ETL Project
September 13, 2021

Three Wise Desserts

Ericka Houle, Kelli Smith and Kidist Gebremedhin

Introduction:

Three Wise Desserts will be your go to database for desserts. It will allow you to query by name, ingredients and general cooking information.

Inspiration:

Our inspiration has been taken from our childhoods where we grew up learning how to bake from our parents. We have taken inspiration from other websites and how they allow users to sort and search for different recipes.

Sources of Data:

We were able to utilize the following websites to extract multiple individual dessert recipe URL's:

- https://www.allrecipes.com/recipes/79/desserts/
- https://www.allrecipes.com/search/results/?search=desserts
- https://www.allrecipes.com/search/results/?search=cake
- https://www.allrecipes.com/search/results/?search=cupcake
- https://www.allrecipes.com/search/results/?search=pie
- https://www.allrecipes.com/search/results/?search=cookies
- https://www.allrecipes.com/search/results/?search=chocolate
- https://www.allrecipes.com/recipes/1556/desserts/frozen-desserts/ice-cream/
- https://www.allrecipes.com/recipes/1684/healthy-recipes/sugar-free/desserts/

From this list of URL's we were able to extract specific recipe information to create our tables.

Extraction:

When deciding what we wanted to extract from the individual recipe pages we decided to focus on the following information:

- Dessert URL
- Dessert Name
- Ingredients
- Times: Prep time, cook time, additional time and total time
- Servings
- Recipe Yield
- Star Rating
- Rankings
- Short description of recipe
- Directions

One of the biggest challenges we faced during extraction process was the inconsistency of the information available on each recipe page. For example, not every recipe had ratings, reviews, or prep time or even cook time. To solve for this, we used multiple Try-Except statements in our code to append NaN when a specific data point was not available.

Transformation:

The information we got from the individual recipe websites was relatively clean, however the challenge arose when we hit the ingredient list. Each ingredient was listed as a single string that consisted of quantity, measurement type and ingredient. To insert data into our database, we needed to separate these 3 components. This was accomplished by using the re python library that allowed us to separate each string on a list of delimiters. The delimiter list included the different types of measurements, such as cup, tablespoons, ounces, and our favorite, tiny pinch. Everything to the left of the delimiter was captured as the quantity, and everything to the right of the ingredient was captured as the ingredient.

Next, we realized that our quantities included fractions that were represented as symbols. We located each instance of a symbol using the .loc method and replaced it with its corresponding decimal value.

