

Brainstorming:

Users

User login information (email, password)

Recipes

Recipe creator

Recipe status (public, private)

Recipe ingredients

Recipe instructions

User grocery lists

User occasions

Recipes for occasion

User > Log In {email, password} > Browse Public Recipes {recipe name, recipe creator, recipe status, ingredients, instructions}

User > Log In {email, password} > Grocery List {ingredients}

User > Log In {email, password} > Occasions { recipes}

Table Ideas:

users		
Users: Table to store information about the app users, with each row being an individual user.		
user_id (Primary key)	email	password

recipes					
Recipes: Table to store information about the recipes in the app. Each row will be an individual recipe.					
Recipe_id (Primary key)	name	User_id (Foreign key from users)	status	ingredients	instructions

grocery_lists		
Grocery_lists: Table to store information about the users' grocery lists. Each row will be an individual grocery list.		
List_id (Primary key)	ingredients	User_id (Foreign key from users)

occasions			
Occasions: Table to store information about the users' added occasions, with each row being an individual occasion.			
Occasion_id (Primary key)	name	description	User_id (foreign key from users)

recipe_occasions		
Recipe_occasions: Associate table to connect recipes and occasions.		
recipe_occasion_id (Primary key)	Occasion_id (foreign key from occasions)	Recipe_id (foreign key from recipes)

recipe_grocery_lists		
Recipe_grocery_lists: Associate table to connect recipes and grocery lists.		
recipe_grocery_list_id (Primary key)	Recipe_id (foreign key from recipes)	Grocery_list_id (foreign key from recipes)

Relationships:

Many to One:

Users to recipes: a user can have many recipes and a recipe belongs to one user.

Users to occasions: a user can define many occasions and an occasion belongs to one user.

One to One:

Users to grocery_lists: A user has one grocery list and a grocery list belongs to one user.

Many to Many:

Recipes to occasions: A recipe can belong to many occasions and an occasion can have many recipes.

Recipes to grocery_lists: A grocery list can have ingredients from many recipes and the ingredients from the recipe can belong to many grocery lists.

Columns:

users:

- user_id: INTEGER and PRIMARY KEY: to assign an integer value to every user for identification
- email: VARCHAR: to store users' email addresses, which are strings of limited characters
- Password: VARCHAR: to store users' passwords, which are strings of limited characters

recipes:

- recipe_id: INTEGER and PRIMARY KEY: to assign an integer value to every recipe for identification
- title: VARCHAR: to store recipe names, which are strings of limited characters. This could be useful in search functions
- public_status: BOOLEAN: to designate the recipe as public (True) or private (false)
- ingredients: VARCHAR: to store recipe ingredients, which are strings of limited characters
- instructions: VARCHAR: to store recipe instructions, which are strings of limited characters
- user_id: INTEGER and FOREIGN KEY: to connect the recipe with the user that created them

grocery_lists:

- list_id: INTEGER and PRIMARY KEY: to assign an integer value to every grocery list for identification
- items: VARCHAR: to store the grocery list items, which are strings of limited characters
- user_id: INTEGER and FOREIGN KEY: to connect the grocery list with the user that created it

occasions:

- occasion_id: INTEGER and PRIMARY KEY: to assign an integer value to every occasion for identification
- occasion_name: VARCHAR: to store occasion names (mostly for the users' benefit). These are strings of limited characters
- occasion_description: VARCHAR: to store occasion descriptions (mostly for the users' benefit). These are strings of limited characters

- user_id: INTEGER and FOREIGN KEY: to connect the occasion to the user that created it

recipes_occasions:

- rec_occ_id: INTEGER and PRIMARY KEY: to assign an integer value to every occasion-recipe pair for identification
- occasion_id: INTEGER and FOREIGN KEY: to connect the occasion with the recipes assigned to it
- recipe_id: INTEGER and FOREIGN KEY: to connect the recipe with the occasions it belongs to

recipes_grocery_lists:

- rec_list_id: INTEGER and PRIMARY KEY: to assign an integer value to every recipe-grocery_list pair for identification
- grocery_list_id: INTEGER and FOREIGN KEY: to connect the grocery list to the recipe ingredients that make up the list
- recipe_id: INTEGER and FOREIGN KEY: to connect the recipe ingredients to the grocery lists that include them

SQL Statements:

```
CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
    email VARCHAR(60),
    password VARCHAR(60)
);
```

```
CREATE TABLE recipes (
    recipe_id SERIAL PRIMARY KEY,
    title VARCHAR(100),
    public_status BOOLEAN,
    ingredients VARCHAR(500),
    instructions VARCHAR(2000),
    user_id INTEGER NOT NULL REFERENCES users(user_id)
);
```

```
CREATE TABLE grocery_lists (
    list_id SERIAL PRIMARY KEY,
    items varchar(1000),
    user_id INTEGER NOT NULL REFERENCES users(user_id)
);
```

```
CREATE TABLE occasions (  
    occasion_id SERIAL PRIMARY KEY,  
    occasion_name VARCHAR(100),  
    occasion_description VARCHAR(300),  
    user_id INTEGER NOT NULL REFERENCES users(user_id)  
);
```

```
CREATE TABLE recipes_occasions (  
    rec_occ_id SERIAL PRIMARY KEY,  
    occasion_id INTEGER NOT NULL REFERENCES occasions(occasion_id),  
    recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)  
);
```

```
CREATE TABLE recipes_grocery_lists (  
    rec_list_id SERIAL PRIMARY KEY,  
    grocery_list_id INTEGER NOT NULL REFERENCES grocery_lists(list_id),  
    recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)  
);
```