Scenario

Summary

We want to create a recipe creating/sharing and grocery list app. You'll be planning out what tables we'll need, what information they'll store, and how the data will relate to each other.

Features

- users can sign into the app with their email and password
- users can create recipes with ingredients and instructions
- recipes can be marked as public or private
- users can view other people's recipes
- ingredients from recipes can be added to user's grocery lists
- users can create their own occasions and assign recipes to occasions

Users:

- Email
- Password
- user name
- Id
- Birthday
- Phone #

Recipes

- Recipe ID
- Text
- Photo
- ingredients
- Keywords
- occasion
- Public?

Grocery Lists

- Ingredients
- List id

- Title this next section "Table Ideas".
- Based off the ideas you just brainstormed, list out what tables you think you'll need. It's okay if you change it up later.

- Write a brief description of each table. For example: “products: this table will hold information about the products offered, each row will be an individual product”.

Table ideas

Users: information about users

- email
- password
- user name
- user_id
- birthday
- phone #

Recipes: information about recipes, each column an ingredient, users by row
Grocery Lists: informationrecipe_id

- text
- image
- ingredients
- keywords
- occasion
- public?

about the grocery lists, users and ingredients by column

- ingredients
- list_id
- User_id
- recipe_ids/

Ingredients: holds all ingredients and which recipes and lists they belong to in rows

- ingredient_id
- Recipe_ids
- grocery_list_ids

Relationships

One-to-Many

- User -> recipes
- User -> grocery lists

Many to Many

- Recipes -> ingredients
- Grocery lists →ingredients

One to One

Columns

- Users
    - Email - login and contact, includes all characters
    - Password - login, includes all characters
    - user_name - interact, display, identify, includes all characters
    - User_id- indentify, it's whole + numbers
    - birthday- security, it's a date
    - phone #-security and contact, it's whole + numbers
    -
- Recipes
    - Recipe_id, identify, it will be an integer
    - Name-to identify recipe, it will be text nd likely shorter
    - Text- instructions for recipe, it's text and not sure how long it will be
    - Image- social aspect, attractive display, it's a text url and not sure how long it will be
    - Ingredients- necessary for recipe, from ingredients list
    - Keywords-to search for recipes, will be text
    - Occasion- we had to include it, will be text
    - Public?- we had to include it, it will be either public or not public(private)
    -
- Grocery Lists
    - Ingredients_id - it's the ingredients to buy, references ingredients list
    - Name- to identify it for user, will be a variety of characters and not too long
    - List_id- to identify the list, will be integer
    - User_id- to identify the user the list belongs to, references user list
    - Recipe_ids- to identify recipe the list belongs to, references recipe list
-
- Ingredients
    - Ingredients_id- to identify ingredient, will be integer
    - Name-so we know what it is, will be a variety of characters and not too long
    - List_id- to add it to grocery lists, references grocery list
    - Recipe_ids-to add it to recipe lists, references recipe list
    -

```
CREATE TABLE users(
  user_id SERIAL PRIMARY KEY,
  name VARCHAR(75) NOT NULL,
  email VARCHAR(100) NOT NULL,
  password VARCHAR(500) NOT NULL,
  phone_number VARCHAR(11) NOT NULL,
  birth_date DATE NOT NULL
);

CREATE TABLE recipes(
  recipe_id SERIAL PRIMARY KEY,
  name VARCHAR(75) NOT NULL,
```

```
  instructions VARCHAR(1500) NOT NULL,
  image TEXT,
  keywords VARCHAR(75) NOT NULL,
  occasion VARCHAR(75) NOT NULL,
  public BOOLEAN
);

CREATE TABLE grocery_lists(
  grocery_list_id SERIAL PRIMARY KEY,
  name  VARCHAR(75) NOT NULL,
  user_id INTEGER NOT NULL REFERENCES users(user_id),
 recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)
);

CREATE TABLE ingredients(
  ingredient_id SERIAL PRIMARY KEY,
  name  VARCHAR(75) NOT NULL,
  list_id INTEGER NOT NULL REFERENCES grocery_lists( grocery_list_id),
 recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)
);

ALTER TABLE recipes
ADD ingredients INTEGER NOT NULL REFERENCES ingredients(ingredient_id)
```