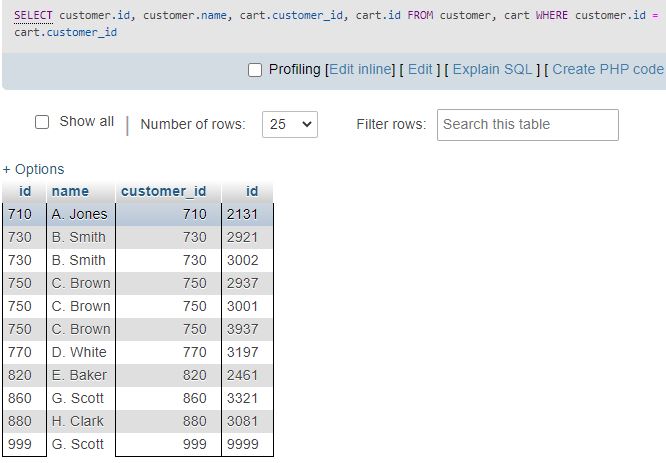
# Ryan Demboski

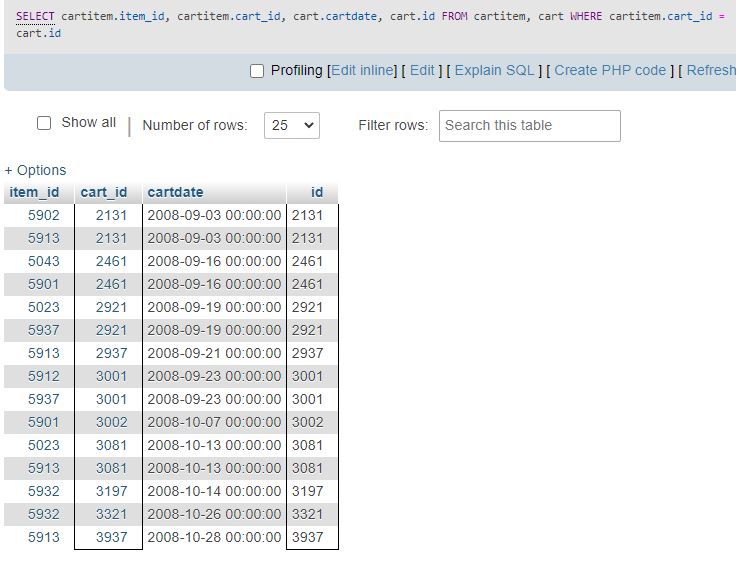
# Joining tables

The following exercises use the “supermarket” database

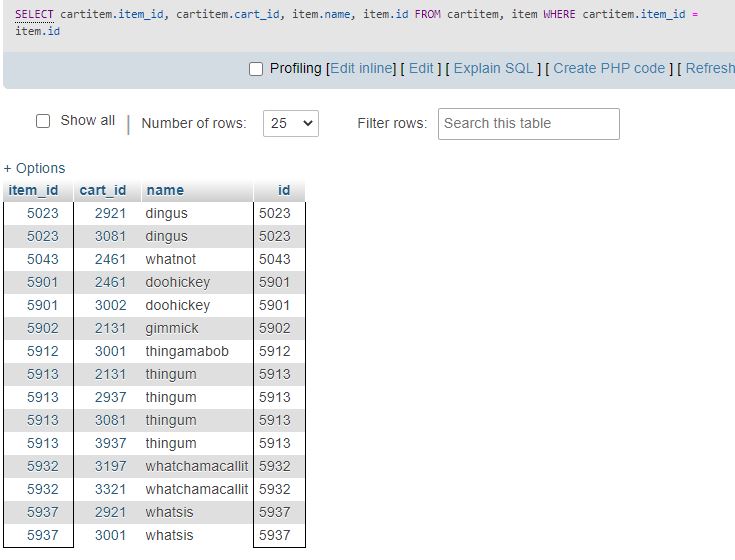
1. Join tables customer and cart.



1. Join tables cart and cartitem.



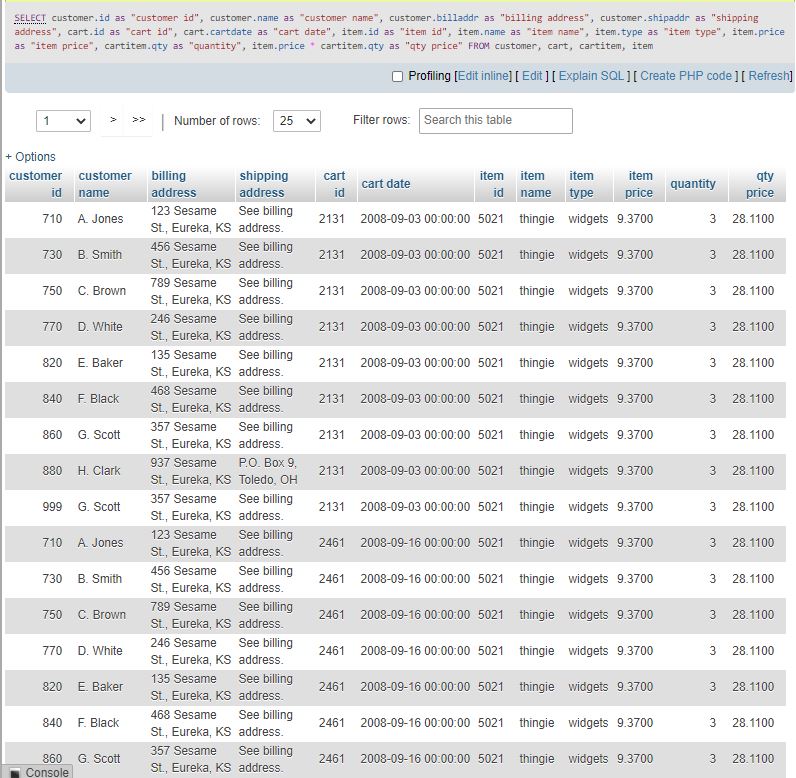
1. Join tables cartitem and item.



