Brainstorming:

1. User data: email, password, name, profile picture
2. Recipe data: title, description, ingredients, instructions, tags, public/private flag, user ID
3. Occasion data: name, description, user ID
4. Grocery list data: name, ingredients, user ID
5. Relationships between data:
    a. a user can have many recipes, occasions, and grocery lists
    b. a recipe can have many ingredients and instructions
    c. an ingredient can be in many recipes and grocery lists
    d. an occasion can have many recipes
    e. a grocery list can have many ingredients
6. User authentication and authorization
7. Search and filter functionality for recipes
8. User interactions: create, read, update, delete for recipes, occasions, and grocery lists
9. User roles and permissions (e.g., admin, editor, viewer)
10. User activity log for auditing purposes.

Table Ideas:

1. Users: this table will hold information about users, such as email, password, name, and profile picture. Each row will represent an individual user.
2. Recipes: this table will hold information about recipes, such as title, description, ingredients, instructions, tags, public/private flag, and user ID. Each row will represent an individual recipe.
3. Occasions: this table will hold information about occasions, such as name, description, and user ID. Each row will represent an individual occasion.
4. Grocery lists: this table will hold information about grocery lists, such as name, ingredients, and user ID. Each row will represent an individual grocery list.
5. Ingredients: this table will hold information about ingredients, such as name, unit of measurement, and category. Each row will represent an individual ingredient.
6. Recipe ingredients: this table will hold information about the relationship between recipes and ingredients, such as the quantity of each ingredient needed for a recipe. Each row will represent the quantity of a specific ingredient needed for a specific recipe.
7. User recipe ratings: this table will hold information about the ratings users give to recipes they have tried, such as the rating and user ID. Each row will represent a rating for a specific recipe by a specific user.
8. User recipe favorites: this table will hold information about the recipes users have marked as their favorites, such as the recipe ID and user ID. Each row will represent a favorite recipe for a specific user.
9. User activity log: this table will hold information about user activities, such as the type of activity (e.g., login, recipe creation), timestamp, and user ID. Each row will represent a specific user activity.

Relationships:

1. One-to-one:
    - None identified.

2. one-to-many:

    - Users to Recipes: a user can have many recipes, but a recipe can only belong to one user.
    - Users to Occasions: a user can have many occasions, but an occasion can only belong to one user.
    - Users to Grocery Lists: a user can have many grocery lists, but a grocery list can only belong to one user.
    - Recipes to Recipe Ingredients: a recipe can have many ingredients, but each ingredient belongs to only one recipe.
    - Ingredients to Recipe Ingredients: an ingredient can be used in many recipes, but each recipe ingredient belongs to only one ingredient.
    - Recipes to User Recipe Ratings: a recipe can have many ratings from different users, but each rating belongs to only one recipe and one user.
    - Recipes to User Recipe Favorites: a recipe can be marked as a favorite by many users, but each favorite recipe belongs to only one recipe and one user.
3. many-to-many:

    - Recipes to Ingredients: a recipe can have many ingredients, and an ingredient can be used in many recipes.
    - Users to Recipes: a user can have many favorite recipes and can rate many recipes, and a recipe can have many ratings and can be marked as a favorite by many users.

Columns:

USERS
id: int (primary key)
email: string (unique, not null) - to identify the user and allow them to sign in
password: string (not null) - to authenticate the user
name: string (not null) - to display the user's name in the app
profile_picture: string - to store the URL of the user's profile picture, if they choose to upload one

RECIPES
id: int (primary key)
title: string (not null) - to display the recipe's title in the app
description: string - to provide a brief description of the recipe
instructions: text (not null) - to store the recipe's instructions, which may be lengthy
public: boolean (not null) - to indicate whether the recipe is public or private
user_id: int (foreign key references USERS.id) (not null) - to associate the recipe with the user who created it

OCCASIONS
id: int (primary key)
name: string (not null) - to display the occasion's name in the app
description: string - to provide a brief description of the occasion
user_id: int (foreign key references USERS.id) (not null) - to associate the occasion with the user who created it

GROCERY_LISTS
id: int (primary key)
name: string (not null) - to display the grocery list's name in the app
user_id: int (foreign key references USERS.id) (not null) - to associate the grocery list with the user who created it

INGREDIENTS
id: int (primary key)
name: string (unique, not null) - to identify the ingredient
unit_of_measurement: string (not null) - to indicate the unit of measurement for the ingredient
category: string - to categorize the ingredient (e.g. "vegetable", "protein", etc.)

RECIPE_INGREDIENTS
recipe_id: int (foreign key references RECIPES.id) (not null) - to associate the recipe ingredient with the recipe
ingredient_id: int (foreign key references INGREDIENTS.id) (not null) - to associate the recipe ingredient with the ingredient
quantity: float (not null) - to indicate the quantity of the ingredient required for the recipe

USER_RECIPE_RATINGS
user_id: int (foreign key references USERS.id) (not null) - to associate the rating with the user who submitted it
recipe_id: int (foreign key references RECIPES.id) (not null) - to associate the rating with the recipe it applies to
rating: int (not null) - to store the rating (e.g. 1-5 stars)

USER_RECIPE_FAVORITES
user_id: int (foreign key references USERS.id) (not null) - to associate the favorite with the user who marked it
recipe_id: int (foreign key references RECIPES.id) (not null) - to associate the favorite with the recipe it applies to

USER_ACTIVITY_LOG
id: int (primary key)
user_id: int (foreign key references USERS.id) (not null) - to associate the activity with the user who performed it
activity_type: string (not null) - to indicate the type of activity (e.g. "created recipe", "added ingredient to grocery list", etc.)

timestamp: datetime (not null) - to record the time the activity was performed