This is CS50x

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://orcid.org/0000-0001-5338-2522)
(https://www.quora.com/profile/David-J-Malan)
(https://www.reddit.com/user/davidjmalan)
(https://www.tiktok.com/@davidjmalan) (https://twitter.com/davidjmalan)

Fiftyville

Write SQL queries to solve a mystery.

A Mystery in Fiftyville

The CS50 Duck has been stolen! The town of Fiftyville has called upon you to solve the mystery of the stolen duck. Authorities believe that the thief stole the duck and then, shortly afterwards, took a flight out of town with the help of an accomplice. Your goal is to identify:

- Who the thief is,
- What city the thief escaped to, and
- Who the thief's accomplice is who helped them escape

All you know is that the theft **took place on July 28, 2020** and that it **took place on Chamberlin Street**.

How will you go about solving this mystery? The Fiftyville authorities have taken some of the town's records from around the time of the theft and prepared a SQLite database for you, `fiftyville.db`, which contains tables of data from around the town. You can query that table using SQL `SELECT` queries to access the data of interest to you. Using just the information in the database, your task is to solve the mystery.

Getting Started

Here's how to download this problem into your own CS50 IDE. Log into CS50 IDE (https://ide.cs50.io/) and then, in a terminal window, execute each of the below.

- Navigate to your `pset7` directory that should already exist.
- Execute `wget https://cdn.cs50.net/2020/fall/psets/7/fiftyville/fiftyville.zip` to download a (compressed) ZIP file with this problem's distribution.
- Execute `unzip fiftyville.zip` to uncompress that file.
- Execute `rm fiftyville.zip` followed by `yes` or `y` to delete that ZIP file.
- Execute `ls`. You should see a directory called `fiftyville`, which was inside of that ZIP file.
- Execute `cd fiftyville` to change into that directory.
- Execute `ls`. You should see a `fiftyville.db` file, a `log.sql` file, and an `answers.txt` file.

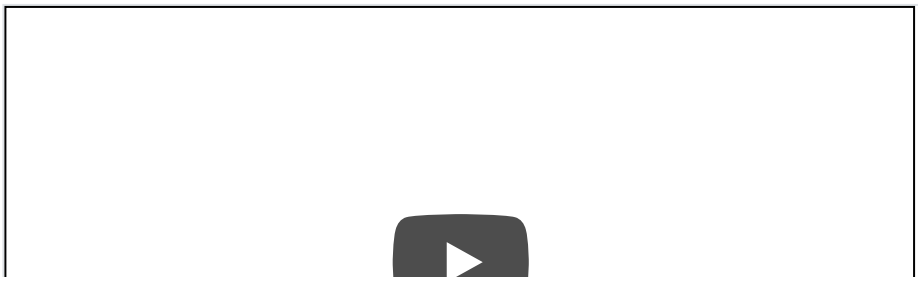
Specification

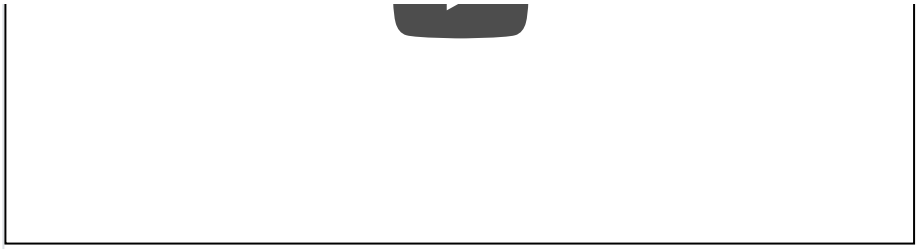
For this problem, equally as important as solving the mystery itself is the process that you use to solve the mystery. In `log.sql`, keep a log of all SQL queries that you run on the database. Above each query, label each with a comment (in SQL, comments are any lines that begin with `--`) describing why you're running the query and/or what information you're hoping to get out of that particular query. You can use comments in the log file to add additional notes about your thought process as you solve the mystery: ultimately, this file should serve as evidence of the process you used to identify the thief!

Once you solve the mystery, complete each of the lines in `answers.txt` by filling in the name of the thief, the city that the thief escaped to, and the name of the thief's accomplice who helped them escape town. (Be sure not to change any of the existing text in the file or to add any other lines to the file!)

Ultimately, you should submit both your `log.sql` and `answers.txt` files.

Walkthrough





Hints

- Execute `sqlite3 fiftyville.db` to begin running queries on the database.
  - While running `sqlite3`, executing `.tables` will list all of the tables in the database.
  - While running `sqlite3`, executing `.schema TABLE_NAME`, where `TABLE_NAME` is the name of a table in the database, will show you the `CREATE TABLE` command used to create the table. This can be helpful for knowing which columns to query!
- You may find it helpful to start with the `crime_scene_reports` table. Start by looking for a crime scene report that matches the date and the location of the crime.
- See this SQL keywords reference (https://www.w3schools.com/sql/sql_ref_keywords.asp) for some SQL syntax that may be helpful!

Testing

Execute the below to evaluate the correctness of your code using `check50`.

```
check50 cs50/problems/2021/x/fiftyville
```

How to Submit

Execute the below, logging in with your GitHub username and password when prompted. For security, you'll see asterisks (`*`) instead of the actual characters in your password.

```
submit50 cs50/problems/2021/x/fiftyville
```

Acknowledgements

Inspired by another case over at SQL City (http://mystery.knightlab.com/).