Brainstorming:

- User Info
- Profile
- Posts
- Recipes Saved
- Private Posts
- Grocery List

Table

- **Users**: This table allows a user to be identified and to sign in. This table also provides the user with the ability to add personal info to their profiles. (user_id SERIAL PRIMARY KEY, first_name VARCHAR, last_name VARCHAR, email VARCHAR, password VARCHAR).
- **Profile**: This table is the location where the user info can be stored, this includes posts, and specific recipes. (profile_id SERIAL PRIMARY KEY, user_id INT NOT NULL REFERENCES users(user_id), recipe_id INT NOT NULL REFERENCES recipes(recipe_id), list_id INT NOT NULL REFERENCES GroceryList(list_id), private_post_id INT NOT NULL REFERENCES PrivatePosts(private_post_id)).
- **Posts**: This table stores the data of all posts (regardless of recipe type) and allows a user to save that info onto their profile for future recipes and/or grocery lists. (post_id SERIAL PRIMARY KEY, user_id INT NOT NULL REFERENCES users(user_id), recipe_body TEXT, post_pics TEXT).
- **Recipes**: This table stores all the posts/recipes that a user would want to reference for later, can be found through the profile table. (recipe_id SERIAL PRIMARY KEY, user_id INT NOT NULL REFERENCES users(user_id), post_id INT NOT NULL REFERENCES posts(post_id)).
- **PrivatePosts**: This table contains private data that only a specific user can access, functions similarly to Recipes, but these are private ingredients and recipes that can only be accessed by the user who saved it. (private_post_id SERIAL PRIMARY KEY, user_id INT NOT NULL REFERENCES users(user_id), post_id INT NOT NULL REFERENCES posts(post_id)).
- **GroceryList**: This table functions similarly to the Recipes and PrivatePosts table, only difference is the content, the other tables save whole recipes, this table only saves ingredients. (list_id SERIAL PRIMARY KEY, user_id INT NOT NULL REFERENCES users(user_id), post_id INT NOT NULL REFERENCES posts(post_id), private_post_id INT NOT NULL REFERENCES PrivatePosts(private_post_id)).

Relationships:

ONE-TO MANY:

- User and Posts (One user can have many posts)
- Recipe and Posts (One recipe (save) can contain many posts)
- PrivatePosts and Posts (One PrivatePost (save) can contain many posts)

MANY-TO MANY:

- User to Profile to Recipes/PrivatePosts (A user can save many recipes and Private posts through their profile)
- Profile to Recipe/PrivatePosts to GroceryList (Many ingredients can be)

ONE-TO ONE:

- User and Profile (Each User and Profile can contain only one of each. There can be many users, but each have one profile and vice versa)

Tables and Datatype columns:

- Users: SERIAL PRIMARY KEYs are necessary for the tables. First/Last name, email, and password use VARCHAR because specific text needs to be added for the user.
- Profile: The profile will contain columns pertaining to its page content, mainly being recipes (public and private), and grocery lists. INT was used because only one primary key can exist within a table.
- Posts: The columns required in the posts section are the user_id and text options. The user_id is necessary so that individuals can identify where the recipe comes from. The text is necessary to write the recipes, even the picture requires the text datatype.
- Recipes: The columns in this table are meant to hold data saved from other sections of the app. These Integers are saved in the recipes folder to be accessed by users.
- Privateposts: Like the Recipe table, this folder holds the data from posts, the only exception is that posts made by a user can be submitted as a private post and saved here, or a recipe made by another user can be stored here privately.
- GroceryList: Like the Recipe and Private tables, this table holds specific ingredient data from the recipes and stores them on a specific user's profile.

  Overall, integers were the most commonly used Datatype in these columns, the main goal was to link one feature's data with another.

```
-- CREATE TABLE users(
--      user_id SERIAL PRIMARY KEY,
--   first_name VARCHAR,
--   last_name VARCHAR,
--          password VARCHAR,
--   email VARCHAR
-- );

-- CREATE TABLE profile(
--          profile_id SERIAL PRIMARY KEY,
--   user_id INT NOT NULL REFERENCES users(user_id),
--   recipe_id INT NOT NULL REFERENCES recipes(recipe_id),
--   private_post_id INT NOT NULL REFERENCES privateposts(private_post_id),
--   list_id INT NOT NULL REFERENCES grocerylist(list_id)
-- );

-- CREATE TABLE posts(
--          post_id SERIAL PRIMARY KEY,
--   user_id INT NOT NULL REFERENCES users(user_id),
--   recipe_body TEXT,
--   post_pics TEXT
-- );

-- CREATE TABLE recipes(
--          recipe_id SERIAL PRIMARY KEY,
--   user_id INT NOT NULL REFERENCES users(user_id),
--   post_id INT NOT NULL REFERENCES posts(post_id)
-- );

-- CREATE TABLE privateposts(
--        private_post_id SERIAL PRIMARY KEY,
--   user_id INT NOT NULL REFERENCES users(user_id),
--   post_id INT NOT NULL REFERENCES posts(post_id)
-- );

-- CREATE TABLE grocerylist(
--        list_id SERIAL PRIMARY KEY,
--   user_id INT NOT NULL REFERENCES users(user_id),
--   post_id INT NOT NULL REFERENCES posts(post_id),
--   private_post_id INT NOT NULL REFERENCES privateposts(private_post_id)
-- );
```