



CS 341, Spring 2025  
Homework 1  
Due: Monday 1/27/2024 at 11:59pm

## Overview

In this homework assignment, you are going to read about SQL, and then write SQL queries to retrieve data from the SQLite database on Chicago red light camera violations. You can work on Codio, or locally by installing SQLite. Either way, your queries must be saved in **separate .sql files** and submitted to Gradescope for grading.

## Learning SQL

Start by reading and learning some SQL basics. Here's a good site that features integrated SQL databases for practice: <http://sqlzoo.net/>

Work through the following exercises. The associated quizzes are recommended but optional:

SELECT Basics: exercises 1-3

SELECT from World: exercises 1-7

SELECT from Nobel: exercises 1-8

SELECT in SELECT: exercises 1-2

SUM and COUNT: exercises 1-5

In general, here's a good reference site for SQL:  
<http://www.w3schools.com/sql/default.asp>

The documentation for working with dates can be found at:  
[https://www.sqlite.org/lang\\_datefunc.html](https://www.sqlite.org/lang_datefunc.html).



**CS 341, Spring 2025**  
**Homework 1**  
**Due: Monday 1/27/2024 at 11:59pm**

## **The Database**

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:

[https://www.chicago.gov/city/en/depts/cdot/supp\\_info/red-light\\_cameraenforcement.html](https://www.chicago.gov/city/en/depts/cdot/supp_info/red-light_cameraenforcement.html).

This database contains information on 1) the 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](https://data.cityofchicago.org/Transportation/Red-Light-Camera-Violations/spqx-js37/about_data).

As such, there are 2 tables in the database: **Cameras** and **Violations**.

The **Cameras** table provides information on all the red light cameras in Chicago. An intersection may have more than one camera (in fact, this is usually the case).

- **Camera\_ID**: primary key, integer that uniquely identifies each camera
- **Intersection**: string representing the two streets of the intersection, e.g. Western and Cermak
- **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 **Violations** table denotes the number of violations that occurred each day for each camera.

- **Camera\_ID**: foreign key, integer that uniquely identifies each camera
  - This is connected to the **Camera\_ID** of the **Cameras** table
- **Violation\_Date**: string representing the date corresponding to the data, in the format "MM/DD/YYYY"
- **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**  
**Homework 1**  
**Due: Monday 1/27/2024 at 11:59pm**

## Exercises

Create a separate .sql file for each of the following exercises. These files should be submitted to Gradescope.

### Exercise 1:

Find the number of unique intersections in the data. An intersection can have more than one camera, but this exercise is looking for unique intersections, e.g. there are 2 cameras at Western and Cermak, but Western and Cermak should only be counted once for this calculation.

The name of the file for this exercise should be "hw01-01.sql" and it should be done in a single SQL SELECT query.

The answer should be 183.

(Hint: The DISTINCT keyword may be useful.)

### Exercise 2:

Find the address of the camera that corresponds to the ID 2612.

The name of the file for this exercise should be "hw01-02.sql" and it should be done in a single SQL SELECT query.

The answer should be "7100 S WESTERN AVENUE".

### Exercise 3:

List the camera IDs, intersections, and addresses, of all cameras that are located at intersections involving Roosevelt. Order the results by camera IDs, from smallest to largest.

The name of the file for this exercise should be "hw01-03.sql" and it should be done in a single SQL SELECT query.

The answer should be:

1	2233 ROOSEVELT AND HALSTED	800 W ROOSEVELT ROAD
2	2234 ROOSEVELT AND HALSTED	800 W ROOSEVELT ROAD
3	2291 CANAL AND ROOSEVELT	1200 S CANAL STREET
4	2294 CANAL AND ROOSEVELT	500 W ROOSEVELT ROAD
5	2311 ROOSEVELT AND PULASKI	1200 S PULASKI ROAD
6	2314 ROOSEVELT AND PULASKI	4000 W ROOSEVELT ROAD
7	2321 ROOSEVELT AND KOSTNER	1200 S KOSTNER
8	2323 ROOSEVELT AND KOSTNER	4400 W ROOSEVELT

Note that the numbers shown on the left of each row (1-8) are not part of the results, but are provided in case they are helpful to you in understanding the results.



**CS 341, Spring 2025**  
**Homework 1**  
**Due: Monday 1/27/2024 at 11:59pm**

**Exercise 4:**

Find the total number of violations recorded by all the cameras located at the intersection of Roosevelt and Halsted during **2020**.

Also find the total number of violations recorded by all the cameras located at the intersection of Roosevelt and Halsted during **2023**.

The name of the file for this exercise should be "hw01-04.sql". There should be two separate SQL queries for this exercise, one query for 2020 and another query for 2023. For each query, use SELECT in SELECT, rather than hard-coding the camera ID(s).

The answer of the first SQL query for 2020 should be 7388. The answer of the second SQL query for 2023 should be 8189.

**Exercise 5:**

Find the ID of the camera that recorded the highest number of violations across all dates in the database. Also retrieve this number of violations.

The name of the file for this exercise should be "hw01-05.sql" and it should be done in a single SQL SELECT query.

The answer should be camera 1413 with 201709 violations recorded.

**Exercise 6:**

Find the total number of violations on July 4<sup>th</sup> from 2014 to 2024, inclusive. Also retrieve the date. The results should be in order by date, from the least recent to the most recent.

The name of the file for this exercise should be "hw01-06.sql" and it should be done in a single SQL SELECT query.

The answer should be:

1	07/04/2014	1958
2	07/04/2015	1658
3	07/04/2016	1785
4	07/04/2017	1419
5	07/04/2018	1795
6	07/04/2019	1668
7	07/04/2020	2318
8	07/04/2021	2551
9	07/04/2022	1984
10	07/04/2023	2168
11	07/04/2024	1597

Note that the numbers shown on the left of each row (1-11) are not part of the results, but are provided in case they are helpful to you in understanding the results.



**CS 341, Spring 2025**  
**Homework 1**  
**Due: Monday 1/27/2024 at 11:59pm**

## **Submission**

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

Submit your individual query files “hw01-01.sql”, “hw01-02.sql”, etc., under “Homework 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.**

Comments are not necessary.

On Codio, you have the option to “Export as ZIP.” Note that this will export the whole project, including the .db file. Submitting this ZIP file may cause the process of uploading to Gradescope to run slowly. To speed things up, download as .zip, open the .zip, remove the .db file from the .zip, and then upload the much smaller .zip to Gradescope.

**No late submissions will be accepted for this assignment.**

## **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.