

Brainstorming

email, password, first name, last name, ingredients, instructions, public, private, occasions, reviews

Table Ideas

Users

Recipes

Grocery List

Occasions

Reviews

Relationships

One to one

Grocery List to User: because there will only be one grocery list per user.

One to many

User to Recipes, because there can be many recipes for one user but not many users for one recipe

Many to Many

Occasions to user and recipe: Occasions links a user to a recipe with the occasion being the link?

Reviews to users and recipes: the review brings together a user and the recipe they're reviewing.

Columns

USERS - user_id(to have a specific ID for the record) email(for login and contacting user), password(the way they get into the site and to confirm identity), first name, last name(names to personalize the profile)

RECIPES - recipe_id(to have a specific ID for the record) , user_id(to show which user's recipe it is), ingredients(to show what's in the recipe), instructions(to show how to make the item), private (it'll be a boolean to show if it's just for the one user or not)

GROCERY LIST - list_id(to have a specific ID for the record) , user_id(show who's list), ingredients(show what they need)

OCCASIONS - occasions_id(to have a specific ID for the record) , user_id(show who's occasion we're talking about), occasion_name(which occasion we're talking about), recipe_id(which recipe the user is connecting to the occasion)

REVIEWS - review_id(to have a specific ID for the record) , user_id(to correlate it with a specific user), recipe_id(to show which recipe is being referenced), review_content(the actual review itself.

```
CREATE TABLE users(  
  user_id SERIAL PRIMARY KEY,  
  email VARCHAR(100) NOT NULL,  
  password VARCHAR(100) NOT NULL,  
  first_name VARCHAR(40) NOT NULL,  
  last_name VARCHAR(40) NOT NULL  
)
```

```
CREATE TABLE recipes(  
  recipe_id SERIAL PRIMARY KEY,  
  ingredients VARCHAR(3000) NOT NULL,  
  instructions VARCHAR(6000) NOT NULL,  
  private BOOLEAN NOT NULL,  
  creator INTEGER NOT NULL REFERENCES users(user_id)  
)
```

```
CREATE TABLE grocery_list(  
  list_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  ingredients VARCHAR(3000)  
)
```

```
CREATE TABLE occasions(  
  occasion_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  occasion_name VARCHAR(50),  
  recipe_id INTEGER REFERENCES recipes(recipe_id)  
)
```

```
CREATE TABLE reviews(  
  review_id SERIAL PRIMARY KEY,  
  user_id INTEGER REFERENCES users(user_id),  
  review VARCHAR(3000)  
)
```

```
INSERT INTO users(email, password, first_name, last_name)  
VALUES ('realemail', 'benni', 'michael', 'dwyer');  
INSERT INTO users(email, password, first_name, last_name)  
VALUES ('fakeemail', 'fakjepassword', 'nate', 'tako');  
INSERT INTO users(email, password, first_name, last_name)  
VALUES ('asdfa', 'fakjepasdfasdfassword', 'naasdfaste', 'taasdfasko');  
SELECT * FROM users
```

```
INSERT INTO recipes(ingredients, instructions, private, creator)
VALUES ('carrot', 'put the carrot in the door', False, 1);
INSERT INTO recipes(ingredients, instructions, private, creator)
VALUES ('gun', 'take the gun apart', False, 1);
SELECT * FROM recipes
```