

## 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

## Brainstorming:

- **Users**
  - User\_id
  - User\_password
  - User\_email
- **Ingredients**
  - Ingredient\_id
  - Ingredient\_name
- **Recipe**
  - Recipe\_id
  - Recipe\_instructions\*\*
  - Recipe\_Public(Boolean)
  - User\_id \*\*
  - //Ingredient\_id\*\*
  - Grocery\_list\_id\*\*
- **Grocery List**
  - Grocery\_list\_id
  - Ingredient\_id\*\*
- **Occasion**
  - Occasion\_id
  - recipe\_id\*\*
- **Recipe\_share**
  - View\_User\_id
  - recipe\_id\*\*

- Recipe\_Public(Boolean)\*\*
- **Recipe\_instructions**
  -

## RELATIONSHIPS

- **One to One**
  - User to Boolean
    - User and boolean are one to one because: the recipe being public is directly decided by the user
- **One to Many**
  - User ⇒ Ingredients
    - User and ingredients are one to many because: one user can add in multiple ingredients
  - User ⇒ Recipes
    - User and recipes are one to many because: one user can create multiple recipes
  - User ⇒ Occasion
    - User and occasion are one to many because: one user can add in multiple occasions
- **Many to Many**
  - Recipe\_share ⇔ Users
    - Recipe\_share and users are many to many because: Many recipes can be shared with many users and vise versa
  - Ingredients ⇔ Grocery Lists
    - Ingredients and grocery lists are many to many because many ingredients can be apart of various grocery lists
  - Ingredients ⇔ recipes
    - Ingredients and recipes are many to many because many ingredients can be apart of various recipes and vise versa
    -
  - Recipes ⇔ occasion
    - Recipes and occasions are a many to many relationship because many different recipes can be apart of various occasions and vise versa

## Columns

- **Users**

- User\_id
  - A unique user identifier will be stored, in order to be called upon in other tables, for a specific user.
  - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
- User\_password
  - Passwords will be stored in the users table to group the user information with their id and email.
  - VARCHAR was used to set a maximum number of characters allowed.
- User\_email
  - Variable that stores the user's email in order for the user to create their account.
  - VARCHAR was used to set a maximum number of characters allowed.

- **Ingredients**

- Ingredient\_id
  - A unique identifier will be stored, in order to be called upon in other tables, for a specific ingredient.
  - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
- Ingredient\_name
  - To identify what ingredient is being used for the recipe.
  - VARCHAR was used to set a maximum number of characters allowed.

- **Recipe**

- Recipe\_id
  - A unique identifier will be stored, in order to be called upon in other tables, for a specific recipe.
  - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
- Recipe\_instructions
  - Instructions for each recipe are stored in the recipe table.
  - VARCHAR was used to set a maximum number of characters allowed.

- Recipe\_Public(Boolean)
  - To see if the recipe is either public or private we added this into our recipe table.
  - BOOLEAN was used to see if it is True or False.
- User\_id \*\*
  - We need to grab the specific user creating the recipe from the user table. We do this by referencing that table. We decided to keep the variable name the same (although, it could have been changed—as long as we referenced the original variable correctly)
  - INT NOT NULL REFERENCING user(user\_id)-- We are referencing the user table, at the user\_id.
- Ingredient\_id\*\*
  - We need to grab the specific ingredient(s) creating the recipe from the ingredients table. We do this by referencing that table. We decided to keep the variable name the same (although, it could have been changed—as long as we referenced the original variable correctly)
  - INT NOT NULL REFERENCING ingredients(ingredient\_id)-- We are referencing the ingredients table, at the ingredient\_id.
- **Grocery List**
  - Grocery\_list\_id
    - A unique identifier will be stored, in order to be called upon in other tables, for a specific grocery list.
    - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
  - Ingredient\_id\*\*
    - We need to grab the specific ingredient(s) for the grocery list from the ingredients table. We do this by referencing that table. We decided to keep the variable name the same (although, it could have been changed—as long as we referenced the original variable correctly)
    - INT NOT NULL REFERENCING ingredients(ingredient\_id)-- We are referencing the ingredients table, at the ingredient\_id.
- **Occasion**
  - Occasion\_id

- A unique identifier will be stored, in order to be called upon in other tables, for a specific occasion.
  - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
- recipe\_id\*\*
  - We need to grab the specific recipe for the specific occasion from the recipe table. We do this by referencing that table. We decided to keep the variable name the same (although, it could have been changed—as long as we referenced the original variable correctly)
  - INT NOT NULL REFERENCING recipe(recipe\_id)-- We are referencing the recipe table, at the recipe\_id.
- Recipe\_share
  - View\_User\_id
    - A unique identifier will be stored, in order to be called upon in other tables, for a specific shared recipe.
    - SERIAL PRIMARY KEY used for automatic integer storing, giving unique value.
  - recipe\_id\*\*
    - INT NOT NULL REFERENCING recipe(recipe\_id)-- We are referencing the recipe table, at the recipe\_id.
  - Recipe\_Public(Boolean)\*\*
    - INT NOT NULL REFERENCING recipe(recipe\_public)-- We are referencing the recipe table, at the recipe\_public.

## Tables

```
CREATE TABLE users(  
  user_id SERIAL PRIMARY KEY,  
  user_password VARCHAR(50),  
  user_email VARCHAR(50)  
  user_name VARCHAR(25)  
);
```

```
CREATE TABLE ingredients(  
  ingredient_id SERIAL PRIMARY KEY,  
  ingredient_name VARCHAR(50)  
);
```

```
CREATE TABLE recipe(  
  recipe_id SERIAL PRIMARY KEY,  
  recipe_instructions VARCHAR(1000),  
  recipe_public BOOLEAN,  
  user_id INT NOT NULL REFERENCES users(user_id),  
  ingredient_id INT NOT NULL REFERENCES ingredients(ingredient_id)  
);
```

```
CREATE TABLE grocery_list(  
  grocery_list_id SERIAL PRIMARY KEY,  
  ingredient_id INT NOT NULL REFERENCES ingredients(ingredient_id)  
);
```

```
CREATE TABLE occasion(  
  occasion_id SERIAL PRIMARY KEY,  
  recipe_id INT NOT NULL REFERENCES recipe(recipe_id)  
);
```

```
CREATE TABLE recipe_share(  
  view_user_id SERIAL PRIMARY KEY,  
  recipe_id INT NOT NULL REFERENCES recipe(recipe_id)  
);
```