PKWY (5136) : 3 (0.822%)
  . . .
  WESTERN BLVD AND PERSHING RD (5002) : 2 (0.548%)
  PULASKI RD AND FOSTER AVE (5001) : 2 (0.548%)
  DAMEN AVE AND ELSTON AVE (5250) : 1 (0.274%)
  COTTAGE GROVE AVE AND 71ST ST (5228) : 1 (0.274%)
  LARAMIE AVE AND FULLERTON AVE (5137) : 1 (0.274%)

Number of Speed Cameras at Each Intersection
  PULASKI RD AND 55TH ST (5029) : 6 (3.261%)
  PULASKI RD AND 79TH ST (5033) : 5 (2.717%)
  WESTERN AVE AND CERMAK RD (5052) : 4 (2.174%)
  NARRAGANSETT AVE AND BELMONT AVE (5050) : 4 (2.174%)
  WESTERN AVE AND 79TH ST (5031) : 4 (2.174%)
  . . .
  OGDEN AVE AND TAYLOR ST (5010) : 1 (0.543%)
  KEDZIE AVE AND AUGUSTA BLVD (5006) : 1 (0.543%)
  HUMBOLDT DR AND NORTH AVE (5005) : 1 (0.543%)
  CENTRAL PARK AVE AND MADISON ST (5004) : 1 (0.543%)
  CENTRAL PARK AVE AND JACKSON BLVD (5003) : 1 (0.543%)
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 5

Allow the user to enter a year. Then output the number of red light violations recorded at each intersection for that year, along with the percentage, which is taken out of the total number of red light violations for that year in all of Chicago. Display the ID and name of each intersection as well. At the end, print the total number of red light violations for that year in all of Chicago. Repeat for the speed camera violations at each intersection. Each list should be ordered from the greatest number of violations to the least number of violations. If there are no records in that year, display a message to indicate that.

The number of red light cameras at each intersection, the number of speed cameras at each intersection, the total number of red light cameras in Chicago, and the total number of speed cameras in Chicago **must all** be computed using SQL. You may use Python to calculate percentages.

Partial output is shown below, as it is quite long.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 5

Enter the year that you would like to analyze: 2001

Number of Red Light Violations at Each Intersection for 2001
No red light violations on record for that year.

Number of Speed Violations at Each Intersection for 2001
No speed violations on record for that year.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 5

Enter the year that you would like to analyze: 2018
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Number of Red Light Violations at Each Intersection for 2018
  CICERO AVE AND I55 (5203) : 24,561 (4.363%)
  UPPER WACKER DR AND LAKE ST (5249) : 20,762 (3.688%)
  LAKE SHORE DR AND BELMONT AVE (5135) : 18,835 (3.346%)
  LAFAYETTE AVE AND 87TH ST (5223) : 16,475 (2.927%)
  WESTERN AVE AND VAN BUREN ST (5189) : 15,019 (2.668%)
  . . .
  WESTERN AVE AND TOUHY AVE (5120) : 823 (0.146%)
  PULASKI AVE AND LAWRENCE AVE (5117) : 681 (0.121%)
  CENTRAL AVE AND DIVERSEY AVE (5153) : 547 (0.097%)
  LARAMIE AVE AND FULLERTON AVE (5137) : 431 (0.077%)
  DAMEN AVE AND FULLERTON AVE (5154) : 67 (0.012%)
Total Red Light Violations in 2018 : 562,917

Number of Speed Violations at Each Intersection for 2018
  CICERO AVE AND LAWRENCE AVE (5071) : 64,183 (6.675%)
  COTTAGE GROVE AVE AND 55TH ST (5007) : 48,064 (4.999%)
  PULASKI RD AND FOSTER AVE (5001) : 46,964 (4.884%)
  HALSTED ST AND 127TH ST (5022) : 44,944 (4.674%)
  CALIFORNIA AVE AND IRVING PARK RD (5037) : 40,478 (4.210%)
  . . .
  BROADWAY AND MONTROSE AVE (5045) : 810 (0.084%)
  KEDZIE AVE AND 55TH ST (5078) : 771 (0.080%)
  LINCOLN AVE AND MONTROSE AVE (5027) : 675 (0.070%)
  PULASKI RD AND DIVERSEY AVE (5083) : 326 (0.034%)
  CALIFORNIA AVE AND MONTROSE AVE (5036) : 275 (0.029%)
Total Speed Violations in 2018 : 961,543

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 5

Enter the year that you would like to analyze: 2014

Number of Red Light Violations at Each Intersection for 2014
  LAKE SHORE DR AND BELMONT AVE (5135) : 8,135 (3.047%)
  CICERO AVE AND I55 (5203) : 6,825 (2.556%)
  WESTERN AVE AND VAN BUREN ST (5189) : 6,592 (2.469%)
  CALIFORNIA AVE AND DIVERSEY AVE (5143) : 5,768 (2.160%)
  SHERIDAN RD AND HOLLYWOOD AVE (5128) : 4,496 (1.684%)
  . . .
  CENTRAL AVE AND FULLERTON AVE (5019) : 387 (0.145%)
```

