Homework 3

Homework 3: Essential SQL

- Please share your completed assignments with Ananya & Thomas: ananya.jha@mail.utoronto.ca ; thomas.rosenthal@utoronto.ca
- Submissions can be in the form of a zip folder, Dropbox link, Google Drive link, etc
- Due on Saturday, January 13 at 11:59pm
- Weight: 10% of total grade

AGGREGATE

1. Write a query that determines how many times each vendor has rented a booth at the farmer's market by counting the vendor booth assignments per `vendor_id`.
2. The Farmer's Market Customer Appreciation Committee wants to give a bumper sticker to everyone who has ever spent more than $50 at the market. Write a query that generates a list of customers for them to give stickers to, sorted by last name, then first name.
   HINT: This query requires you to join two tables, use an aggregate function, and use the HAVING keyword.

Temp Table

1. Insert the original vendor table into a temp.new_vendor and then add a 10th vendor: Thomass Superfood Store, a Fresh Focused store, owned by Thomas Rosenthal
   HINT: This is two total queries -- first create the table from the original, then insert the new 10th vendor. When inserting the new vendor, you need to appropriately align the columns to be inserted (there are five columns to be inserted, I've given you the details, but not the syntax)

To insert the new row use VALUES, specifying the value you want for each column:
`VALUES(col1,col2,col3,col4,col5)`

Date

1. Get the customer_id, month, and year (in separate columns) of every purchase in the customer_purchases table.
   HINT: you might need to search for strfrtime modifers sqlite on the web to know what the modifers for month and year are!

COALESCE

1. Our favourite manager wants a detailed long list of products, but is afraid of tables! We tell them, no problem! We can produce a list with all of the appropriate details.

Using the following syntax you create our super cool and not at all needy manager a list:

```
SELECT
product_name || ', ' || product_size|| ' (' || product_qty_type || ')'
FROM product
```

But wait! The product table has some bad data (a few NULL values).
Find the NULLs and then using COALESCE, replace the NULL with a blank for the first problem, and 'unit' for the second problem.

HINT: keep the syntax the same, but edited the correct components with the string. The `||` values concatenate the columns into strings. Edit the appropriate columns -- you're making two edits -- and the NULL rows will be fixed. All the other rows will remain the same.