Project Three: Advanced SQL

You may work on these problem sets together (citing your collaborators), but the write up must be your own. Assignments must be submitted digitally;

---

1. (10 Points) Using the ***MealPlanning*** database, write a function that determines if an ingredient is in the *Ingredient* table.

2. In our in class exercise, we wrote a stored procedure that added a new recipe to the ***MealPlanning*** database, taking into account that it might need to insert a new entry into the *Cookbook* table, along with a new entry into the *Recipe* table. But, we did not touch the *Ingredient* or the *Meal* table, so clearly our interface for inserting new data into this database is incomplete.
    a. (5 Points) Use the stored procedure from the in class exercise, and insert your favorite recipe into the ***MealPlanning*** database.
    b. (30 Points) Write a stored procedure that inserts all the needed ingredients for your favorite recipe, ensuring that:
        i. No duplicate ingredients are inserted. You should use the function that you created in question one.
        ii. The *Meal* table is properly updated, mapping all your ingredients to your favorite recipe.

3. (15 Points) Write a stored procedure that takes an input a recipe name, and returns all the ingredients used in that recipe, in alphabetical order.

4. (10 Points) Write a view that displays the names of all the recipes, the cookbooks they are in, and the ingredients they contain. The results should be ordered by cookbook name, then by recipe name and then by ingredient name.

5. (20 Points) Write a query that tells me how many ingredients are contained within each recipe. Remember that aggregate functions are useful when you can group tuples/rows together. Also, let's order the results by the number of ingredients, in descending order.

6. (10 Points) Write a query to tell me what ingredients are not contained within your favorite recipe. Make sure you order the ingredients in alphabetical order.