```
  WESTERN AVE AND ARMITAGE AVE (5174) : 359 (0.134%)
  PULASKI RD AND MONTROSE AVE (5124) : 338 (0.127%)
  CICERO AVE AND ARMITAGE AVE (5184) : 214 (0.080%)
  COTTAGE GROVE AVE AND 71ST ST (5228) : 187 (0.070%)
Total Red Light Violations in 2014 : 267,007

Number of Speed Violations at Each Intersection for 2014
  RACINE AVE AND GARFIELD BLVD (5056) : 37,030 (5.616%)
  COTTAGE GROVE AVE AND 55TH ST (5007) : 36,654 (5.558%)
  PULASKI RD AND FOSTER AVE (5001) : 28,962 (4.392%)
  CALIFORNIA AVE AND IRVING PARK RD (5037) : 27,249 (4.132%)
  INDIANAPOLIS AVE AND 103RD ST (5057) : 27,100 (4.110%)
  . . .
  CENTRAL AVE AND HARRISON ST (5062) : 613 (0.093%)
  ASHLAND AVE AND LINCOLN AVE (5080) : 585 (0.089%)
  BROADWAY AND MONTROSE AVE (5045) : 409 (0.062%)
  LINCOLN AVE AND MONTROSE AVE (5027) : 304 (0.046%)
  WESTERN AVE AND 103RD ST (5073) : 156 (0.024%)
Total Speed Violations in 2014 : 659,424
```

Command 6

Given a camera ID, output the number of violations recorded by that camera for each year, in ascending order by year. Show an error message if the camera ID does not exist.

[HINT: When the user enters a camera ID, you may need to check more than one table to see if the camera ID exists in the database.]

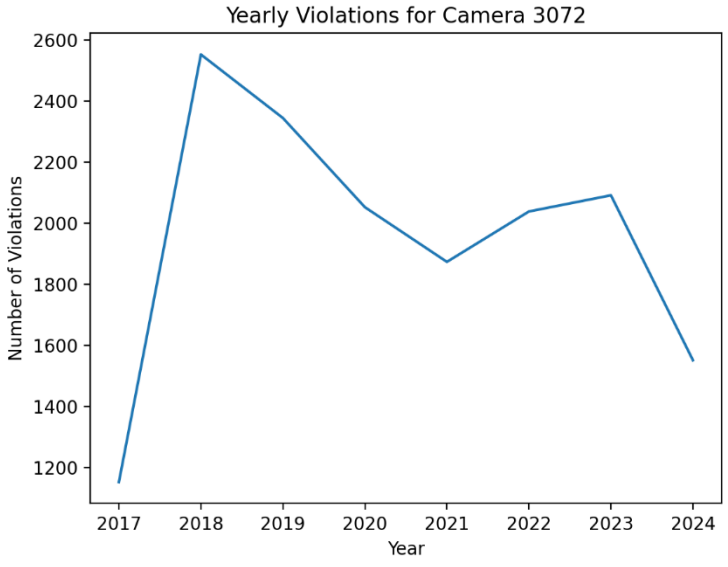
After the output, the user is given the option to plot the data. Make sure the axis labels and title of the figure are set appropriately. If the user responds with any input other than “y”, do not plot.

A couple of plots are shown along with the output below. They are pictured off to the side, so as to not interfere with the output itself.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 6

Enter a camera ID: 999999
No cameras matching that ID were found in the database.

Select a menu option:
  1. Find an intersection by name
```

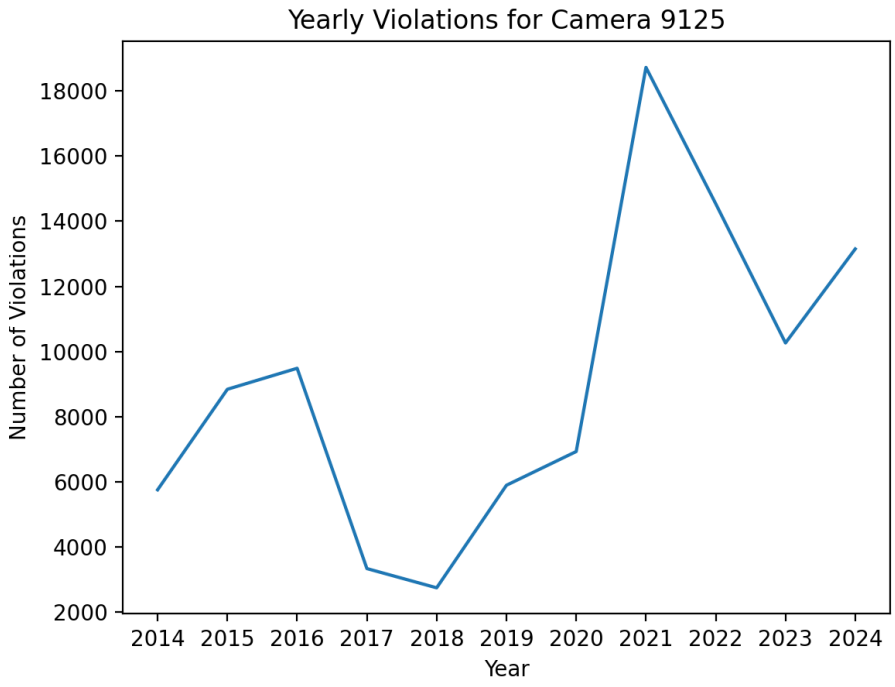
CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 6

