Brainstorming

- users can sign into the app with their email and password
    - user_id
    - user_email
    - user_password
- users can create recipes with ingredients and instructions
    - recipe_id
    - ingredient_id
- recipes can be marked as public or private
    - is_public
- users can view other people's recipes
    - user_id
    - recipe_id
- ingredients from recipes can be added to user's grocery lists
    - grocery_list_id
- users can create their own occasions and assign recipes to occasions
    - occasion_id
    - recipe_id

Table Ideas

- user table (holds all users and their info)
    - user_id
    - user_email
    - user_password
- recipes table (holds all recipes and their ingredients and instructions)
    - recipe_id
    - ingredient_id
    - instructions_body
    - is_public
- grocery_list table (holds grocery list)
    - grocery_list_id
    - ingredient_id
- occasion table (holds list of occasions and their details)
    - occasion_id
    - recipe_id
    - user_id
    - occasion_date
- ingredients table (holds master list of all ingredients)
    - ingredient_id

Relationships

- one-to-one

    - each user has only one user_id, one user_email, and one user_password. Each of those values belong to only one user.
- one-to-many
    - user table & grocery_list table (one user can have many grocery lists, and each grocery list has only one user)
- many-to-many
    - ingredients table & recipes table (one recipe uses many ingredients, and one ingredient can be used in many recipes.)

Columns

- users table
    - user_id
        - to keep track of users
        - serial – auto-incrementing
    - user_email
        - user contact info and login
        - string – able to store email
    - user_password
        - user uses password to login
        - string – able to create password using string

```
CREATE TABLE users(
        user_id SERIAL PRIMARY KEY,
  user_email VARCHAR(40) NOT NULL,
  user_password VARCHAR(40) NOT NULL
);
```

- recipe table
    - recipe_id
        - to keep track of recipes
        - serial – auto-incrementing
    - is_public
        - to keep track of if a recipe is public or not
        - Boolean – true = public, false = private
    - user_id
        - to tie a recipe to the user who created it
        - integer – same as in user table
    - instructions_body
        - where the actual recipe directions will be stored
        - text – allows paragraphs to be stored
    - ingredient_id
        - to tie the ingredients used to the recipe
        - integer – same as in ingredient table

```
CREATE TABLE recipe(
```

```
        recipe_id SERIAL PRIMARY KEY,
  is_public BOOLEAN NOT NULL,
  user_id INTEGER NOT NULL REFERENCES users(user_id),
  instructions_body TEXT NOT NULL,
  ingredient_id INTEGER REFERENCES ingredient(ingredient_id)
);
```

- occasion table
    - occasion_id
        - to keep track of occasions
        - serial – auto-incrementing
    - occasion_name
        - name of occasion
        - string
    - recipe_id
        - to tie the recipe to the occasion it is being made for
        - integer – same as in recipe table
    - user_id
        - to tie the occasion to the user having the occasion
        - integer – same as in user table
    - occasion_date
        - the date the occasion will be taking place
        - timestamp – easily store date and time this way

```
CREATE TABLE occasion(
  occasion_id SERIAL PRIMARY KEY,
  occasion_name VARCHAR(80) NOT NULL,
  recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
  user_id INTEGER NOT NULL REFERENCES users(user_id),
        occasion_date TIMESTAMP NOT NULL
);
```

- grocery_list table
    - grocery_list_id
        - to keep track of different grocery lists
        - serial – auto-incrementing
    - ingredient_id
        - to tie which ingredients are needed to the grocery list
        - integer – same as in ingredient table
    - user_id
        - to tie the list to the user who created it
        - integer – same as in user table

```
CREATE TABLE grocery_list(
  grocery_list_id SERIAL PRIMARY KEY,
  ingredient_id INTEGER REFERENCES ingredient(ingredient_id),
```

```
  user_id INTEGER NOT NULL REFERENCES users(user_id)
);
```

- ingredient table
    - ingredient_id
        - to keep track of all ingredients
        - serial – auto-incrementing
    - ingredient_name
        - name of actual ingredient
        - string – easy to store names of ingredients this way

```
CREATE TABLE ingredient(
  ingredient_id SERIAL PRIMARY KEY,
  ingredient_name VARCHAR(30) NOT NULL
);
```