

CS50's Introduction to Databases with SQL

OpenCourseWare

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

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>)



The logo for [Union Square Donuts](https://www.unionsquaredonuts.com/about)

(<https://www.unionsquaredonuts.com/about>)

Problem to Solve

Around Boston, Brookline, and Somerville, you might catch sight of Union Square Donuts, a shop famous in the area for its delicious confections. Suppose that Union Square Donuts has been around for a while and, while their paper receipts have worked well so far, they're hoping for a system to help them handle more customers. In fact, they've just hired you to design a database fitting for their shop. In exchange, free donuts? 🍩

In a file called `schema.sql` in a folder called `donuts`, write a set of SQL statements to create a database that Union Square Donuts could use to manage their day-to-day operations.

Specification

Your task at hand is to create a SQLite database for Union Square Donuts from scratch, as by writing a set of `CREATE TABLE` statements in a `schema.sql` file. The implementation details are up to you, though you should minimally ensure that your database meets the team's expectations and that it can represent the sample data.

Expectations

To understand the team's expectations for their database, you sat down to talk with them after the shop closed for the day.

Ingredients

We certainly need to keep track of our ingredients. Some of the typical ingredients we use include flour, yeast, oil, butter, and several different types of sugar. Moreover, we would love to keep track of the price we pay *per unit* of ingredient (whether it's pounds, grams, etc.).

Donuts

We'll need to include our selection of donuts, past and present! For each donut on the menu, we'd love to include three things:

- The name of the donut
- Whether the donut is gluten-free (https://en.wikipedia.org/wiki/Gluten-free_diet)
- The price per donut

Oh, and it's important that we be able to look up the ingredients for each of the donuts!

Orders

We love to see customers in person, though we realize a good number of people might order online nowadays. We'd love to be able to keep track of those online orders. We think we would need to store:

- An order number, to keep track of each order internally
- All the donuts in the order
- The customer who placed the order. We suppose we could assume only one customer places any given order.

Customers

Oh, and we realize it would be lovely to keep track of some information about each of our customers. We'd love to remember the history of the orders they've made. In that case, we think we should store:

- A customer's first and last name
- A history of their orders

Sample Data

Your database should be able to represent...

- **Cocoa**, for which Union Square Donuts pays \$5.00 for one pound.
- **Sugar**, for which Union Square Donuts pays \$2.00 for one pound.
- Union Square Donuts' **"Belgian Dark Chocolate" donut**, which is *not* gluten-free, costs \$4.00, and includes the following ingredients:
 - Cocoa
 - Flour
 - Buttermilk
 - Sugar
- Union Square Donuts' **"Back-To-School Sprinkles" donut**, which is *not* gluten-free, costs \$4.00, and includes the following ingredients:
 - Flour
 - Buttermilk
 - Sugar
 - Sprinkles
- **Order 1** from **Luis Singh** for 3 Belgian Dark Chocolate donuts and 2 Back-To-School Sprinkles donuts.

Usage

To create a database from your schema, within your `donuts` folder, type

```
sqlite3 donuts.db
```

The above will create an empty SQLite database called `donuts.db`.

Then, in the `sqlite3` prompt, type

```
.read schema.sql
```

to read the statements from `schema.sql`.

Recall you can use `DROP TABLE tablename`, where `tablename` is the name of your table, to delete a table from your database.

How to Test

While `check50` exists for this problem, only you can ensure your database meets the team's expectations and that it can store the sample data efficiently. Consider whether your database is fully normalized!

Correctness

```
check50 cs50/problems/2023/sql/donuts
```

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2023/sql/donuts
```