1. Return, with the very same attribute names, 'customer id', 'customer name', 'billing address', 'shipping address', 'cart id', 'cart date', 'item id', 'item name', 'item type', 'item price', 'quantity'.



1. Add on to 4 such that you also return the price of an item multiplied by its quantity. What did you need to add on?

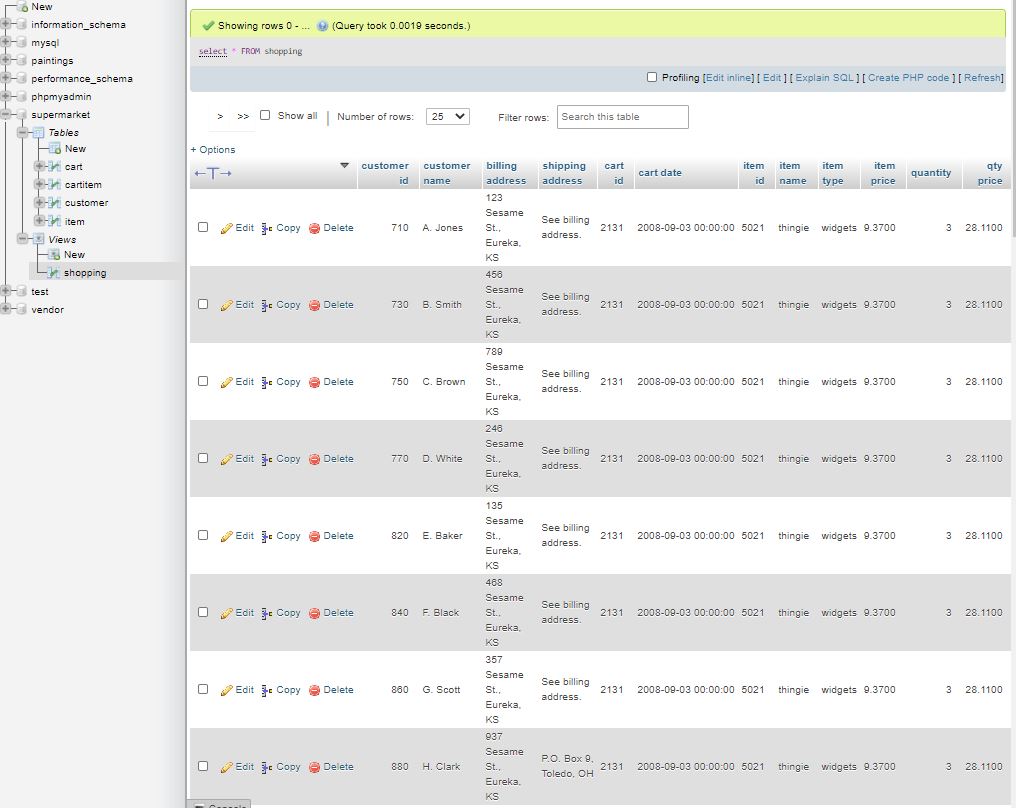


Added a new attribute by multiplying 2 the existing attributes and storing the result under a new name

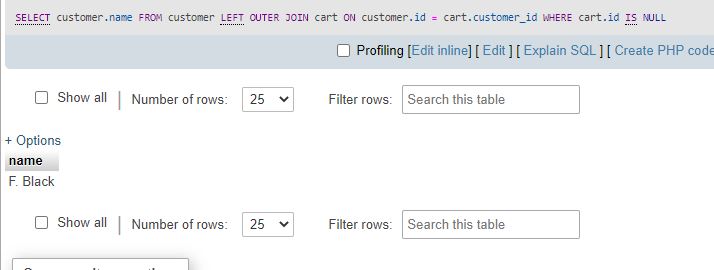
1. What is a database view?

It’s the printed result data of a stored query on that data -- it’s a virtual table from data existing in the database and accessible when that view is requested by the user.

