Recipe & Grocery List App

Users:

- user_id
- user_name
- email
- password

One to many - to recipe list, to occasion list, grocery list

Recipes

- recipe_id
- recipe_name
- recipe_author (relate to user_id in users one to many)
- body
- ?isPrivate
- rating

many to one to user
many to many to ingredients
many to many to occasions

Comments

- comment_id
- user_id (relate to Users - one to many)
- recipe_id
- body

one to one to User

Ingredients

- ingredient_id
- ingredient_name

Many to many to grocery list,
Many to many to recipe list

RecipeIngred Table

- recing_id
- recipe_id (relate to recipe_id in Recipes - many to many)
- ingredient_id (relate to ingredient_id in Ingredients - many to many)

GroceryList

- list_id
- list_name
- user_id (relation to Users one to many)
- ingredient_id (relate to the same in Ingredients - one to many)
- ?isPrivate

Many to one to User
Many to many to ingredients

Occasions

- occ_id
- user_id (relate to Users one to many)
- recipe_id (relate to Recipes one to many)
- Occasion name
- ?isPrivate

many to one to user
many to many to recipe

Table Ideas:

1. Users:
   Description: This table will store information about the app users. Each row will represent an individual user with details such as user_id, user_name, email, password, and rating. The table will be related to other tables like Recipes, Comments, GroceryList, and Occasions through various one-to-many relationships.

2. Recipes:
   Description: The Recipes table will contain data about the recipes created by users. Each row will represent a specific recipe with attributes like recipe_id, recipe_name, recipe_author (related to user_id in Users table), body (instructions and description), isPrivate (indicating if the recipe is private or public), and rating. It will have many-to-one relationships with the Users table (for the author) and many-to-many relationships with the Ingredients and Occasions tables.

3. Comments:
   Description: This table will hold comments made by users on different recipes. Each row will represent a comment with attributes like comment_id, user_id (related to user_id in Users table), recipe_id (related to recipe_id in Recipes table), and body (comment text). The table will have a one-to-one relationship with the Users table for the user who posted the comment.

4. Ingredients:
   Description: The Ingredients table will store information about various ingredients used in recipes. Each row will represent an ingredient with attributes like ingredient_id and ingredient_name. This table will have many-to-many relationships with both the Recipes and GroceryList tables.

5. RecipeIngred Table:
   Description: This table will establish a many-to-many relationship between recipes and ingredients. It will store the associations between recipes and the ingredients they require. Each row will have a unique recing_id, along with recipe_id (related to recipe_id in Recipes table) and ingredient_id (related to ingredient_id in Ingredients table).

6. GroceryList:
Description: The GroceryList table will hold data about grocery lists created by users. Each row will represent a grocery list with attributes like list_id, list_name, user_id (related to user_id in Users table), ingredient_id (related to ingredient_id in Ingredients table), and isPrivate (indicating if the list is private or public). It will have many-to-one relationships with the Users table and many-to-many relationships with the Ingredients table.

7. Occasions:
Description: This table will contain information about occasions created by users and associated with recipes. Each row will represent an occasion with attributes like occ_id, user_id (related to user_id in Users table), recipe_id (related to recipe_id in Recipes table), and isPrivate (indicating if the occasion is private or public). It will have many-to-one relationships with the Users table and many-to-many relationships with the Recipes table.

These tables will form the foundation of the data model for the Recipe and Grocery List App, allowing efficient storage and retrieval of information related to users, recipes, comments, ingredients, grocery lists, and occasions.

Relationships:

1. One-to-One:
- Comments and Users:
  - Explanation: Each comment in the Comments table is made by one specific user, and each user can have multiple comments. Therefore, there is a one-to-one relationship between the Comments and Users tables.

