Brainstorming

User
- User email
- Password
- User ID

Recipes (parent)
- Recipe id
- User ID
- Cooking instructions id ref
- Ingredients id reference
- Name - VARCHAR
- Privacy setting (public/private)
- Timestamp

Cooking Instructions(child)
- Cooking instructions id
- Text body of instructions

Ingredients (child)
- Ingredient ID
- Name - VARCHAR

Grocery List
- Grocery List ID
- Recipes.ingredients
- User ID ref

Occasions
- Occasion ID?
- Name
- User ID
- Recipe id

Table Ideas

User: This table will hold information for users - email, password, and user ids
Recipes: Holds information for individual recipes
Recipe Instructions: Holds the text for the instructions
Ingredients: holds the info for individual ingredients for recipes
Grocery list: references the recipe list to see what ingredients are needed
Occasions: Holds information for occasion, can add recipes to occasion

Relationships

One-to-one

- *User* => *grocery list* - Each user can only have one grocery list, and each grocery list is only assigned to one user
- *Recipes* => *recipe instructions* - Each recipe can only have one set of instructions, each set of instructions can only be assigned to one recipe (can't use the instructions for pie to make a soup)

One-to-many

- *Users* => *recipes* - Each user can have multiple recipes, but each recipe can only be assigned to one user
- *Users* => *occasions* - Each user can have multiple occasions, but each occasion can only be assigned to one user

Many-to-many

- *Recipes* => *ingredients* - One recipe can have many ingredients, but one ingredient can be assigned to multiple recipes

Columns

User:
Email - to associate recipes, lists, and occasions with a specific user.
Pass- so the users information is safe.
User id- to reference the user in another part of the database.

Recipes
Recipe id - unique value for reference.
User ID - associating the recipe back to the user
Cooking instructions id ref - associating the cooking instructions with a recipe.
Ingredients id reference - associating ingredients with the recipe.
Name VARCHAR - name of the recipe
Privacy setting (public/private) - whether the recipe is private or public
Timestamp - showing when the recipe was posted.

Cooking Instructions
Cooking instructions id - unique value for reference.
Text body of instructions - the instructions for the recipe.

Ingredients (child)
Ingredient ID - unique value for reference.
list_of_ingredients VARCHAR - what the name says

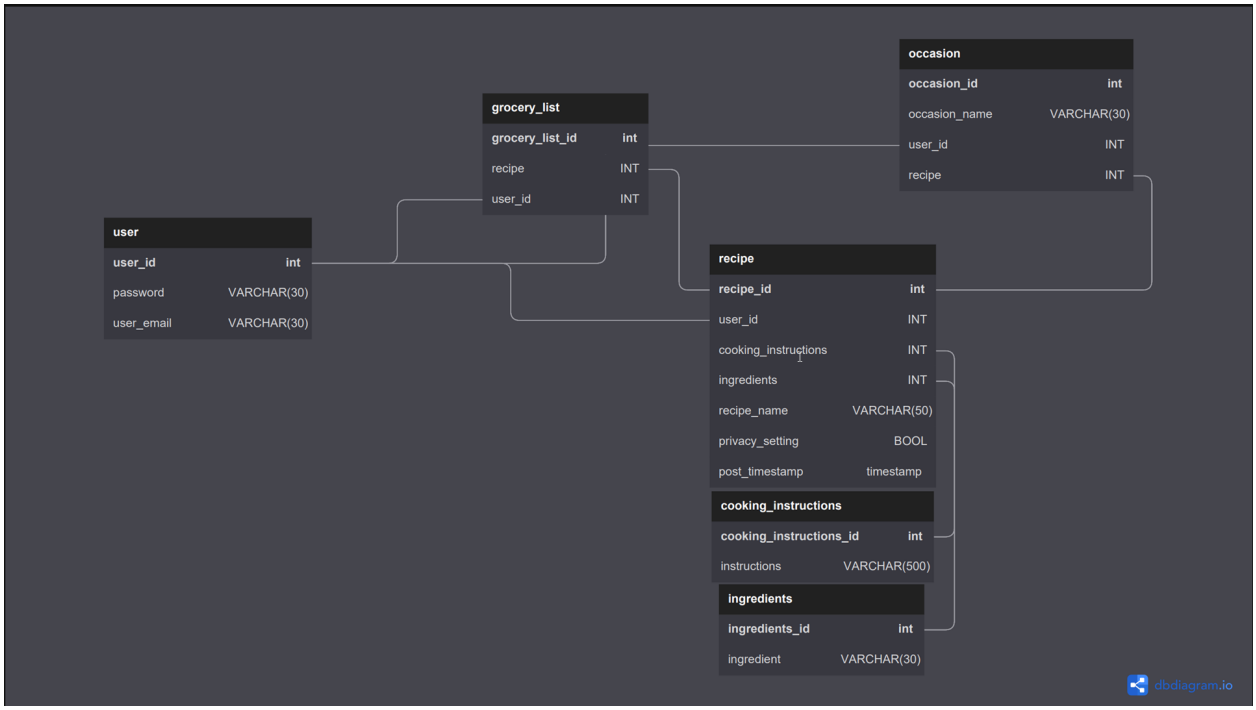
Grocery List
Grocery List ID - unique value for reference.
Recipes.ingredients - referencing recipes to add to the grocery list.
User ID ref - associating it with a user.