Enter a camera ID: 1994
Yearly Violations for Camera 1994
2014 : 326
2015 : 819
2016 : 1,092
2017 : 656
2018 : 865
2019 : 552
2020 : 679
2021 : 1,030
2022 : 831
2023 : 1,054
2024 : 1,087

Plot? (y/n) x

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 6

Enter a camera ID: 3072
Yearly Violations for Camera 3072
2017 : 1,152
2018 : 2,552
2019 : 2,344
2020 : 2,052
2021 : 1,873
2022 : 2,038
2023 : 2,091
2024 : 1,551

Plot? (y/n) y
```



CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 6

Enter a camera ID: 9125
Yearly Violations for Camera 9125
2014 : 5,759
2015 : 8,843
2016 : 9,487
2017 : 3,342
2018 : 2,752
2019 : 5,898
2020 : 6,933
2021 : 18,718
2022 : 14,535
2023 : 10,264
2024 : 13,149

Plot? (y/n) y
```



CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 7

Given a camera ID and a year, output the number of violations recorded by that camera for each month in the specified year, in ascending order by month. Show an error message if the camera ID does not exist.

[HINT: When the user enters a camera ID, you may need to check more than one table to see if the camera ID exists in the database.]

After the output, the user is given the option to plot the data. Make sure the axis labels and title of the figure are set appropriately. If the user responds with any input other than “y”, do not plot.

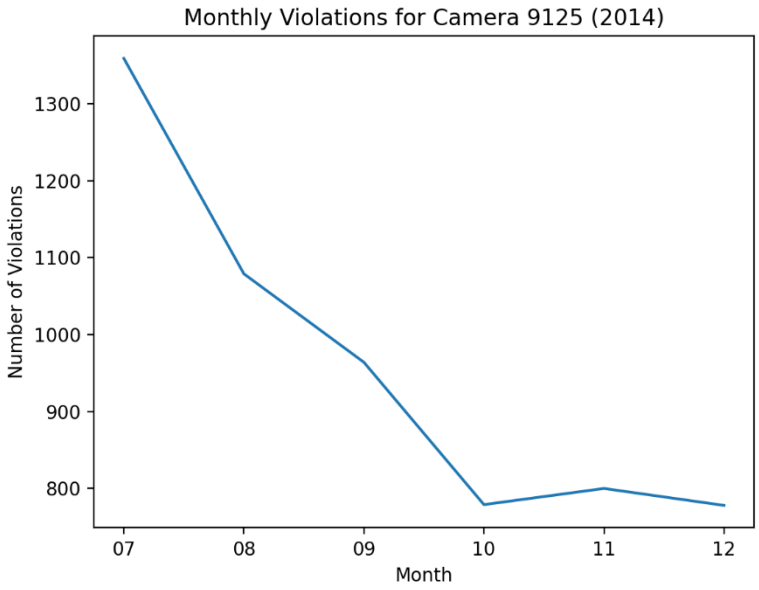
Note that if the user enters a year for which there is no data, no error message is necessary. The output and plot will be empty, which is sufficient.

A couple of plots are shown along with the output below. They are pictured off to the side, so as to not interfere with the output itself.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 7

Enter a camera ID: 123456
No cameras matching that ID were found in the database.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 7

Enter a camera ID: 9125
Enter a year: 2001
Monthly Violations for Camera 9125 in 2001

Plot? (y/n) n
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

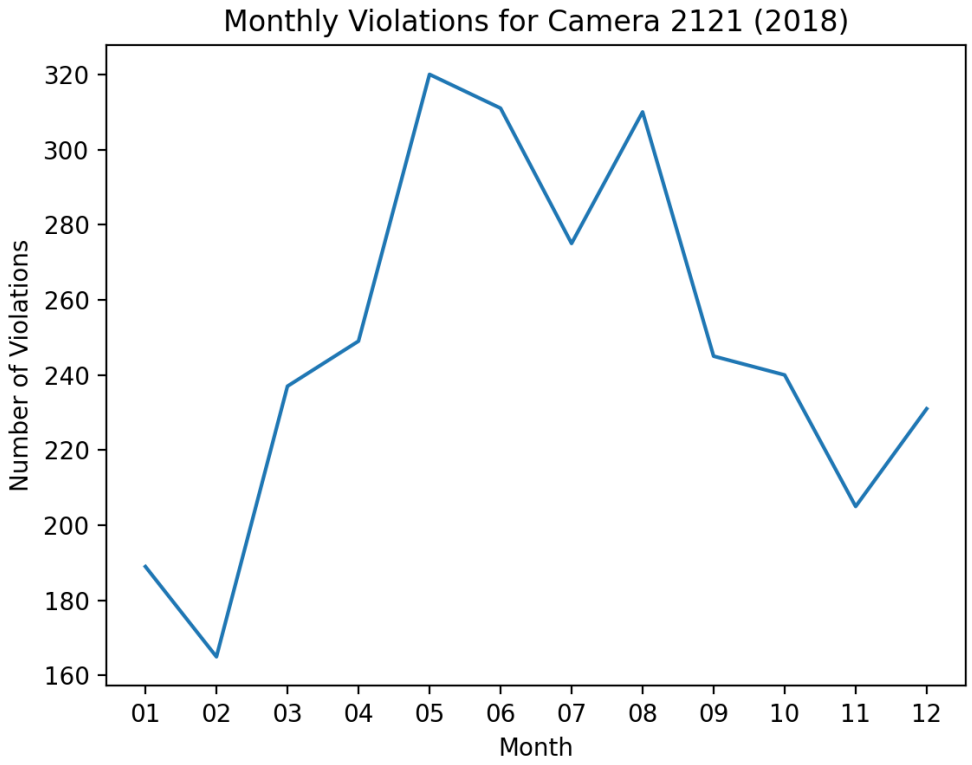
```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 7

Enter a camera ID: 9125
Enter a year: 2014
Monthly Violations for Camera 9125 in 2014
07/2014 : 1,359
08/2014 : 1,079
09/2014 : 964
10/2014 : 779
11/2014 : 800
12/2014 : 778

Plot? (y/n) y
```




```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 7

Enter a camera ID: 2121
Enter a year: 2018
Monthly Violations for Camera 2121 in 2018
01/2018 : 189
02/2018 : 165
03/2018 : 237
04/2018 : 249
05/2018 : 320
06/2018 : 311
07/2018 : 275
08/2018 : 310
09/2018 : 245
10/2018 : 240
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
11/2018 : 205
12/2018 : 231

Plot? (y/n) y
```



CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Command 8

Given a year, output the number of red light violations across all red light cameras, and the number of speed violations across all speed cameras, for each day in that year. The data should be displayed in ascending order by the date.

Since the full output would be quite long, you should only output the first 5 days and last 5 days of data for each station. *Notice that the first 5 days and last 5 days here is in reference to the dates that are in the database for that year.*

Also give the user the option to see a plot of the results. If the user responds with "y" your program should plot as shown below (with appropriate title, legend, and axis labels). If the user responds with any other input, do not plot.

In the plot, notice the following:

- There are two distinct colors to indicate which line on the graph corresponds to which type of violation is being shown. Red is used for red light camera violations, and orange is used for speed camera violations.
- The days on the x-axis correspond one-to-one with the dates in the year, *not* the dates in the database. Some years in the database do not have records for all dates in that year (e.g. for 2014, records start on July 1, not January 1). For such dates, the corresponding y-value on the plot should be zero. Do not hardcode for this – you have the foundational knowledge you need to be able to solve this problem! The **`datetime`** module in Python may be useful.

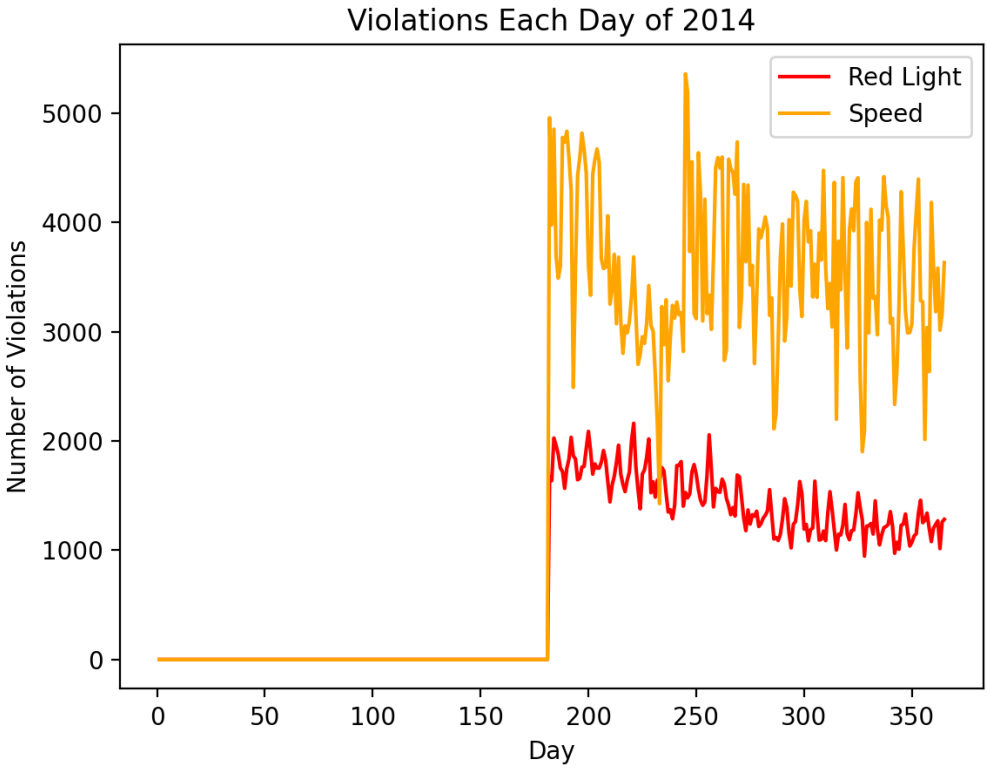
Note that if the user enters a year for which there is no data, no error message is necessary. The output and plot will be empty, which is sufficient.

