Carter Zenke (https://carterzenke.me)
carter@cs50.harvard.edu
(https://github.com/carterzenke) (https://www.linkedin.com/in/carterzenke/)

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://www.reddit.com/user/davidjmalan) (https://www.threads.net/@davidjmalan)
(https://twitter.com/davidjmalan)

Final Project

The climax of this course is its final project. The final project is your opportunity to take your newfound savvy with SQL for a spin and develop your very own database. We ask that you build something of interest to you, that you solve a substantive problem, that you positively impact others, or that you change the world. Strive to create something you're proud of.

Since software development is rarely a one-person effort, you are allowed an opportunity to collaborate with one or two classmates for this final project. Needless to say, it is expected that every student in any such group contribute equally to the design and implementation of the project. Moreover, it is expected that the scope of a two- or three-person group's project be, respectively, twice or thrice that of a typical one-person project. Although no more than three students may design and implement a given project, you are welcome to solicit advice from others, so long as you respect the course's policy on academic honesty.

Note that CS50's staff audits submissions to CS50 SQL, including this final project. Students found to be in violation of **the Academic Honesty policy** will be removed from the course and deemed ineligible for a certificate. Students who have already completed CS50 SQL, if found to be in violation, will have their CS50 Certificate (and edX Certificate, if applicable) revoked.

Ideas

The ideas for what you could build are endless, though to get your thinking started, a few others have attempted projects like the below!

- A database to find your favorite songs, representing artists, playlists, albums, and more (à la Spotify (https://spotify.com/) or Apple Music (https://music.apple.com/))
- A database to manage personal finances, storing bank account balances, transactions, budgets, and more (à la Mint (https://mint.intuit.com/), Quicken (https://www.quicken.com/), or your favorite banking app)
- A database to help others find friends when they move to a new city, representing people, cities, events, connections, and more (à la Meetup (https://www.meetup.com/) or Bumble BFF (https://bumble.com/bff))

Getting Started

Template Files

To provide you some structure with which to start building your final project, we've provided you with three template files: `DESIGN.md`, `schema.sql`, and `queries.sql`. Start by downloading these template files and reading on, below!

▶ **Download the template files**

Specification

Your final project should be composed of three files:

- `DESIGN.md`, which is a rigorous design document describing your database's purpose, scope, entities, relationships, optimizations, and limitations. The goal of the design document is to make your thinking visible. Your design document should include:
  - An entity relationship diagram for your database.
  - A video overview, no more than three minutes long.
- `schema.sql`, which is an annotated set of `CREATE TABLE`, `CREATE INDEX`, `CREATE VIEW`, etc. statements that compose your database's schema.
- `queries.sql`, which is an annotated set of `SELECT`, `INSERT`, `UPDATE`, `DELETE`, etc. statements that users will commonly run on your database.

The requirements for each of these components are described in more detail below.

`DESIGN.md`

Within `DESIGN.md`, write about your database's purpose, scope, entities, relationships, optimizations, and limitations. Write for a technical audience that has taken a course such as CS50 SQL. To help you, the template `DESIGN.md` file includes sections with questions for you to answer.

`DESIGN.md` is a *Markdown* file, which allows you to easily format your document using Markdown syntax. If you're new to the format, learn more at markdownguide.org (https://www.markdownguide.org/).

> Your `DESIGN.md` file should include text in all sections prescribed by the template, be minimally multiple paragraphs in length, and explain why you made certain design choices. Ensure you allocate sufficient time and energy to writing a `DESIGN.md` that documents your project thoroughly. Be proud of it! A `DESIGN.md` in the neighborhood of 1000 words is likely to be sufficient for describing your project and all aspects of its functionality. If unable to reach that threshold, that probably means your project is insufficiently complex.

Entity Relationship Diagram

Your `DESIGN.md` file should include an entity relationship diagram for your database. You can create your entity relationship diagram any way you'd like, but allow us to suggest a few!

- If you're the pencil-and-paper type, you can draw your diagram, take a picture, and upload it to your codespace.
- If you're digitally inclined, you can use a tool such as the Mermaid.js live editor (https://mermaid.live/). Mermaid.js is a toolkit via which you can create and export diagrams (including entity relationship diagrams!). See the documentation (https://mermaid.js.org/syntax/entityRelationshipDiagram.html) to learn the relevant syntax and see examples.
- You're also welcome to use any other software that helps you draw the types of shapes you'd like to draw.

When you have an image of your entity relationship diagram, you can embed it in your `DESIGN.md` file using the following syntax:

```
![IMAGE TITLE](FILENAME)
```

Where IMAGE TITLE and FILENAME are your image's title (entirely up to you!) and its filename (e.g., `diagram.jpg`), respectively.

Video Overview

Create a short video (that's no more than 3 minutes in length) in which you present your project to the world, as with slides, screenshots, voiceover, and/or live action. Your video **must** begin with an opening section that displays:

- your project's title;
- your name;
- your GitHub and edX usernames;
- your city and country;
- and, the date you have recorded this video.

See howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen (https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/) for tips on how to make a "screencast," though you're welcome to use an actual camera. Upload your video to YouTube (or, if blocked in your country, a similar site) and take note of its URL; it's fine to flag it as "unlisted," but don't flag it as "private."

`schema.sql`

Your `schema.sql` file should include a set of SQL statements to define your database's schema, annotated with brief SQL comments. Recall that you can use the following syntax to write a SQL comment:

```
-- This is a SQL comment
```

Your `schema.sql` file will likely contain `CREATE TABLE`, `CREATE INDEX`, and `CREATE VIEW` statements.

`queries.sql`

Your `queries.sql` file should include a set of SQL queries typically run on your database, annotated with brief SQL comments.

Your `queries.sql` file will likely contain `SELECT`, `INSERT`, `UPDATE`, and `DELETE` statements.

Sample Project

If you'd find it helpful to see a sample project, consider exploring the staff's own! The sample project creates a database for managing students, instructors, and submissions in this very course.

▶ **Download the sample project**

How to Submit

You must complete all three steps!

Step 1 of 3

Submit this form (https://forms.cs50.io/7911d665-9e2c-4536-8e30-8d02102e9882)!

Step 2 of 3

Execute the `submit50` command below from within your `project` directory (or from whichever directory contains `DESIGN.md`, `schema.sql`, and `queries.sql`).

```
submit50 cs50/problems/2024/sql/project
```

Step 3 of 3

Be sure to visit your gradebook at cs50.me/cs50sql (https://cs50.me/cs50sql) a few minutes after you submit. It's only by loading your Gradebook that the system can check to see whether you have completed the course, and that is also what triggers the (instant) generation of your free CS50 Certificate and the (within 30 days) generation of the Verified Certificate from edX, if you've completed all of the other assignments. Be sure to claim your free certificate (by following the link at the top of your gradebook) before 1 January 2025.

Don't skip the above step! The course is not considered complete until you do the above and see the green banner saying you've completed the course. If you do not do the above prior to 1 January 2025, your status in the course will be subject to the **carryover rules** in the FAQ. The staff will not make any manual corrections in early 2025 based on this being skipped!

That's it! Your project should be graded within a few minutes. If you don't see any results in your gradebook, best to resubmit (running the above `submit50` command). No need to resubmit your form.

This was CS50 SQL!