Occasions
Occasion ID - unique value for reference.
Name - Occasion name
User ID - associating it with a user.
Recipe id - referencing recipes to add to the occasions list

PostgreSQL Export

```
CREATE TABLE "users" (
  "user_id" SERIAL PRIMARY KEY,
  "user_email" VARCHAR(40),
  "user_password" VARCHAR(1000)
);

CREATE TABLE "recipe_instructions" (
  "instructions_id" SERIAL PRIMARY KEY,
  "instructions_body" VARCHAR(1000)
);

CREATE TABLE "recipe_ingredients" (
  "ingredients_id" SERIAL PRIMARY KEY,
  "ingredients_body" VARCHAR(500)
);

CREATE TABLE "recipes" (
  "recipe_id" SERIAL PRIMARY KEY,
  "user_id" INT NOT NULL,
  "ingredients_id" INT NOT NULL,
  "instructions_id" INT NOT NULL,
  "name" VARCHAR(100),
  "private" BOOL,
  "recipe_timestamp" timestamp
);

CREATE TABLE "grocery_list" (
  "grocery_list_id" SERIAL PRIMARY KEY,
  "recipe_id" INT,
```

```
  "user_id" INT NOT NULL
);

CREATE TABLE "occasions" (
  "occasion_id" SERIAL PRIMARY KEY,
  "user_id" INT NOT NULL,
  "recipe_id" INT,
  "name" VARCHAR(100)
);

ALTER TABLE "recipes" ADD FOREIGN KEY ("user_id") REFERENCES "users"
("user_id");

ALTER TABLE "recipes" ADD FOREIGN KEY ("ingredients_id") REFERENCES
"recipe_ingredients" ("ingredients_id");

ALTER TABLE "recipes" ADD FOREIGN KEY ("instructions_id") REFERENCES
"recipe_instructions" ("instructions_id");

ALTER TABLE "grocery_list" ADD FOREIGN KEY ("recipe_id") REFERENCES
"recipes" ("recipe_id");

ALTER TABLE "grocery_list" ADD FOREIGN KEY ("user_id") REFERENCES "users"
("user_id");

ALTER TABLE "occasions" ADD FOREIGN KEY ("user_id") REFERENCES "users"
("user_id");

ALTER TABLE "occasions" ADD FOREIGN KEY ("recipe_id") REFERENCES "recipes"
("recipe_id");
```


occasion
occasion_id int
occasion_name VARCHAR(30)
user_id INT
recipe INT
grocery_list
grocery_list_id int
recipe INT
user_id INT
user
user_id int
password VARCHAR(30)
user_email VARCHAR(30)
recipe
recipe_id int
user_id INT
cooking_instructions INT
ingredients INT
recipe_name VARCHAR(50)
privacy_setting BOOL
post_timestamp timestamp
cooking_instructions
cooking_instructions_id int
instructions VARCHAR(500)
ingredients
ingredients_id int
ingredient VARCHAR(30)
dbdiagram.io