Exercises

1. Do the following two steps:

   a. Write a query that selects from the customer_purchases table and numbers each customer's visits to the farmer's market (labeling each market date with a different number). Each customer's first visit is labeled 1, second visit is labeled 2, etc. (We are of course not counting visits where no purchases are made, because we have no record of those.) You can either display all rows in the customer_purchases table, with the counter changing on each new market date for each customer, or select only the unique market dates per customer (without purchase details) and number those visits.
   HINT: One of these approaches uses ROW_NUMBER() and one uses DENSE_RANK().

   b. Reverse the numbering of the query from a part so each customer's most recent visit is labeled 1, then write another query that uses this one as a subquery and filters the results to only the customer's most recent visit.

Exercises

2. Using a COUNT() window function, include a value along with each row of the customer_purchases table that indicates how many different times that customer has purchased that product_id.

3. In the last query we used LAG and sorted by market_date. Can you think of a way to use LEAD in place of LAG, but get the exact same output?

Exercises

1. Get the customer_id, month, and year (in separate columns) of every purchase in the farmers_market.customer_purchases table.

2. Write a query that filters to purchases made in the past two weeks, returns the earliest market_date in that range as a field called sales_since_date, and a sum of the sales (quantity * cost_to_customer_per_qty) during that date range.

   - Your final answer should use the CURDATE() function, but if you want to test it out on the Farmer's Market database, you can replace your CURDATE() with the value '2019 03-31' to get the report for the two weeks prior to March 31, 2019 (otherwise your query will not return any data, because none of the dates in the database will have occurred within two weeks of you writing the query).