# Lab 13: SQL (Optional) lab13.zip (lab13.zip)

*Due by 11:59pm on Tuesday, November 24.*

## Starter Files

Download lab13.zip (lab13.zip). Inside the archive, you will find starter files for the questions in this lab, along with a copy of the Ok (ok) autograder.

## Submission

**THIS LAB IS OPTIONAL!!** We encourage you to do it if you have time, but you will receive credit whether or not you submit the lab.

## Topics

SQLite

Aggregation

## Suggested Questions

## Cyber Monday

### Q1: Price Check

After you are full from your Thanksgiving dinner, you realize that you still need to buy gifts for all your loved ones over the holidays. However, you also want to spend as little money as possible (you're not cheap, just looking for a great sale!).

Let's start off by surveying our options. Using the `products` table, write a query that creates a table `average_prices` that lists categories and the average price of items in the category (using MSRP (https://en.wikipedia.org/wiki/List_price) as the price).

You should get the following output:

```
sqlite> SELECT category, ROUND(average_price) FROM average_prices;
computer|109.0
games|350.0
phone|90.0
```

```
CREATE TABLE average_prices AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q cyber-monday-part1
```

## Q2: The Price is Right

Now, you want to figure out which stores sell each item in products for the lowest price. Write a SQL query that uses the `inventory` table to create a table `lowest_prices` that lists items, the stores that sells that item for the lowest price, and the price that the store sells that item for.

You should expect the following output:

```
sqlite> SELECT * FROM lowest_prices;
Hallmart|GameStation|298.98
Targive|QBox|390.98
Targive|iBook|110.99
RestBuy|kBook|94.99
Hallmart|qPhone|85.99
Hallmart|rPhone|69.99
RestBuy|uPhone|89.99
RestBuy|wBook|114.29
```

```
CREATE TABLE lowest_prices AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q cyber-monday-part2
```

## Q3: Bang for your Buck

You want to make a shopping list by choosing the item that is the best deal possible for every category. For example, for the "phone" category, the uPhone is the best deal because the MSRP price of a uPhone divided by its ratings yields the lowest cost. That means that uPhones cost the lowest money per rating point out of all of the phones. Note that the item with the lowest MSRP price may not necessarily be the best deal.

Write a query to create a table `shopping_list` that lists the items that you want to buy from each category.

After you've figured out which item you want to buy for each category, add another column that lists the store that sells that item for the lowest price.

You should expect the following output:

```
sqlite> SELECT * FROM shopping_list;
GameStation|Hallmart
uPhone|RestBuy
wBook|RestBuy
```

```
CREATE TABLE shopping_list AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

**Hint**: You should use the `lowest_prices` table you created in the previous question.

**Hint 2**: One approach to this problem is to create another table, then create shopping_list by selecting from that table.

Use Ok to test your code:

```
python3 ok -q cyber-monday-part3
```

## Q4: Driving the Cyber Highways

Using the Mb (megabits) column from the `stores` table, write a query to calculate the total amount of bandwidth needed to get everything in your shopping list.

```
CREATE TABLE total_bandwidth AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

**Hint**: You should use the `shopping_list` table you created in the previous question.

Use Ok to test your code:

```
python3 ok -q cyber-monday-part4
```

# Troubleshooting/Advanced SQLite

## Troubleshooting

Python already comes with a built-in SQLite database engine to process SQL. However, it doesn't come with a "shell" to let you interact with it from the terminal. Because of this, until now, you have been using a simplified SQLite shell written by us. However, you may find the shell is old, buggy, or lacking in features. In that case, you may want to download and use the official SQLite executable.

If running `python3 sqlite_shell.py` didn't work, you can download a precompiled sqlite directly by following the following instructions and then use `sqlite3` and `./sqlite3` instead of `python3 sqlite_shell.py` based on which is specified for your platform.

Setup

## CS 61A (/~cs61a/fa20/)

Weekly Schedule (/~cs61a/fa20/weekly.html)

Office Hours (/~cs61a/fa20/office-hours.html)

Staff (/~cs61a/fa20/staff.html)

## Resources (/~cs61a/fa20/resources.html)

Studying Guide (/~cs61a/fa20/articles/studying.html)

Debugging Guide (/~cs61a/fa20/articles/debugging.html)

Composition Guide (/~cs61a/fa20/articles/composition.html)

## Policies (/~cs61a/fa20/articles/about.html)

Assignments (/~cs61a/fa20/articles/about.html#assignments)

Exams (/~cs61a/fa20/articles/about.html#exams)

Grading (/~cs61a/fa20/articles/about.html#grading)