RECIPE CREATING/SHARING AND GROCERY LIST APP

Brainstorming list:

1. User profile
2. Sign in using email and password.
3. Occasions (almost like groups)
4. Create page (can assign to an occasion)
5. Posts
6. Comments
7. Ingredients/instructions
8. Grocery lists
9. Recipes (to be marked as public or private) (able to save it to your grocery list)
10. Categories
11. Followers or following accounts.
12. Search page

Table ideas:

Recipes table: This will have the info for the different recipes.

1. Recipes_id
2. User_id(foreign key)
3. name
4. Ingredients
5. Instructions

6. Grocery list
7. Comments
8. Category_id
9. Public / private feature (true/false)
10. Saved/shared (true/false)

Users table: This table will be for the user and its info.

1. User_id
2. User_name
3. Email
4. Password
5. Followers
6. Following
7. recipes
8. posts
9. recipe_id
10. saved

Search table: This table will be for searching different recipes, categories, and users.

1. category_id

2. recipes_id
3. search feature(button)
4. follow(user_id)

Posts/feed table: This will be where the user can see the different posts from users and categories.

1. people you are following. User_id
2. Search
3. Category_id (foreign key)
4. Comments

Grocery table: This table will have the different ingredients for recipes and the categories that they may be in.

1. Grocery_id
2. Recipies_id(foreign keys)
3. Category_id(foreign key)
4. text

Category table: This table will be for the categories and the different recipes that they will have.

1. category_id
2. occasions name
3. recipes_id
4. user_id

relationships:

one to one:

1. user to recipe because user can have a recipe.

one to many:

1. Recipe table to user table because a user can have many recipes.
2. Recipe to category one recipe can be in many different categories
3. Grocery table to categories and recipes

many to many:

1. Feed to category is here because the feed is a combo of everything.
2. Category to recipe. Different categories that can show more than one recipe.
3. Categories to groceries to recipes

Columns:

In **category** the **category_id** is a **serial** because it will be a **primary key** because there will be different categories.

The name in **category** is **VARCHAR** because it will just be a name.

The **recipe_id** in **category** is a **foreign key** and an **interger** because it will just be getting the **recipe_id** from **recipe**.

The **user_id** in **category** is a **foreign key** and an **integer** because it will just be getting the **id** from **user**.

The **user_id** is a **serial primary key** because there will be different users.

The **username** is **VARCHAR** because it will just simply be a name.

Followers will be an **integer** to just who how many there are.

Following will also be an **integer** to show how many there are.

Recipe_id will be a **foreign key** and an **integer** because it will be getting the id from recipe.

Password will be **VARCHAR** just because it will be a password.

Email will also be **VARCHAR** and unique in order to make it so that one email has one account.

Post will be **text**.

Saved will be a **Boolean** just because it will tell whether it was saved or not.

The **category_id** will be a **foreign key** and an **integer** in **search**.

The **recipe_id** will be a **foreign key** and an **integer** as **well**.

Search will be **text**.

User_id will be a **foreign key** and an **integer**.

The **grocery_id** will be a **serial primary key** because there will be different lists.

The **recipe_id** will be a **foreign key** and an **integer** in **grocery**.

The **category_id** will also be a **foreign key** and an **integer**.

Text will just be **text**.

The **user_id** in feed will be **a foreign key** and an **integer.**

Search will simply be **text**.

Category_id will be a **foreign key** and an **integer**.

Comments will be **VARCHAR** as to not have to big of a comment.

Recipe_id will be a **serial primary key** because there will be **different recipes**.

User_id will be **an integer** and a **foreign key** because it will be getting it from **user**.

Name will be **VARHCAR** to keep names simple.

Ingredients will also be **VARCHAR** for the same reason.

Instructions will be **text** to make the instructions as specific as possible.

Grocery_id will be a **foreign key** and **an integer** because it will get it form **grocery.**

Comments will be **VARCHAR** as to make comments not too big.

Category_id will be a **foreign key** and **an integer** because it will get it from **category.**

Public/private will be a **Boolean** for true or false.

Saved and **shared** will also be a **Boolean** for that same reason.

```
CREATE TABLE user (
        user_id SERIAL PRIMARY KEY,
  username VARCHAR(255),
        followers INTEGER(255),
  following INTEGER(255),
  recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
  password VARCHAR(255),
  email VARCHAR(255),
  posts TEXT,
  saved BOOLEAN
);

CREATE TABLE category (
        category_id SERIAL PRIMARY KEY,
  name VARCHAR(255),
```

```
  recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
  user_id INTEGER NOT NULL REFERENCES user(user_id)
  );

CREATE TABLE grocery (
        grocery_id SERIAL PRIMARY KEY,
  recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
  category_id INTEGER NOT NULL REFERENCES category(category_id),
  text
);

CREATE TABLE recipe (
        recipe_id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL REFERENCES user(user_id),
  name VARCHAR(255),
  ingredients VARCHAR(255),
  instructions text,
        grocery_id INTEGER NOT NULL REFERENCES grocery(grocery_id),
  comments TEXT,
        category_id INTEGER NOT NULL REFERENCES
category(category_id),
  public_private BOOLEAN,
```

```
  saved BOOLEAN,
  shared BOOLEAN,
);


CREATE TABLE feed (
        user_id INTEGER NOT NULL REFERENCES user(user_id),
  search TEXT,
  category_id INTEGER NOT NULL REFERENCES category(category_id),
  comments VARCHAR(255)
);

CREATE TABLE search (
        category_id INTEGER NOT NULL REFERENCES
category(category_id),
        recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
  search TEXT,
  user_id INTEGER NOT NULL REFERENCES user(user_id)
);
```