A couple of plots are shown along with the output below. They are pictured off to the side, so as to not interfere with the output itself.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 8

Enter a year: 1975
Red Light Violations:
Speed Violations:

Plot? (y/n) n

Select a menu option:
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

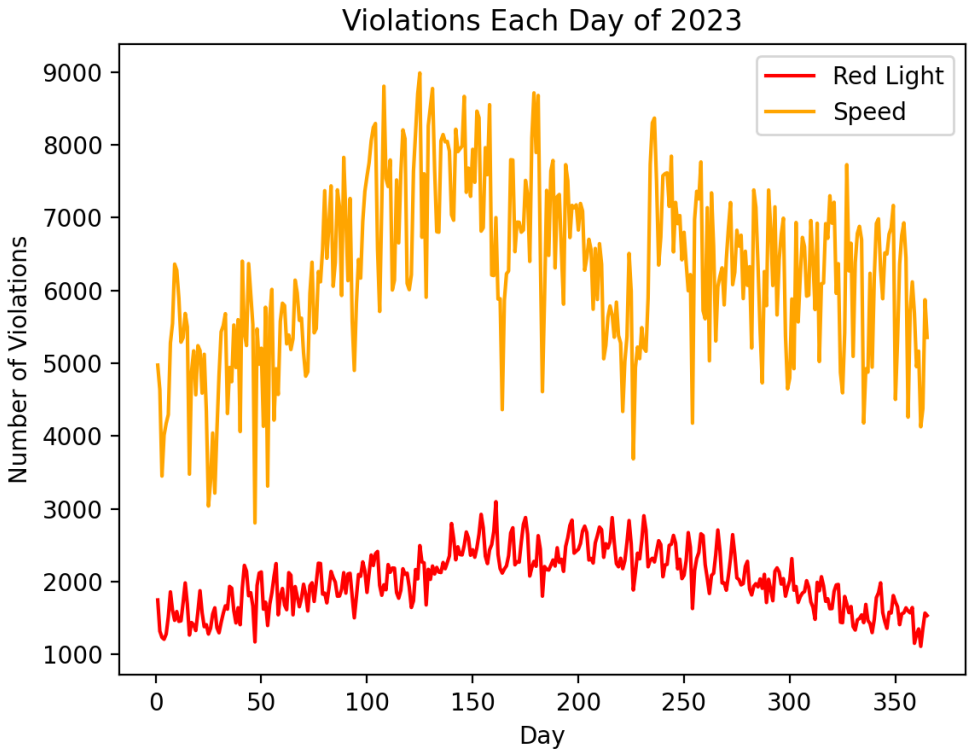
```
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 8

Enter a year: 2014
Red Light Violations:
2014-07-01 1673
2014-07-02 1637
2014-07-03 2026
2014-07-04 1958
2014-07-05 1880
2014-12-27 1236
2014-12-28 1269
2014-12-29 1015
2014-12-30 1258
2014-12-31 1282
Speed Violations:
2014-07-01 4956
2014-07-02 3977
2014-07-03 4852
2014-07-04 3695
2014-07-05 3489
2014-12-27 3182
2014-12-28 3581
2014-12-29 3013
2014-12-30 3162
2014-12-31 3632
```




```
Plot? (y/n) y

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 8

Enter a year: 2023
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Red Light Violations:
2023-01-01 1749
2023-01-02 1320
2023-01-03 1235
2023-01-04 1209
2023-01-05 1281
2023-12-27 1351
2023-12-28 1112
2023-12-29 1359
2023-12-30 1569
2023-12-31 1535
Speed Violations:
2023-01-01 4975
2023-01-02 4636
2023-01-03 3451
2023-01-04 4013
2023-01-05 4175
2023-12-27 5167
2023-12-28 4128
2023-12-29 4380
2023-12-30 5873
2023-12-31 5355
```




```
Plot? (y/n) y

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 8

Enter a year: 2024
Red Light Violations:
2024-01-01 1514
2024-01-02 1223
2024-01-03 1188
2024-01-04 1357
2024-01-05 1379
2024-11-24 862
2024-11-25 1003
2024-11-26 1467
2024-11-27 1447
2024-11-28 1308
Speed Violations:
2024-01-01 5550
```

```
2024-01-02 4797
2024-01-03 4091
2024-01-04 4888
2024-01-05 4809
2024-11-24 5629
2024-11-25 3862
2024-11-26 4601
2024-11-27 4599
2024-11-28 6536

Plot? (y/n) Y
```

Command 9

Given the name of a street, find all red light cameras and all speed cameras that are physically located on that street. Display the ID, address, latitude and longitude of those cameras, in ascending order by the camera ID.

Give the user the option to plot these cameras on a map as shown below. Notice that the color of the dots are different: red dots indicate red light cameras and orange dots indicate speed cameras.

Note that the user only enters the street name. [HINT: You may need to add wildcards to find the cameras physically located on that street.]

If no cameras are located on the street entered by the user, display a message to indicate that.

