Users

This table stores user data for users to log back into later, saving their contributions and allow them to build a reputation on the website
Many to Many - There can be an endless amount of users, and these users interacts with many different tables

- Email
- Password hash
- Username
- User_id
- First Name
- Last Name
- isTopContributor

Recipes

This table hosts ingredient data and post data for recipes posted to the website
One to many - Many users can access these recipes, but only one copy of the original recipe needs to exist

- isPublic - boolean
- Recipe name
- Recipe owner user_id
- Instructions
- Recipe description
- Likes user_id
- Comments/reviews user_id

Ingredients

One to one - Only one ingredients list can be given to one recipe, however individual ingredients can apply to multiple recipes depending on what the recipes call for
This table holds data for what groceries are used in a given recipe

- list_id
- Ingredient_name - QTY as integer value

Grocery List

One to Many - Many users can choose to add the groceries to their cart to try out a recipe

This table holds data for what groceries to buy for the recipe the user wants to make at home

- List_id - ingredient list to reference
- Preferred_online_store (Which website user wants to buy grocery list from. Ex. Amazon, HelloFresh, Walmart, Target, InstaCart)

— Postgres commands

```
CREATE TABLE users (
  user_id SERIAL PRIMARY KEY,
  email VARCHAR(75),
  password_hash VARCHAR(500),
  isTopContributor boolean,
  username VARCHAR(25),
  first_name VARCHAR(40),
  last_name VARCHAR(40)
  )

CREATE TABLE ingredients(
  list_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(user_id),
  recipe_id INT REFERENCES recipes(recipe_id),
  ingredients TEXT
  )

ALTER TABLE ingredients
  ADD recipe_id INT REFERENCES recipes(recipe_id)
```

```
CREATE TABLE groceries(
  list_id INT REFERENCES ingredients(list_id),
  user_id INT REFERENCES users(user_id),
  grocery_id SERIAL PRIMARY KEY,
  preferred_store VARCHAR(30)
  )

CREATE TABLE recipes(
  recipe_id SERIAL PRIMARY KEY,
  recipe_name VARCHAR(60),
  user_id INT REFERENCES users(user_id),
  isPublic BOOLEAN,
  recipe_description TEXT,
  like_counter INT,
  list_id INT REFERENCES ingredients(list_id),
  allow_comments BOOLEAN
  )

CREATE TABLE comments(
  user_id INT REFERENCES users(user_id),
  content VARCHAR(300),
  recipe_id INT REFERENCES recipes(recipe_id),
  comment_id SERIAL PRIMARY KEY
  )

CREATE TABLE recipe_likes(
  post_like_id SERIAL PRIMARY KEY,
  recipe_id INT REFERENCES recipes(recipe_id),
  user_id INT REFERENCES users(user_id)
  )
```