Exercise

For the product table:

1. Products can be sold by the individual unit or by bulk measures like lbs. or oz. Write a query that outputs the product_id and product_name columns from the product table, and add a column called prod_qty_type_condensed that displays the word “unit” if the product_qty_type is “unit,” and otherwise displays the word “bulk.”

2. We want to flag all of the different types of pepper products that are sold at the market. Add a column to the previous query called pepper_flag that outputs a 1 if the product_name contains the word “pepper” (regardless of capitalization), and otherwise outputs 0.

Exercise

1. Write a query that INNER JOINs the vendor table to the vendor_booth_assignments table on the vendor_id field they both have in common, and sorts the result by vendor_name, then market_date.

2. Is it possible to write a query that produces an output identical to the output of the following query, but using a LEFT JOIN instead of a RIGHT JOIN?

```
SELECT *
FROM customer AS c
RIGHT JOIN customer_purchases AS cp
    ON c.customer_id = cp.customer_id
```

Exercise

1. Write a query that determines how many times each vendor has rented a booth at the farmer's market. In other words, count the vendor booth assignments per vendor_id.

2. We asked earlier "When is each type of fresh fruit or vegetable in season, locally?" Write a query that displays the product category name, product name, earliest date available, and latest date available for every product in the "Fresh Fruits & Vegetables" product category.

3. 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.)