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**

Part 1: Conceptual Plain

Step 1 Brainstorming

How many tables you think you may need and what information will be provided. Steps one and two work together in the thought process. I have retyped and played with ideas for the tables. Creating their own occasion is a personal decision. Not needed but can be implied by the name of the recipe.

Step 2

User table (one to one)

* user email
* user\_id
* password

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

Email VARCHAR(50),

name VARCHAR(50),

password VARCHAR(20)

);

CREATE TABLE recipe (

recipe\_id SERIAL PRIMARY KEY,

user\_id INTEGER,

recipe\_name VARCHAR(50),

ingredients\_id TEXT,

instructions\_id TEXT

);

User Recipe table

* ingredients\_id
* instructions\_id

CREATE TABLE user\_recipes(

user\_recipe\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id),

recipe\_id INTEGER NOT NULL REFERENCES recipe(recipe\_id)

);

Shared Recipe table (many to many)

* user\_recipe\_id
* user\_ingredients\_id
* user\_instructions\_id

CREATE TABLE shared\_recipe (

shared\_recipe\_id SERIAL PRIMARY KEY,

user\_recipe\_id INTEGER NOT NULL REFERENCES user\_recipes(user\_recipe\_id),

print\_date TIMESTAMP

);

Grocery List (many to many: Genres)

* user\_recipe\_id
* lists ingredients\_id

Making table for grocery is doesn’t really work or fit into the logic. This works.

SELECT \* FROM recipe

WHERE recipe\_id = 3;

Step 3 Relationships

**User table** is a one-one provides the user information. Password should be hashed if it can be view by others. Should also be set up with UNIQUE NOT NULL. Because you what it to be unique to user who is a person who owns their recipe on site.

**User Recipe** table is a one-to-many relationship where one being the user and many being the recipes that can be accessed. Keeping that in mind Boolean should be used when deciding if the user wants to publicly share recipes. Joins user and recipe creating a printings\_id. Can include a print\_date TIMESTAMP to show how many people printed recipes.

**Shared Recipe** table is many-to-many: Genre where there are many users and many recipes. This is a public and can be viewed by all. But is an association table that does not have any unique information. It shows that what recipe belongs to what user. Joined/ middle should be userRecipe\_id.

**Grocery List** table is many-to-many shows user and recipe where RecipeGrocery\_id should be the joined/middle.

Part 2: Table Plaining

Step 1 – DB Designer

Enter in my tables as from code above. Created the foreign keys to show relationships with tables and fields. As I went back and carefully read what relations are and coded my tables.

My original thoughts were changed when I started to code. Some of my tables changed and some of the relations change. I guess I had to see it in code before I could picture it as a flow chart.

Users table

Holds all the basic information used to engage in the use of the recipe app as an individual. The user table has one-to-many relationship with users recipe table. The user table creates the user id needed to make the app useful to the individual. Contains the user name, email and password. Used varchar on all because it is a string that can be accessed with defined number of characters. We all have common knowledge about the amount of information needed for those fields. Has a one-to-one relationship with recipe table so they can be identified as a user who has recipes of their own.

Recipe table

Holds the reason for visiting app for the recipes that will be stored and most likely shared. Most people like to brag and share good recipes. The table creates the recipe id bringing together the user and their recipe by name including the ingredients and instructions. The recipe table makes the recipes unique to the individual user. Making each recipe attached and viewable by id. The ingredients id and instructions id are created with text that is unlimited characters in a string. Both can be as long or short as you want. Has a one-to-many relationship with user table.

User recipe table

Creates the user recipe table id bring two unrelated tables together it is a many-to-many relationship. Many users and many recipes brought together. User id is a integer because each person has an individual id connected to recipes that is text with lots or little information.

Shared recipe table

Creates shared recipe id that can be viewed many times and tracked by time stamp. Has a one-to-one relationship with the user recipe id table. This where all recipes can be viewed by all users. User recipe id is an integer assigned a whole number has lots of information that has already been brought together. Time stamp is only there if you want to know how many times and when user recipes have been accessed.

Grocery list table

Creates grocery list id that gives the user a list body with all the items needed to make a recipe. Has a one-to-many relationship to recipe table. Which is kind of a lie because we are only asking for a part of the table.

Grocery list table was a somewhat of a good idea but the data base doesn’t think that way and believes that some data doesn’t exist or match. It makes more sense to look at getting a grocery list as a query of the database not a table. Visitors / app users will choose to select all recipes from the recipe table and ask for a specific recipe\_id they would like to try. It should be obvious that if you want to try a recipe you want all the information for that recipe that includes the name of recipe, who’s recipe it is, with the ingredients and instructions.