Database Model Grocery List app

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

Brainstorming

- Users (id)
    - User recipes(if public or private)
    - User occasions With recipes
- Ingredients
- Recipes (general ideas)
- Grocery list (with ingredients maybe)
- Occasions with recipes
    - (holidays,parties, etc)
- Calendar for when occasions will occur:
    - Helps remind them of what groceries they will need to create the meal

EX sketch:
Sign in info
Profile/mini bio about the user and maybe a favorite recipe they can share(if public)
And grocery list of things they need for recipes they have coming up maybe?

Browse page where the user can view other public recipes or maybe follow users that they like.

Home page just includes featured recipes or occasions that may be of interest to the user.

Section 2

Tables:

- Users (for each user of the app)
    - Id
    - Name
    - Email
    - Username
    - password
    - Created recipes (private recipe boolean)
    - Occasions (foreign key)
    - Saved_recipes (comes from public recipes)
    - Grocery list (foreign key of grocery list)
- Ingredients
    - Id
    - Name
    - description
- Recipes
    - Id
    - Name
    - Ingredient_id (foreign key)
    - Instructions
    - User_id (foreign key to users table)
    - grocery_id(foriegn key)
- Grocery list
    - Id
    - recieps_id(foreign key)
- Occasions
    - Id
    - Name
    - Recipe_id
    -
- User_recipes
    - User_id (foreign key)
    - Recipes_id (foreign key)
- Recipe_ingredients (middle man)
    - id_recipes(foreign key)
    - id_ingredients(for

Section 3

Relationships:

One-to-many:

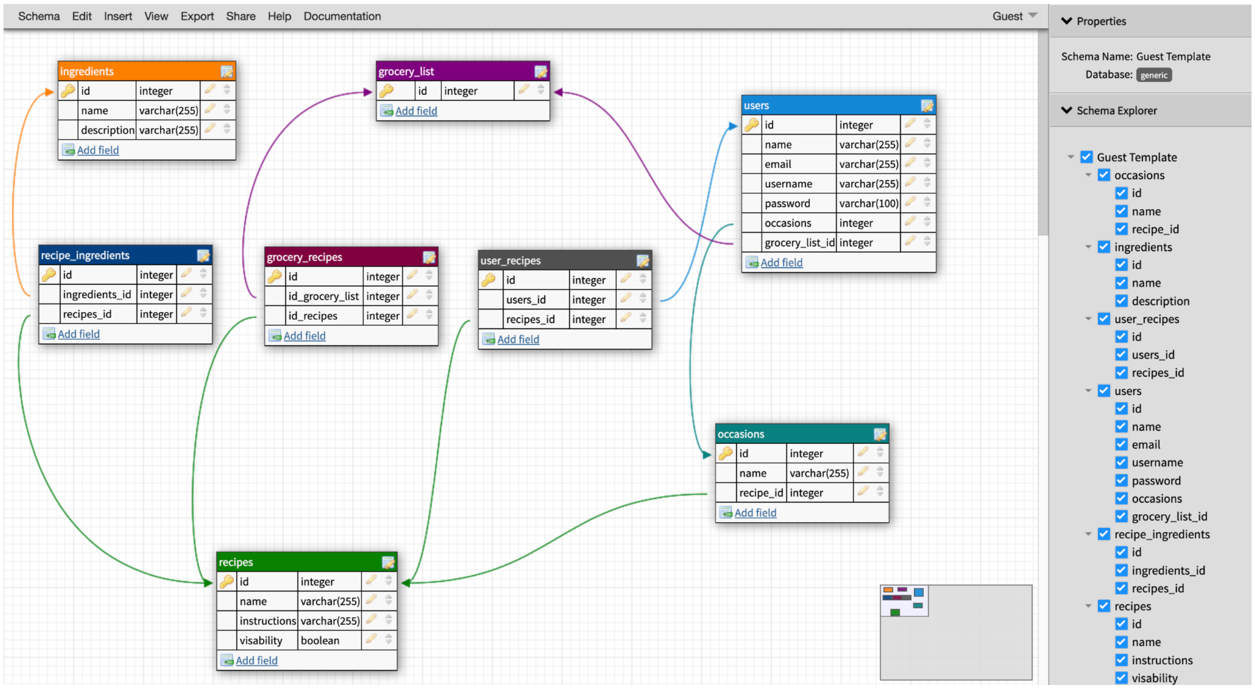
- Users to occasions - one user can have many occasions but those occasions are tied to one user

Many-to-Many:

- Users to recipes - many users can have many recipes and many recipes can go to many users
- Public recipes to ingredients many recipes can have many ingredients and many ingredients can go to many recipes
- Recipe to grocery list: one recipe can go to many grocery list but grocery l

