

Brainstorming

- user id
- username
- email
- user password
- recipe id
- recipe name
- public/private
- ingredients list
- instructions list
- grocery list
- occasions
 - recipes
- share function

Table Ideas

- user table
 - stores their basic information, grocery lists, occasions, list
- recipe table
 - stores id, ingredients, instructions, and public/private
- ingredient table
 - ingredient id, name
- grocery list
 - id, ingredient ids
- occasion
 - id, user, recipe
- view
 - view id, user id, recipe id

Relationships

- one-to-one
 - user to grocery list
 - each user has only one grocery list that is unique to them
- one-to-many
 - user to recipe
 - each user can create multiple recipes
 - user to occasions
 - each user can create multiple occasions
- many-to-many
 - view to user
 - each recipe can be viewed by multiple users
 - view to recipe
 - each recipe can be viewed by multiple users
 - recipe to ingredients
 - multiple recipe can contain multiple ingredients
 - grocery list to ingredients
 - multiple grocery lists can include many ingredients

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
- user_name: 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 occasions lists and track views, integer type to auto-increment for each new recipe
 - recipe_name: stores the recipe of the recipe to display to users
 - public/private: this boolean value will be true for 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 ingredient 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 that owns the list
- 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 the total views occurring by all users on all recipes
 - user_id: this foreign key will connect to the user who is viewing the corresponding recipe
 - recipe_id: this foreign key will connect to the recipe that is being viewed

SQL code

```

create table users (
    user_id serial primary key,
    email varchar(40),
    user_name varchar(40),
    user_password varchar(40)
);

create table recipes (
    recipe_id serial primary key,
    recipe_name varchar(40),
    public boolean,
    instructions text,
    ingredient_id varchar(40),
    user_id integer
);

create table ingredients (
    ingredient_id serial primary key,
    ingredient_name varchar(40)
);

```

```
create table groceryList (  
    grocery_id serial primary key,  
    ingredients_id varchar(40),  
    user_id integer  
);
```

```
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 tortillas'),  
('Buttered Toast', false, 'Toast bread. Spread butter on toasted bread.')
```

```
insert into occasion (recipe_id)  
values ('1,2')
```