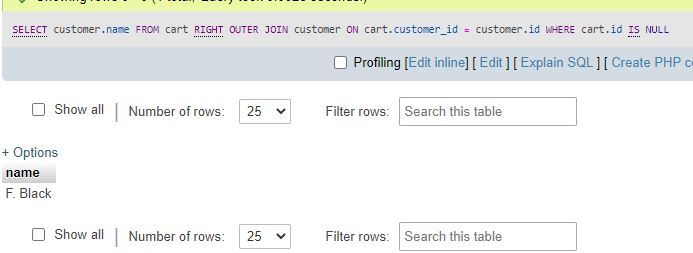
1. Create a view called 'shopping' that returns all the pieces of information of problem 4.



1. Create an (outer) join that returns the names of customers that don’t have a cart.

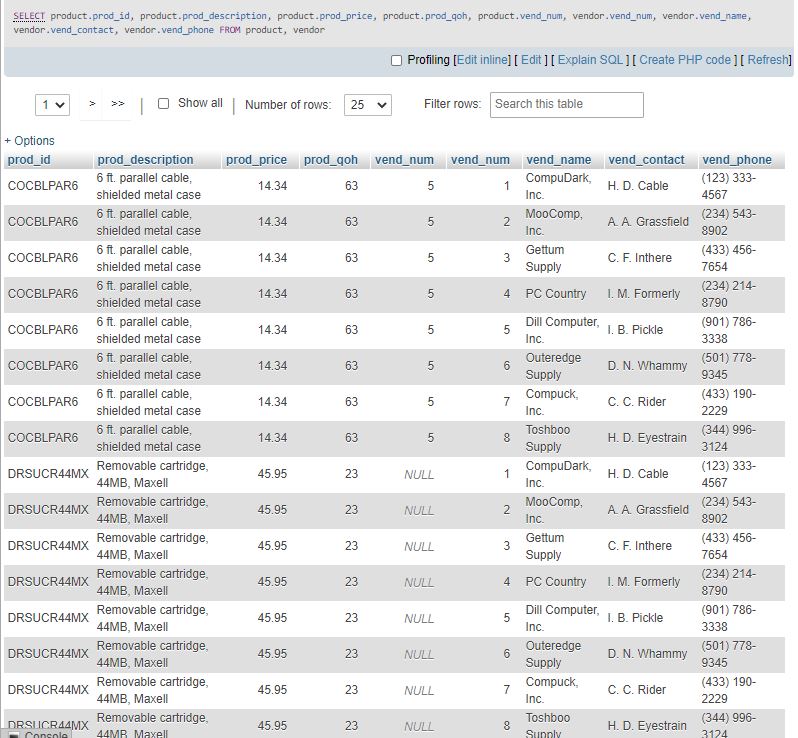


1. See the last problem. Whichever table you named first, switch the positions of the tables and modify your outer join such that the result is exactly the same.

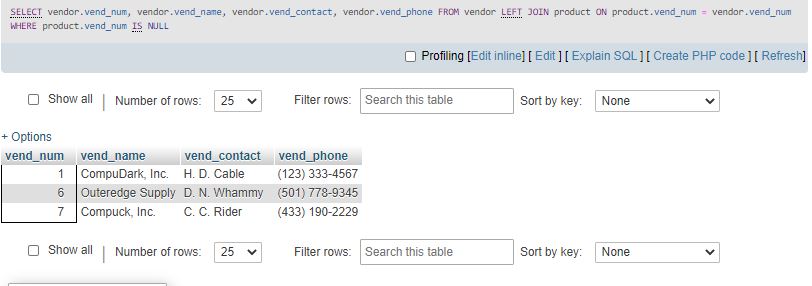


For the following problems refer to the "vendor" database.

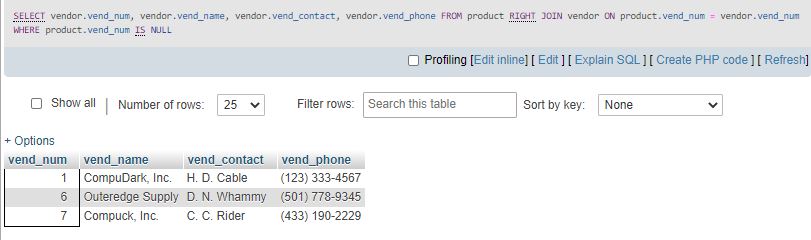
1. Create a join of all data that is there.



1. Create a join that lists all vendor information of vendors that have no product.

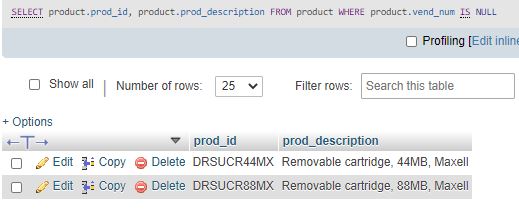


1. See last question: switch the position of the two tables



Not by way of joins, but still interesting:

1. Create a query that returns products that don’t have a vendor



1. Are products optional or mandatory to vendors?

Products are optional to vendors. Not every product needs a vendor. As seen in problem 13, 2 products came back without a vendor.