**Brainstorming**

User- username, password, email, name, user-id, friends
Recipe- user-id, public (y/n), ingredients, instructions, recipe-id, notes
List- items, user id, recipe ingredients, list-id, notes, recipe-id(if items are from recipe)
Occasions- user-id, occasion-id, recipes(by id), notes

**Tables**

User

(holds user data)

- Username
- Password
- Email
- First names
- Last name
- User id

Item

(all items you can buy at a store)

- Item id
- Item name

Recipe

(Holds info for recipes)

- Recipe id
- Recipe name
- Ingredients (from item table)
- Instructions
- Notes
- User id (from user table)
- Occasions used in (from occasion table)

List

(holds items for shopping, as well as recipe ingredients)

- List id
- User id (from user table)
- List contents (from item table)
- Notes
- Recipes included (from recipe table(?))

Occasion

(holds info for occasions, including what recipes will be used)

- Occasion id
- Recipes (from recipe table)
- User id (from user table)
- Notes

**Relationship**

One to one

<u>One to many</u>

User => recipe (A recipe can be made by only one user, even if shared among many)

User => list (A list can be made by only one user, even if shared among many)

User => occasion (An occasion can be made by only one user, even is shared among many)

<u>Many to many</u>

Item <=> recipe (An item can be on multiple recipes, and a recipe has multiple items)

Occasion <=> recipe (An occasion can use multiple recipes, and a recipe can appear during multiple occasions)

Item <=> list (An item can be on multiple lists, and a list can have multiple items)

List <=> recipe (A list can contain items from multiple recipes, and a recipes items may appear on several lists)

**Columns**

<u>user</u>

- user_id int [pk, increment]: So that it will increment with each new user
- username VARCHAR(50) [not null]: Required field. So the user can choose a username. Character limit of 50.
- password VARCHAR(50) [not null] Required field. Character limit of 50.
- first_name VARCHAR(50) [not null] Required field. Character limit of 50. So that people on site can search using first and last names.
- last_name VARCHAR(50) [not null] Required field. Character limit of 50. So that people on site can search using first and last names.
- user_email VARCHAR(50) [not null] Required field. Character limit of 50. So that the app can send promotional emails.

<u>item</u>

- item_id int [pk, increment] Item list will be controlled by the developers. Item Id will increment automatically.
- item_name VARCHAR(50) [not null] So users can search items by name.

<u>recipe</u>

- recipe_id int [pk, increment] Will increment with each new recipe
- recipe_name VARCHAR(50) Limit of 50 characters. Did not make it required so that a user can make a recipe even if they aren’t sure of all details.
- recipe_ingredients VARCHAR(2000) List of ingredients used for the recipe. Limit of 2000 characters. Did not make it required so that a user can make a recipe even if they aren’t sure of all details.
- recipe_instructions VARCHAR(10000) All cooking instructions for the recipe. Limit of 10000 characters. Did not make it required so that a user can make a recipe even if they aren’t sure of all details.

- recipe_notes VARCHAR(10000) Any additional notes that the creator would like to add. Limit of 10000 characters. Not required.
- creator_id INT [not null] Id if the user that made the recipe.
- occaions_used INT Any occasions that the recipe is used in. Listed as occasion’s id.
- private BOOLEAN [not null] Setting recipe as private ‘true’ or ‘false’

list

- list_id int [pk, increment] Will increment with each new list made.
- list_contents VARCHAR(10000) Items that the user needs to purchase. Limit of 10000 characters.
- list_notes VARCHAR(10000) Any additional notes the user wants about the list. Limit of 1000 characters.
- recipes_used INT Id of any recipes whose ingredients are on the list.
- creator_id INT [not null] id of the user who made the list

occasion

- occasion_id int [pk, increment] Will increment with each new occasion made
- occasion_name VARCHAR(100) Limit of 100 characters. Not required if the user wants to make an occasion but is unsure of the details.
- occasion_notes VARCHAR(10000) Any additional notes that the user would like about the occasion. Limit 10000 characters.
- recipes_used INT id of any recipes used for this occasion.
- creator_id INT [not null] id of the user who made the occasion.

**Create tables**

```
CREATE Table "user" (
  "user_id" SERIAL PRIMARY KEY,
  "username" VARCHAR(50) NOT NULL,
  "password" VARCHAR(50) NOT NULL,
  "first_name" VARCHAR(50) NOT NULL,
  "last_name" VARCHAR(50) NOT NULL,
  "user_email" VARCHAR(50) NOT NULL
);

CREATE Table "item" (
  "item_id" SERIAL PRIMARY KEY,
  "item_name" VARCHAR(50) NOT NULL
);

CREATE Table "recipe" (
  "recipe_id" SERIAL PRIMARY KEY,
  "recipe_name" VARCHAR(50),
  "recipe_ingredients" VARCHAR(2000),
  "recipe_instructions" VARCHAR(10000),
  "recipe_notes" VARCHAR(10000),
  "creator_id" INT NOT NULL,
  "occaions_used" INT,
  "private" BOOLEAN NOT NULL
);

CREATE Table "list" (
  "list_id" SERIAL PRIMARY KEY,
  "list_contents" VARCHAR(10000),
  "list_notes" VARCHAR(10000),
  "recipes_used" INT,
  "creator_id" INT NOT NULL
);

CREATE Table "occasion" (
  "occasion_id" SERIAL PRIMARY KEY,
  "occasion_name" VARCHAR(100),
  "occasion_notes" VARCHAR(10000),
  "recipes_used" INT,
  "creator_id" INT NOT NULL
)

-- INSERT INTO item(item_name)
-- VALUES ('carrot');

SELECT * FROM item
```