2. One-to-Many:
- Users and Recipes:
  - Explanation: A user can create multiple recipes, but each recipe is associated with only one user. Hence, there is a one-to-many relationship between the Users and Recipes tables.
- Users and GroceryList:
  - Explanation: A user can have multiple grocery lists, but each grocery list belongs to only one user. Therefore, there is a one-to-many relationship between the Users and GroceryList tables.
- Recipes and Comments:
  - Explanation: A recipe can have multiple comments, but each comment is related to a specific recipe. This establishes a one-to-many relationship between the Recipes and Comments tables.
- Ingredients and RecipeIngred Table:
  - Explanation: An ingredient can be used in multiple recipes, but each instance of an ingredient being used in a specific recipe is unique. This forms a one-to-many relationship between the Ingredients table and the RecipeIngred table.

3. Many-to-Many:
- Recipes and Ingredients:

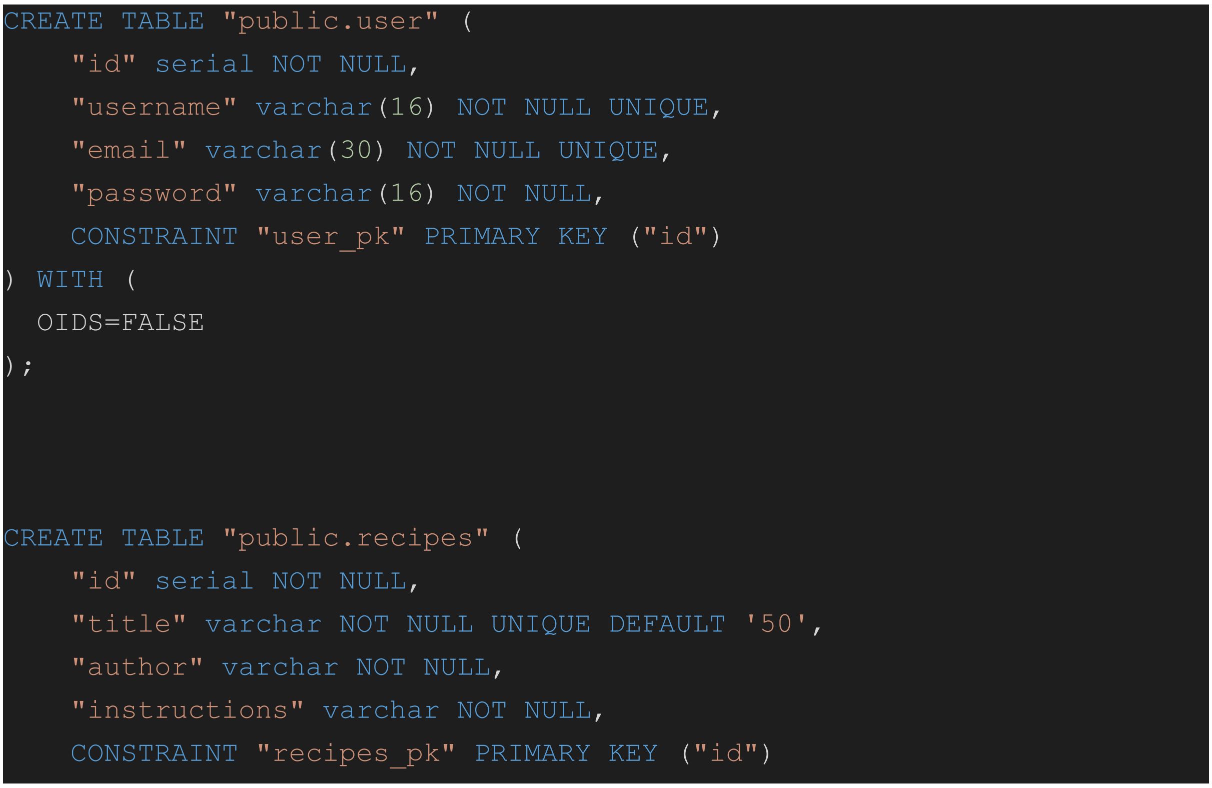
    - Explanation: Multiple recipes can use the same ingredient, and each recipe can have multiple ingredients. This results in a many-to-many relationship between the Recipes and Ingredients tables, which is bridged by the RecipeIngred table.
  - Recipes and Occasions:
    - Explanation: A recipe can be associated with multiple occasions, and each occasion can have multiple recipes. Thus, there is a many-to-many relationship between the Recipes and Occasions tables.
  - Ingredients and GroceryList:
    - Explanation: Multiple grocery lists can contain the same ingredient, and each ingredient can be present in multiple grocery lists. Hence, there is a many-to-many relationship between the Ingredients and GroceryList tables.