One-to-one:

- Users to grocery list - every user can have a grocery list and one grocery list is assigned to one user



ingredients	
id	integer
name	varchar(255)
description	varchar(255)

grocery_list	
id	integer

users	
id	integer
name	varchar(255)
email	varchar(255)
username	varchar(255)
password	varchar(100)
occasions	integer
grocery_list_id	integer

recipe_ingredients	
id	integer
ingredients_id	integer
recipes_id	integer

grocery_recipes	
id	integer
id_grocery_list	integer
id_recipes	integer

user_recipes	
id	integer
users_id	integer
recipes_id	integer

occasions	
id	integer
name	varchar(255)
recipe_id	integer

recipes	
id	integer
name	varchar(255)
instructions	varchar(255)
visability	boolean

Properties

Schema Name: Guest Template
Database: generic

Schema Explorer

- ☑ Guest Template
  - ☑ occasions
    - ☑ id
    - ☑ name
    - ☑ recipe_id
  - ☑ ingredients
    - ☑ id
    - ☑ name
    - ☑ description
  - ☑ user_recipes
    - ☑ id
    - ☑ users_id
    - ☑ recipes_id
  - ☑ users
    - ☑ id
    - ☑ name
    - ☑ email
    - ☑ username
    - ☑ password
    - ☑ occasions
    - ☑ grocery_list_id
  - ☑ recipe_ingredients
    - ☑ id
    - ☑ ingredients_id
    - ☑ recipes_id
  - ☑ recipes
    - ☑ id
    - ☑ name
    - ☑ instructions
    - ☑ visability

Columns

Ingredients:

- I chose id because each ingredient needs a special id
- They will also need a name for each ingredient
- And a description of what the ingredient is

Recipe_ingredients:

- Needs a special id for each recipe and its ingredients

- Then I reference both the ingredients table and the recipes table so that it pulls the data into one so that they can communicate with each other

Recipes:

- Needs a special id for each recipe
- Needs a name so people can identify what the recipe is
- Instructions so that the user can know how to make the recipe
- And if its a private or public recipe

Users:

- Id for each individual user
- The name of who the user is
- And email and username to identify each user
- Password thats specific to the user so that they can sign in.
- Occasions is what occasions the user has made for themselves.
- Grocery list because each user has their own specific grocery list

Occasions:

- Id is the id tied to each specific name of the occasion
- Name of what the occasion was called by the user
- And its tied to a recipe so that it populates what recipe they can make for that occasion

grocery_list :

- Is just an id of the list tied to the specific user
- And this list is linked to the recipes so that it can pull the recipes and add that to the list of things the user needs to purchase

Grocery_recipes and user_recipes:

- Both of these act as middlemen to help connect the data for our many to many tables.
- Both have id's that reference the other tables.

<u>Section 4 code</u>

```
-- CREATE TABLE ingredients (
--   id SERIAL PRIMARY KEY,
--   name VARCHAR(255) NOT NULL,
--   description VARCHAR(255) NOT NULL
--   );

--   CREATE TABLE grocery_list(
--   id SERIAL PRIMARY KEY
--   );

--   CREATE TABLE recipes (
--   id SERIAL PRIMARY KEY,
--   name VARCHAR(255) NOT NULL,
--   instructions VARCHAR(255) NOT NULL,
--   visability BOOLEAN
--   );
--   CREATE TABLE recipe_ingredients (
--   id SERIAL PRIMARY KEY,
--   ingredients_id INT NOT NULL REFERENCES recipes(id),
--   recipes_id INT NOT NULL REFERENCES ingredients(id)

--   );

-- CREATE TABLE grocery_recipes (
--   id SERIAL PRIMARY KEY,
--  id_grocery_list INT NOT NULL REFERENCES grocery_list(id),
--   id_recipes INT NOT NULL REFERENCES recipes(id)

--   );



-- CREATE TABLE occasions (
--   id SERIAL PRIMARY KEY,
--   name VARCHAR(255) NOT NULL,
--   recipe_id INT REFERENCES recipes(id)

--   );

-- CREATE TABLE users (
--   id SERIAL PRIMARY KEY,
--   name VARCHAR(255) NOT NULL,
--   email VARCHAR(255) NOT NULL,
--   username VARCHAR(255) NOT NULL,
--   password VARCHAR(100) NOT NULL,
--   occasions INT NOT NULL REFERENCES occasions(id),
--   grocery_list_id INT NOT NULL REFERENCES grocery_list(id)

--   );

--  CREATE TABLE user_recipes  (
--   id SERIAL PRIMARY KEY,
--          users_id INT NOT NULL REFERENCES users(id),
--   recipes_id INT NOT NULL REFERENCES recipes(id)

--   );
```