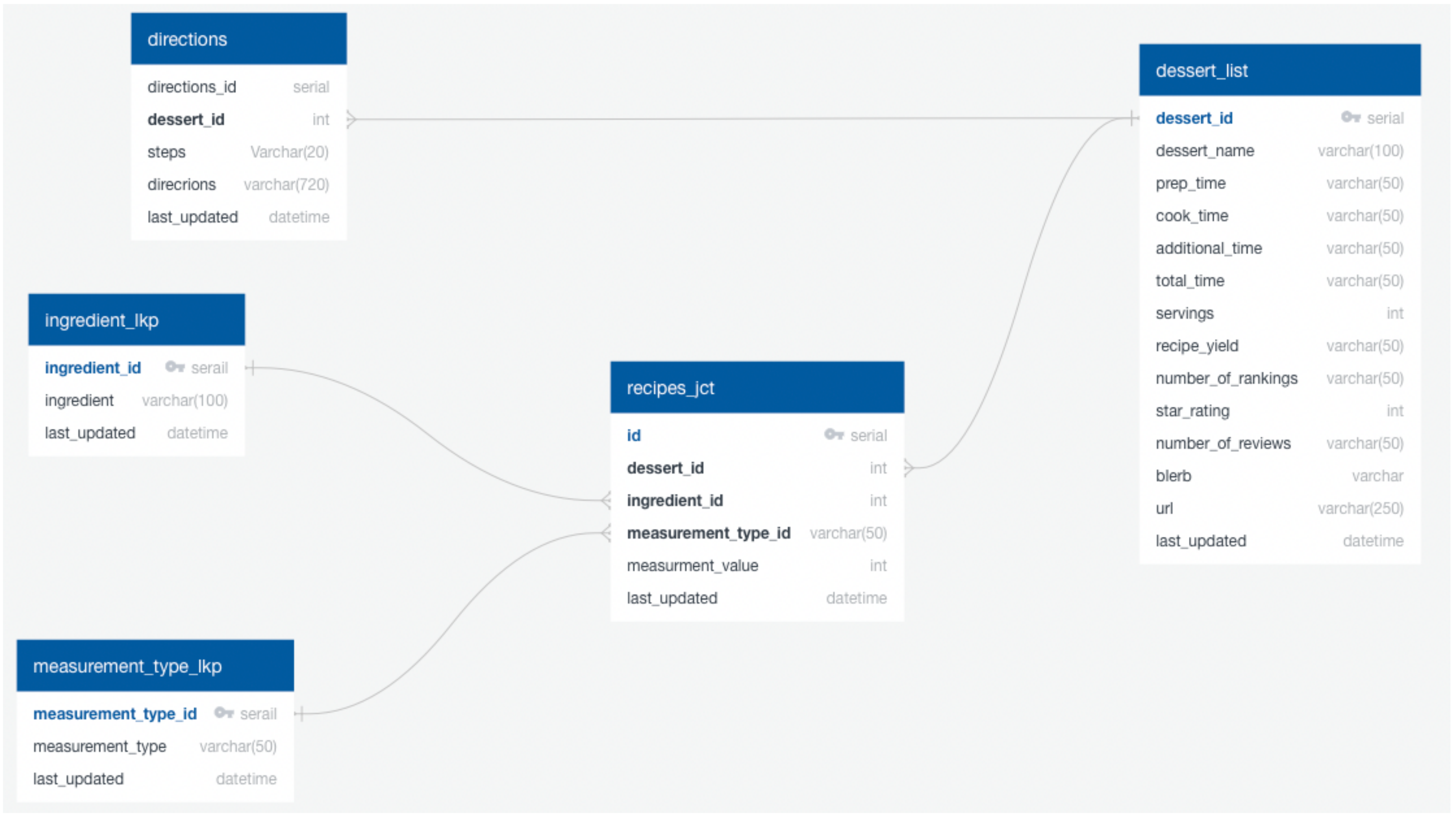
Loading:

Utilizing DBeaver we created five different tables and then loaded them into our google cloud PostgreSQL data base. Our five tables included: Measurement Type Lookup, Ingredient Lookup, Desserts List, Directions and a Recipe Junction tables.

We had to pay particular attention to the order in which we loaded our tables because of the many dependances we had in place. First, we loaded the dessert_list table, which included the recipe metadata and generated a unique ID per recipe that would be referenced as a foreign key in later tables. Next, we loaded the ingredients_lkp and measurement_type_lkp tables. Unique IDs were generated for the unique ingredients and measurements that would later be referenced as foreign keys in our master recipe_jct table. Finally, we were able to load the directions and recipes_jct tables because all foreign key dependencies had been met.

ETL Project
September 13, 2021

See figure below for table structures:



Future Work:

While creating our tables we noticed that number of ratings and reviews were in the form of strings. If we had additional time, we would convert the ratings and reviews into integers for additional analysis and queries.

We ended up hard coding the conversion between factions and decimals. If we had additional time we would have researched and made code modifications to make this an automated function. This would allow our code to be more dynamic and not require code modifications as recipes are added to our database.

We noticed that some of the reviews gave some helpful information regarding possible recipe modifications along with their review. We would like to further scrape this information by looking at the top 5 most helpful reviews.

Login Credentials:

Host: 35.238.117.133
Database: postgres
Username: postgres
Password: password