Step 4:

Additional Tables and Descriptions:

8. Tags:
  Description: The Tags table will store information about tags or labels that can be associated with recipes. Each row will represent a tag with attributes like tag_id, name, and description. Tags will help categorize and organize recipes based on common themes or attributes.

9. RecipeTag Table (Many-to-Many Relationship between Recipes and Tags):
  Description: This table will establish a many-to-many relationship between recipes and tags. It will store the associations between recipes and the tags they are associated with. Each row will have a unique rectag_id, along with recipe_id (related to recipe_id in Recipes table) and tag_id (related to tag_id in Tags table).



```
CREATE TABLE "public.user" (
    "id" serial NOT NULL,
    "username" varchar(16) NOT NULL UNIQUE,
    "email" varchar(30) NOT NULL UNIQUE,
    "password" varchar(16) NOT NULL,
    CONSTRAINT "user_pk" PRIMARY KEY ("id")
) WITH (
  OIDS=FALSE
);



CREATE TABLE "public.recipes" (
    "id" serial NOT NULL,
    "title" varchar NOT NULL UNIQUE DEFAULT '50',
    "author" varchar NOT NULL,
    "instructions" varchar NOT NULL,
    CONSTRAINT "recipes_pk" PRIMARY KEY ("id")
```

```
) WITH (
  OIDS=FALSE
);



CREATE TABLE "public.Ingredients" (
    "ingredient_id" serial NOT NULL,
    "ingredient_name" varchar NOT NULL,
    CONSTRAINT "Ingredients_pk" PRIMARY KEY ("ingredient_id")
) WITH (
  OIDS=FALSE
);



CREATE TABLE "public.GroceryList" (
    "list_id" serial NOT NULL,
    "list_name" varchar NOT NULL,
    "list_details" varchar NOT NULL,
    "isPrivate" BOOLEAN NOT NULL DEFAULT 'True',
    CONSTRAINT "GroceryList_pk" PRIMARY KEY ("list_id")
) WITH (
  OIDS=FALSE
);



CREATE TABLE "public.Occasions" (
    "occ_id" serial NOT NULL,
    "occ_name" varchar NOT NULL,
    "is_private" BOOLEAN NOT NULL DEFAULT 'True',
    CONSTRAINT "Occasions_pk" PRIMARY KEY ("occ_id")
) WITH (
  OIDS=FALSE
);



CREATE TABLE "public.RecAutBridge" (
    "id" serial NOT NULL,
    "recipe_id" serial NOT NULL,
    "user_id" integer NOT NULL,
    CONSTRAINT "RecAutBridge_pk" PRIMARY KEY ("id")
) WITH (
  OIDS=FALSE
);




ALTER TABLE "recipes" ADD CONSTRAINT "recipes_fk0" FOREIGN KEY ("id")
REFERENCES "Ingredients"("ingredient_id");

ALTER TABLE "Ingredients" ADD CONSTRAINT "Ingredients_fk0" FOREIGN KEY
("ingredient_id") REFERENCES "GroceryList"("list_id");

ALTER TABLE "GroceryList" ADD CONSTRAINT "GroceryList_fk0" FOREIGN KEY
("list_id") REFERENCES "user"("id");

ALTER TABLE "Occasions" ADD CONSTRAINT "Occasions_fk0" FOREIGN KEY
("occ_id") REFERENCES "recipes"("id");

ALTER TABLE "RecAutBridge" ADD CONSTRAINT "RecAutBridge_fk0" FOREIGN
KEY ("recipe_id") REFERENCES "recipes"("id");
ALTER TABLE "RecAutBridge" ADD CONSTRAINT "RecAutBridge_fk1" FOREIGN
KEY ("user_id") REFERENCES "user"("id");
```