

## BRAINSTORMING

User\_id  
Username  
Email  
Password

Recipes:  
Recipe\_id  
Recipe\_name  
Instructions  
Ingredients

Public and private feature  
Boolean: true/false

Share function  
User\_id  
Recipe\_id

Occasions  
Occasion\_id  
User\_id  
Recipe\_id

Grocery lists

### **TABLES:** Recipes:

Recipe\_id  
Recipe\_name  
Instructions  
Ingredients

User: This table will handle user information.

User\_id  
Username  
Email  
Password

Recipes:  
Recipe\_id  
Recipe\_name  
Instructions  
Ingredient\_id

User\_id

Grocery List  
Grocery\_id  
Ingredient\_id

Occasion:  
occasion\_id  
Ingredient\_id  
Recipe\_id  
user\_id

View:  
View\_id  
User\_id  
Recipe\_id

## **RELATIONSHIPS:**

### **“One-to-one”**

**Grocery List: user is creating a grocery list. One user and one grocery list.**

### **“one-to-many”**

**Occasions: user to occasion. Can have multiple recipes in one occasion from one user.**

### **“many-to-many”**

**View to recipe: There's multiple users viewing multiple recipes.**

**Grocery list to ingredients: multiple grocery lists can contain multiple ingredients**

**Ingredients to recipe: multiple ingredients can go into multiple recipes.**

## **COLUMNS:**

### **Users:**

**User\_id: tracks our users to connect to views and recipes, integer type to auto-increment for new users.**

**-email: a limited varchar for users to input their contact information to link to their account**

**-username: a unique username with limited varchar for users to use on sign-in**

**-user password: a limited varchar for users to use on sign-in**

#### **RECIPES:**

**Recipe id: identifies each recipe to include in occasion lists and track views**

**Recipe name: identifies it by the name of the recipe. Used with a limited varchar for users to keep track of recipes.**

**public/private: this boolean value will be true or public, false for private**

**Instructions: this text variable includes the set of instructions connected to the current recipe.**

**Ingredient-id: this text variable holds the foreign keys for each ingredient included in the recipe.**

**User id: this foreign key connects to the user table to connect the recipe to its author .**

#### **INGREDIENTS:**

**Ingredient id: identifies each ingredient to include in ingredient lists and occasion lists**

**Integer type to auto increment for each new recipe**

**Ingredient name: this varchar includes the ingredients name to be displayed in recipes and grocery lists.**

#### **GROCERY**

**Grocery id: identifies each grocery list to relate to its respective user.**

**Ingredients id: this varchar contains a list of each ingredient id added to the list**

**User id: this foreign key connects to the user table and corresponding user.**

#### **OCCASION:**

**Occasion id: identifies each occasion created by the user**

**Recipe id: this varchar contains a list of each recipe id added to the list**

**User-id: this foreign key connects to the user that owns the list**

#### **VIEWS:**

**View id: this integer key tracks how many views a recipe has.**

**User id: foreign key connects to corresponding user.**

**Recipe id: this connects to the recipe that's being viewed.**

#### **SQL CODE:**

```
CREATE TABLE users(  
  user_id SERIAL PRIMARY KEY,  
  email varchar(40),  
  username varchar(40),  
  password varchar(40)  
);
```

```
CREATE TABLE recipes (  
  recipe_id SERIAL PRIMARY KEY,  
  recipe_name varchar(40),  
  public boolean,
```

```
instructions text,  
ingredient_id varchar(40)  
);
```

```
CREATE TABLE ingredients (  
ingredient_id SERIAL PRIMARY KEY,  
ingredient_name varchar(40)  
);
```

```
CREATE TABLE groceryList (  
grocery_id SERIAL PRIMARY KEY,  
ingredient_id varchar(40)  
);
```

```
create table occasion (  
occasion_id serial primary key,  
recipe_id varchar(40),  
user_id integer  
);
```

```
create table views (  
view_id serial primary key,  
user_id integer,  
recipe_id integer  
);
```

```
insert into recipes  
-- (recipe_name, public, instructions)  
-- values  
-- ('tacos', true, 'Assemble ingredients on tortilla')
```

```
insert into users (email, username, password)  
-- values  
-- ('schmit@lmhot.email', 'schmitty', 'winstonsux')
```