If the user chooses to view the plot, the locations of the stations are seen overlaying a map of Chicagoland. The map is provided as an image (.png) file in the starter code. This turns out to be surprisingly easy to do in Python. First, make sure the “chicago.png” image file is in the same folder as your Python program and the database file. Then do the following:

```
#
# populate x and y lists with (x, y) coordinates
# note that longitude are the X values and
#      latitude are the Y values
#
x = []
y = []
.
.
.
image = plt.imread("chicago.png")
xydims = [-87.9277, -87.5569, 41.7012, 42.0868] # area covered by the map
plt.imshow(image, extent=xydims)
plt.title("……")

plt.plot(x, y)
#
# annotate each (x, y) coordinate with its station name:
#
for row in rows:
```

```
      plt.annotate(a_camera_id, (xposition, yposition))
plt.xlim([-87.9277, -87.5569])
plt.ylim([41.7012, 42.0868])
plt.show()
```

In case you're curious, here is how the image was created, and where the xydims / xlim / ylim values came from:

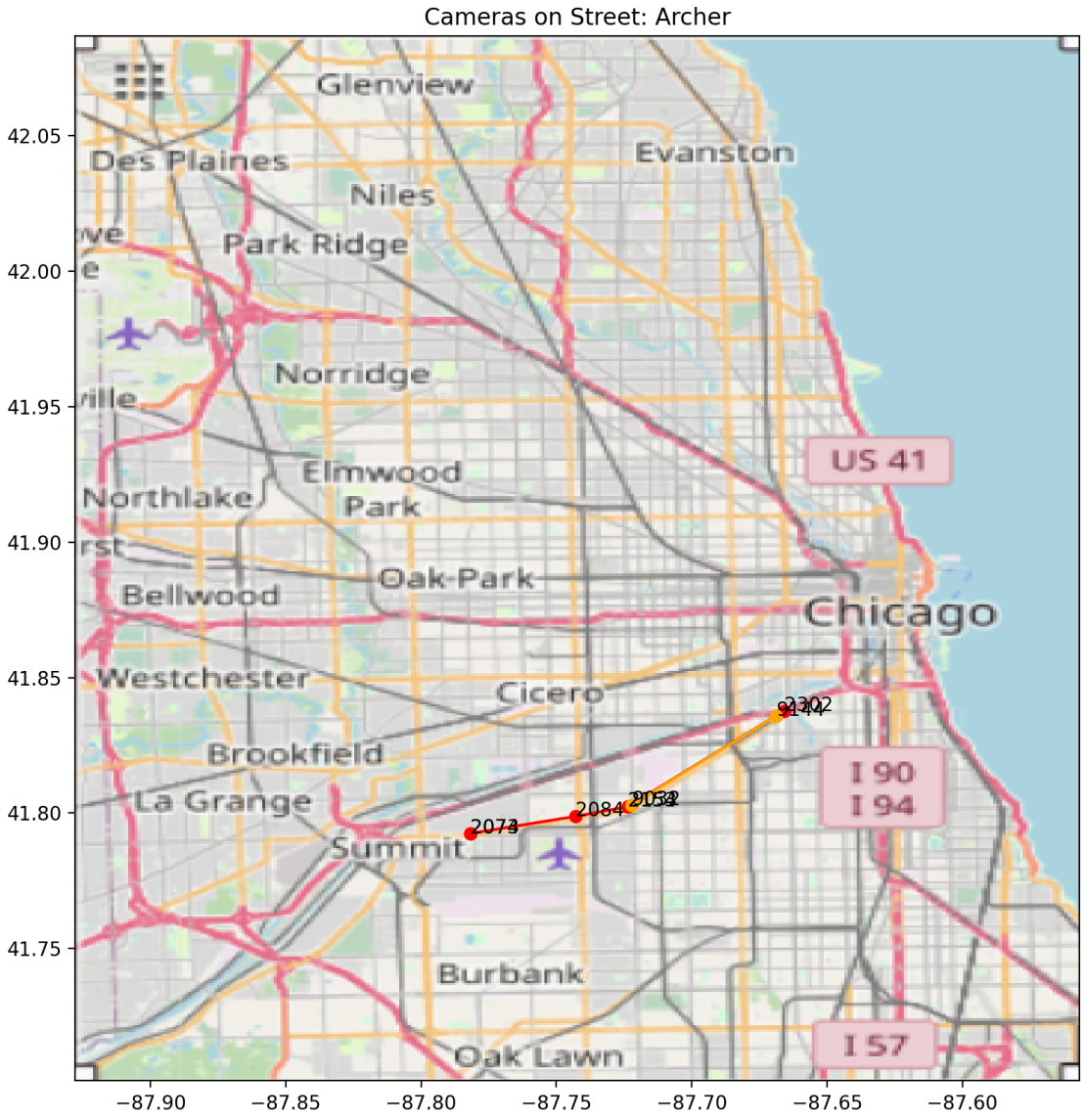
```
# Map grid from min and max values from the data:
#
# Longitude (x): -87.82831694 | -87.52984826
# Latitude (y): 41.66317406 | 42.01237133
#
# How to get map image?
# 1. https://www.openstreetmap.org
# 2. search for say "Chicago"
# 3. click on export in title bar
# 4. click on "manually select a different area"
# 5. resize the box
# 6. screenshot the box, save as .png file
# 7. record the 4 coordinates of the box
```

A couple of plots are shown along with the output below. They are shown inline, so you may refer to the output without plots to see the expected formatting.

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 9

Enter a street name: Kilpatrick
There are no cameras located on that street.

Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Your choice --> 9

Enter a street name: Archer

List of Cameras Located on Street: Archer
  Red Light Cameras:
     2073 : 6400 W ARCHER AVE (41.79245526, -87.78187685)
     2074 : 6400 W ARCHER AVE (41.79245526, -87.78187685)
     2084 : 5400 S ARCHER AVE (41.79875821, -87.74302054)
     2153 : 5000 S ARCHER AVE (41.80236597, -87.72357677)
     2154 : 5000 S ARCHER AVE (41.80236597, -87.72357677)
     2302 : 3100 S ARCHER AVENUE (41.83773268, -87.66589445)
  Speed Cameras:
     9032 : 4965 S ARCHER (41.80293879, -87.72218179)
     9144 : 3200 S ARCHER AVE (41.83595802, -87.66879027)

Plot? (y/n) y
```

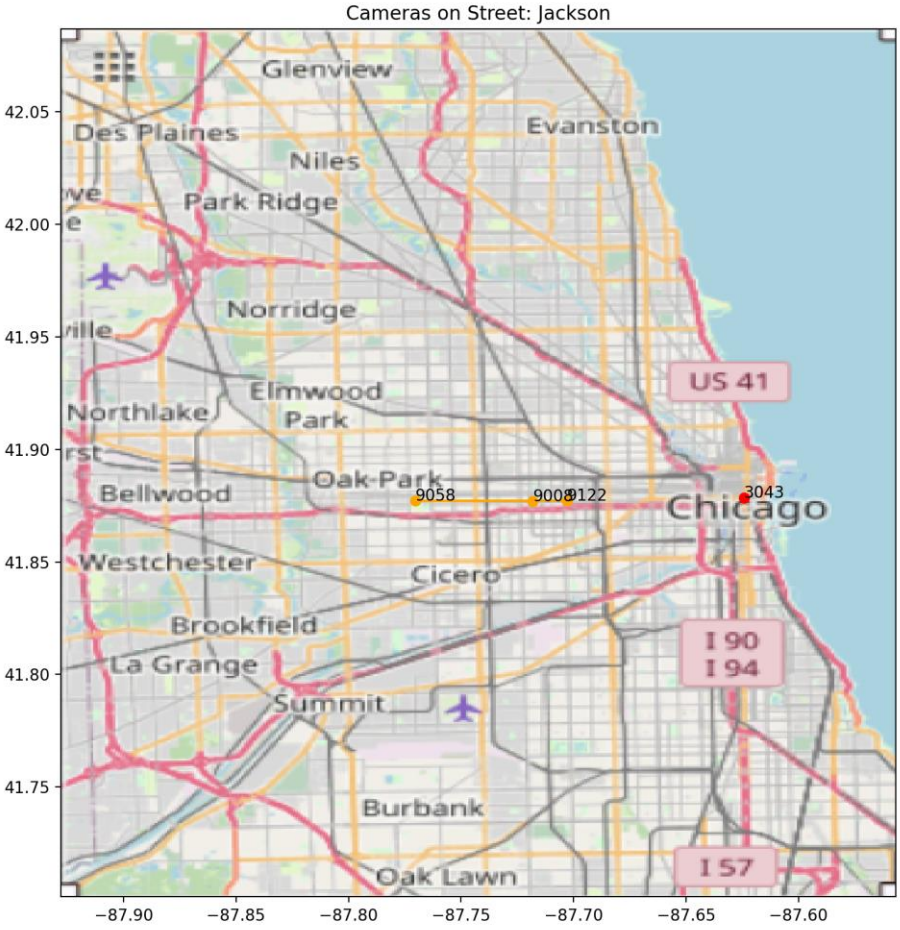


CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 9

Enter a street name: Pulaski

List of Cameras Located on Street: Pulaski
  Red Light Cameras:
     1102 : 5200 N PULASKI RD (41.97562074, -87.72838314)
     1172 : 4800 N PULASKI ROAD (41.9682626, -87.72817532)
     1262 : 6000 N PULASKI ROAD (41.99016092, -87.72879075)
     1292 : 4400 N PULASKI ROAD (41.96102372, -87.72797746)
     1382 : 3200 N PULASKI RD (41.93913815, -87.72733371)
     1481 : 2400 N PULASKI RD (41.92457257, -87.72691911)
     1542 : 2800 N PULASKI ROAD (41.93183103, -87.72712687)
     1572 : 4000 N PULASKI ROAD (41.95373705, -87.72776661)
     1801 : 1200 N PULASKI ROAD (41.9027139, -87.72645146)
     1802 : 1200 N PULASKI ROAD (41.9027139, -87.72645146)
     1982 : 800 N PULASKI ROAD (41.89545646, -87.72623054)
     1992 : 1600 N PULASKI AVENUE (41.90997501, -87.72648052)
     2242 : 5500 S PULASKI (41.79315506, -87.72330376)
     2311 : 1200 S PULASKI ROAD (41.86607934, -87.72530599)
     2332 : 2200 S PULASKI (41.85152144, -87.72490479)
     2342 : 6300 S PULASKI RD (41.77863647, -87.72290636)
     2492 : 7900 S PULASKI ROAD (41.749339, -87.72205209)
     8313 : 1600 N PULASKI AVENUE (41.90997501, -87.72648052)
  Speed Cameras:
     9004 : 5120 N PULASKI (41.97433265, -87.72834702)
     9030 : 4929 S PULASKI (41.80326489, -87.72330351)
     9031 : 5030 S PULASKI (41.80140191, -87.7235446)
     9059 : 5433 S PULASKI (41.79404505, -87.72303636)
     9060 : 5428 S PULASKI (41.79415458, -87.72333321)
     9071 : 7833 S PULASKI (41.75038453, -87.72179454)
     9072 : 7826 S PULASKI (41.7505462, -87.72209266)
     9109 : 732 N PULASKI RD (41.89450331, -87.72620452)
     9157 : 1754 N PULASKI RD (41.91337483, -87.7265886)
     9162 : 1117 S PULASKI RD (41.86740184, -87.72508369)
     9163 : 1110 S PULASKI RD (41.86760272, -87.72538274)
     9202 : 5433 S PULASKI (41.79404505, -87.72303636)
     9203 : 5428 S PULASKI (41.79415458, -87.72333321)

Plot? (y/n) n
```

