

## DATA MODELING

### **Part 1:**

#### **Brainstorming:**

- First name
- Last name
- Email
- Password
- Phone
- Username
- Grocery list
- Recipes/Occasion recipes
- Ingredients
- Instructions
- Occasions

#### **Table Ideas:**

users: holds info about user

user\_id  
first\_name  
last\_name  
password  
username

contact\_info: holds contact info for user

contact\_id  
user\_id  
email  
phone

grocery\_list: holds grocery list for user

grocery\_id  
user\_id  
ingredients

occasions: holds occasions created by user

occasion\_id  
user\_id  
occasion\_name

ingredients: holds ingredients for a recipe

ingredient\_id  
recipe\_id  
text  
grocery\_id

instructions: holds instructions for a recipe

instruction\_id  
recipe\_id  
text

recipes: holds recipe created by user

recipe\_id  
user\_id

occasion\_recipes: holds recipes for an occasion

occasion\_recipe\_id  
occasion\_id  
user\_id  
recipe\_id

## **Relationships:**

### One-to-one:

users = contact info: Certain contact info is specific to a certain user.

users = grocery\_list: Certain grocery list is specific to a certain user.

recipes = ingredients: A recipe can only have one list of ingredients and a list of ingredients only belongs to one recipe.

recipes = instructions: A recipe can only have one set of instructions and a set of instructions can only belong to one recipe.

### One-to-many:

users => recipes: Users can have many recipes, but a recipe can only belong to one user.

users => occasions: Users can create many occasions but an occasion can only be created by one person.

occasions => occasion\_recipes:

### Many-to-many:

## Columns:

### users

first\_name & last\_name- These are being stored for the user's profile. I used VARCHAR as the data type because it will be a string.

username & password - password and username are being stored to allow the user to login. I used VARCHAR for these columns because they will also be strings.

### contact\_info

user\_id - This is being stored because contact\_info has a one-to-one relationship with the users table. It will be an integer because it is the id of another table.

email & phone - These are being stored so a users profile can display their contact info to other users. I used VARCHAR as the data type for both of these because they will be strings.

### grocery\_list

user\_id - This is included because grocery\_list has a one-to-one relationship with users. It will be an integer because it is the id of another table.

ingredients - This is stored because it is the grocery list for a certain user. It will be a VARCHAR because the list will be text.

### occasions

user\_id - This is stored because occasions has a one-to-many relationship with users. It will be an integer.

occasion\_name - Stored so users know the name of the occasion they created. It will be a VARCHAR

### ingredients

recipe\_id - Ingredients table has a one-to-one relationship with the recipe\_table. It is an integer.

ingredients - This will be the list of ingredients for a certain recipe. It will be a VARCHAR.

grocery\_id - This is stored because ingredients can be added to the grocery list table. It is an integer.

## instructions

recipe\_id - Instructions table has a one-to-one relationship with the recipe\_table. It is an integer.

instructions - Set of instructions for a certain recipe.

## recipes

user\_id - Recipes table has a one-to-many relationship with users. This will be an integer.

## occasion\_recipes

occasion\_id - This is stored because the occasion recipe will be assigned to a certain occasion. It is an integer.

user\_id - Stored because the recipe will be created by a certain user. This is an integer.

recipe\_id - occasion\_recipes has a one-to-one relationship with the recipes table. This is also an integer.

## SQL Code:

users:

```
CREATE TABLE users (  
  user_id SERIAL PRIMARY KEY,  
  first_name VARCHAR(255),  
  last_name VARCHAR(255),  
  password VARCHAR(255),  
  username VARCHAR(255)  
);
```

contact\_info:

```
CREATE TABLE contact_info (  
  contact_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  email VARCHAR(255),  
  phone VARCHAR(255)  
);
```

grocery\_list:

```
CREATE TABLE grocery_list (  
  grocery_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  ingredients VARCHAR  
);
```

occasions:

```
CREATE TABLE occasions (  
  occasion_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  occasion_name VARCHAR(255)  
);
```

recipes:

```
CREATE TABLE recipes (  
  recipes_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id)  
);
```

instructions:

```
CREATE TABLE instructions (  
  instruction_id SERIAL PRIMARY KEY,  
  recipes_id INTEGER REFERENCES recipes(recipes_id),  
  instructions VARCHAR  
);
```

ingredients:

```
CREATE TABLE ingredients (  
  ingredient_id SERIAL PRIMARY KEY,  
  recipes_id INTEGER REFERENCES recipes(recipes_id),  
  grocery_id INTEGER REFERENCES grocery_list(grocery_id),  
  ingredients VARCHAR  
);
```

occasion\_recipes:

```
CREATE TABLE occasion_recipes (  
  occasion_recipe_id SERIAL PRIMARY KEY,  
  occasion_id INTEGER REFERENCES occasions(occasion_id),  
  user_id INTEGER REFERENCES users(user_id),  
  recipes_id INTEGER REFERENCES recipes(recipes_id)  
);
```