

# Homework 4

---

## Homework 4: Advanced SQL

- Please upload your completed assignments to Google Drive.
- Due on Saturday, March 2 at 11:59pm
- Weight: 8% of total grade
- Upload one .sql file with your queries

## Windowed Functions

---

1. 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.

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().

2. 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 (or temp table) and filters the results to only the customer's most recent visit.
3. 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.

## String manipulations

---

1. Some product names in the product table have descriptions like "Jar" or "Organic". These are separated from the product name with a hyphen. Create a column using SUBSTR (and a couple of other commands) that captures these, but is otherwise NULL. Remove any trailing or leading whitespaces. Don't just use a case statement for each product!

product_name	description
Habanero Peppers - Organic	Organic

**HINT:** you might need to use INSTR(product\_name, '-') to find the hyphens. INSTR will help split the column.

## UNION

---

1. Using a UNION, write a query that displays the market dates with the highest and lowest total sales.

**HINT:** There are a possibly a few ways to do this query, but if you're struggling, try the following: 1) Create a CTE/Temp Table to find sales values grouped dates; 2) Create another CTE/Temp table with a rank windowed function on the previous query to create "best day" and "worst day"; 3) Query the second temp table twice, once for the best day, once for the worst day, with a UNION binding them.