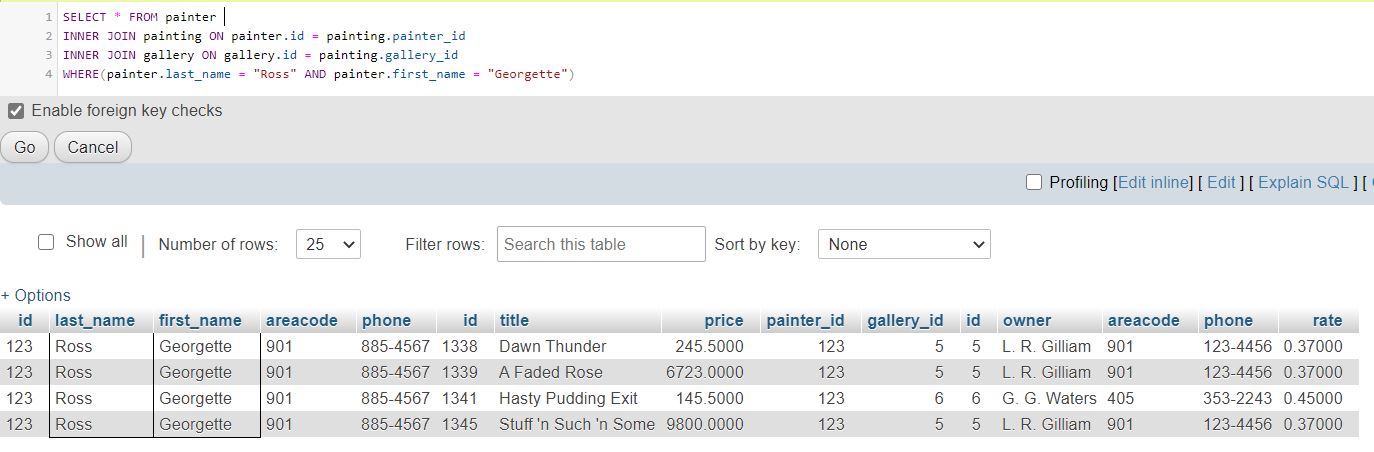
# Ryan Demboski

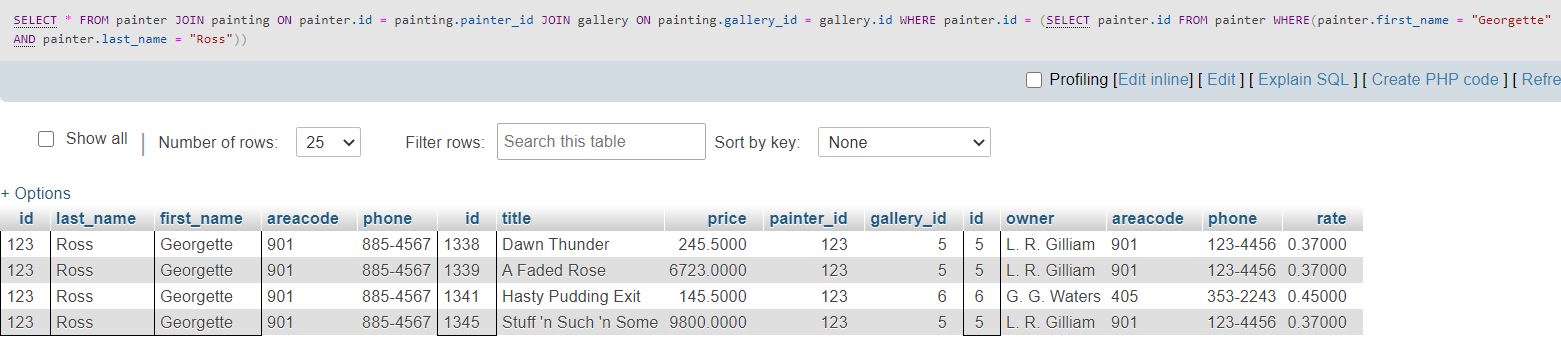
# Subqueries

The following exercises use the “paintings” database that I will hand out in class.

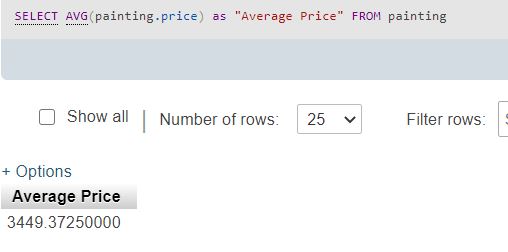
1. Get all accessible information about a painter with first name Georgette and last name Ross. Use a simple Join and Where clause.



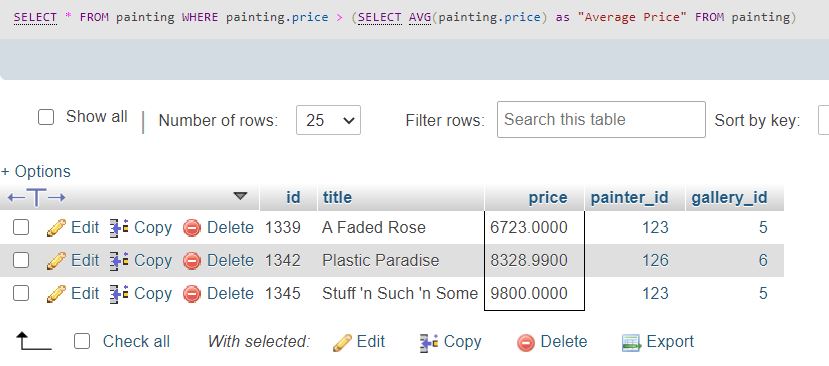
1. See the problem above. Achieve the same with a subquery in the Where clause that returns that painters id. This is not very desirable, but demonstrates how to use a subquery in the Where clause.



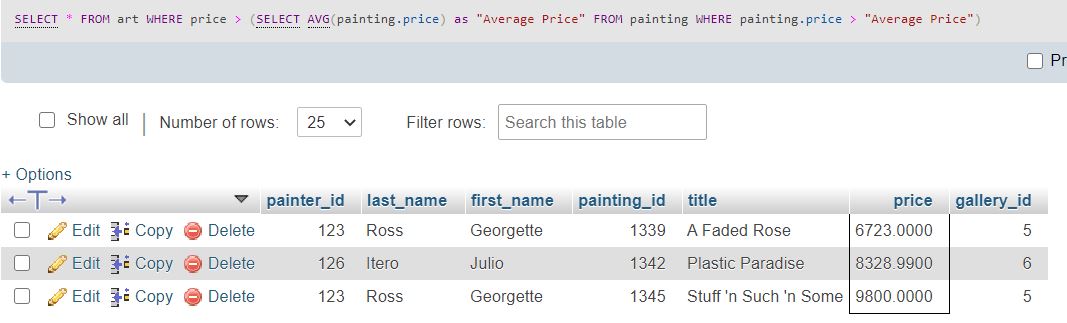
1. In preparation for the next problem below, write a simple query that returns the average price of all paintings.



1. Return all accessible information about paintings that have a higher than average price.



1. Return all accessible information about paintings of a painter that are more expensive than the average price of all paintings of the respective painter. Use a correlated query in the WHERE clause. For simplicity, also use a convenient view, instead of a join in the outer query.



1. Return the most expensive painting per painter. Start out with simple grouping. Does it work?  
     
   No, it does not work. Painting title and painting id are not showing. Subquery is needed.
2. See the last problem. Use a correlated query instead. Again, for convenience, use a view.  
     
   
3. See the last problem, but avoid the cost of correlated queries by creatively using joining, and grouping.  
     
   