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

```
Select a menu option:
  1. Find an intersection by name
  2. Find all cameras at an intersection
  3. Percentage of violations for a specific date
  4. Number of cameras at each intersection
  5. Number of violations at each intersection, given a year
  6. Number of violations by year, given a camera ID
  7. Number of violations by month, given a camera ID and year
  8. Compare the number of red light and speed violations, given a year
  9. Find cameras located on a street
or x to exit the program.
Your choice --> 9

Enter a street name: Jackson

List of Cameras Located on Street: Jackson
  Red Light Cameras:
     3043 : 100 E JACKSON BLVD (41.87839012, -87.62423731)
  Speed Cameras:
     9008 : 3655 W JACKSON (41.87707083, -87.71816832)
     9058 : 5816 W JACKSON (41.8771894, -87.7703695)
     9122 : 3047 W JACKSON BLVD (41.87724277, -87.70295269)

Plot? (y/n) y
```



CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Requirements

The main requirements are that you use Python3 and SQL, and the provided SQLite database. All data retrieval and computation should be performed using explicit, string-based SQL queries executed via the sqlite3 package; no tool-generated code is allowed (e.g. you cannot use SQLAlchemy). This implies that **SQL must be used to do the vast majority of computation, e.g. all searching and summing and sorting**. It is not valid to load all the data into objects and then write code to do the searching and sorting yourself. **Failure to use SQL for computation (unless specifically specified otherwise above) will result in a large point deduction.** For plotting, use the matplotlib package. (Some of the expected output contains percentages; computation of percentages in this case is best done using Python, so the use of SQL is not required.)

It is also expected that you use good programming practices, i.e. functions, comments, consistent spacing, etc. In a 3xx class this should be obvious and not require further explanation. But to be clear, if you submit a program with no functions and no comments, you will be significantly penalized --- in fact you can expect a score of 0, even if the program produces the correct results. How many functions? How many comments? You can decide. Make reasonable decisions and you will not be penalized.

Submission

Login to Gradescope.com and look for the assignment “Project 01”.

Submit just your main.py file under “Project 01”. **You have unlimited submissions. Keep in mind we grade your last submission unless you select an earlier submission for grading. If you do choose to activate an earlier submission, you must do so before the deadline.**

The score reported on Gradescope is only part of your final score (50%). Out of fairness, the autograder is the only means by which your program will be graded for correctness, i.e. functionality.

After the project is due, the TAs will manually review the programs for style (10%) and adherence to requirements (0-100%), and then manually run the programs to check the required plotting functionality (40%).

Suggestion: when you fail a test on Gradescope, we show your output, the correct output, and the difference between the two (as computed by Linux’s diff utility). Please study the output carefully to see where your output differs. If there are lots of differences, or you can’t see the difference, here’s a good tip:

1. Browse to https://www.diffchecker.com/
2. Copy your output as given by Gradescope and paste into the left window
3. Copy the correct output as given by Gradescope and paste into the right window
4. Click the “Find Difference” button

You will get a visual representation of the differences. Modify your program to produce the required output, and resubmit.

CS 341, Spring 2025
Project 1 – Chicago Traffic Camera Analysis
Due: Friday 2/7/2025 at 11:59pm

Late submissions for this project are allowed. You may turn in a project up to 3 days (72 hours) late, and will receive the following penalty on your total score:

Submission made **up to 24 hours late**: **10** point deduction

Submission made **24-48 hours late**: **20** point deduction

Submission made **48-72 hours late**: **30** point deduction

No submissions will be accepted after 72 hours.

Academic Integrity

All work is to be done individually — group work is not allowed.

While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you cannot work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own, etc. The University's policy is available here: https://dos.uic.edu/community-standards/

In particular, note that **you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance**. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums (e.g. you cannot download answers from Chegg). Other examples of academic dishonesty include emailing your program to another student, sharing your screen so that another student may copy your work, copying-pasting code from the internet, working together in a group, and allowing a tutor, TA, or another individual to write an answer for you.

Academic dishonesty is unacceptable, and penalties range from a letter grade drop to expulsion from the university; cases are handled via the official student conduct process described at the link above.