

Data Modeling with SQL

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**

Brainstorming/Data Needed

- User_id
- Username
- Password
- Email address
- First and last name
- Location
- Gender
- Bio
- Favorite cuisines
- Favorite chefs
- Users followed
 - Who the user is following
 - Who is following the user
- Recipes
 - Public recipes
 - Private recipes accessible to permitted followers
 - Recipe title
 - Time posted
- Grocery lists
- Occasions
 - Occasions title

Table Ideas

- **Users:** Each row will hold data on each individual user.
 - User_ID
 - Username
 - Password
 - Email_address
 - First and last name
 - Location
 - Gender
 - Bio
 - Birthday
 - Favorite cuisines
 - Favorite chefs
- **Recipes:** Each row will hold data on different recipes, the level of difficulty in their making, and whether they're accessible to the public or only to followers who the author authorized to view.
 - Recipe_ID
 - Recipe_title
 - Author
 - post_time
 - Difficulty_level
 - Recipe_content
 - Public recipes [toggle/boolean]
 - Private recipes [toggle/boolean]
 - Comments
 - Instructions
 - Ingredients
- **Groceries:** Each row will include the name of the product/ingredient which goes into making the recipe.
 - Grocery_list_ID
 - Ingredient_name
 - Category
 - Quantity
- **Occasions:** Each row will hold a specific title of the occasion which the author comes up with, whether it is indoors or outdoors, and the general category of the occasion.
 - Occasion_ID
 - Occasion_title
 - Outdoor/Indoor
 - Occasion_type [family, social, corporate etc.]
- **Follows:** Holds the ID of who each user follows.
 - Follower
 - Person being followed

Relationships

- One-to-one
 - Recipes ⇒ Groceries
- One-to-many
 - Users ⇒ Recipes
 - Users ⇒ Occasions
- Many-to-many
 - Recipes ⇒ Occasions
 - Follower ⇒ Following

Seed

```
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    username VARCHAR(50) UNIQUE,  
    password VARCHAR (500),  
    email VARCHAR(50) UNIQUE,  
    user_first_name VARCHAR(50),  
    user_last_name VARCHAR(50)  
    location VARCHAR(50),  
    gender VARCHAR(10),  
    bio VARCHAR(1000),  
    user_phone INT,  
    birthday TIMESTAMP,  
    profile_pic TEXT  
);
```

```
CREATE TABLE recipes (  
    recipe_id SERIAL PRIMARY KEY,  
    recipe_title VARCHAR(20),  
    author VARCHAR(30),  
    post_time TIMESTAMP,  
    difficulty_level INT,
```

```
    recipe_content TEXT,  
    is_public_recipe BOOLEAN,  
    is_private_recipe BOOLEAN,  
    comments TEXT,  
    instructions TEXT,  
    Ingredient_name VARCHAR(50)  
    user_id INT NOT NULL REFERENCES users(user_id)  
);
```

```
CREATE TABLE groceries (  
    grocery_list_id SERIAL PRIMARY KEY,  
    ingredient_name VARCHAR(50) NOT NULL REFERENCES recipes(ingredients_name),  
    ingredient_category VARCHAR(50),  
    quantity_needed INT  
);
```

```
CREATE TABLE occasions (  
    occasion_id SERIAL PRIMARY KEY,  
    user_id INT NOT NULL REFERENCES users(user_id),  
    Recipe_id INT NOT NULL REFERENCES recipes(recipe_id),  
    occasion_title VARCHAR(50),  
    is_outdoor BOOLEAN,  
    is_indoor BOOLEAN  
    occasion_type VARCHAR(50)  
);
```

```
CREATE TABLE following (  
    follow_id SERIAL PRIMARY KEY,  
    follower_id INT NOT NULL REFERENCES users(user_id),  
    following_id INT NOT NULL UNIQUE REFERENCES users(user_id)  
);
```