

TA

User

Email varchar unique
Password varchar
Id SERIAL INTEGER KEY
Recipes f.k. id
Grocery list integer f.k
Occasions integer f. K.

Recipe

Id
creator_id
Ingredients
Instructions text

Ingredients

Id
Name
Nutrition facts
Price
Quantity text

Grocery list

Id
User_id
Ingredients_id

Occassions

id
Recipes
Date
budget

PART 2 TABLE IDEAS

User

Email varchar unique
Password varchar
Id SERIAL INTEGER KEY
Recipes f.k. Id this will point to any recipes the user has
Grocery list integer f.k
Occasions integer f. K.

This is going to have information about the person using the app, including the recipes that they are working on, their grocery list and the occasions that they are planning for

Recipe

- Id serial primary key
- Creator_id f.k. The owner of this recipe
- Ingredients f.k the ingredients used in this recipe
- Instructions text varchar or text

This will have links to the creator of the recipe, maybe users would like to trade secrets or compliment each other. Links to the ingredients and a text box with steps and instructions

Ingredients

- Id serial primary key
- Name varchar (unique)
- Nutrition facts textbox or image or RDA table (allow null)
- Price float
- Got_it boolean, do i have, that, (cross it off the list)

This lists and describes the ingredient and has a quantity for each ingredient

Grocery list

- Id: serial primary key
- User_id f.k. the owner of the list
- recipe_id f.k to the ingredients table

Has the recipes it uses and gets the ingredients from them

Occassions

- Id serial primary key
- Recipes f.k link to the recipes we're using
- Date text date box
- Budget float how much you can spend on this

These are the parties you planned.

RECIPE-INGREDIENTS table

- Id serial primary key
- Recipe_id foreign key pointing to the recipe that uses this
- Ingredients_id foreign key pointing to the ingredient
- quantity : varchar how much of this you need.

Association table connecting recipe and ingredients

Occasion_recipe table

Id serial primary key

Recipe_id foreign key pointing to the recipe that is used on the occasion

Occasion_id foreign key pointing to the occasion that will be using this recipe

Quantity integer how many times we will make this recipe.

Association tables

User_recipe table

Id serial primary key

Recipe_id f.k pointing to the recipe that the user has

User_id f.k. Pointing to the user who like this recipe

RELATIONSHIPS

One to one

User to grocery list

We decided you will have just a single grocery list for your session, when you buy stuff it comes off your grocery stuff, and stuff you don't have stays on your list.

One to many

User to occasion

Grocery to recipe

You're only going to have one grocery list, and many recipes that you might be shopping for.

Many to Many

Recipe to ingredients

There are many ingredients that will use the same ingredient list, and there will be many ingredients on each recipe.

Occasion to recipe

You might have multiple dishes you are preparing for your super bowl party

User to recipe

There will be many recipes that a user chooses to follow

COLUMNS

Recipe

Id integer serial primary key

Integer Serial primary is default for an id

Creator_id integer f.k. The owner of this recipe

Integer matches the data type of other ids

Ingredients integer f.k the ingredients used in this recipe

Integer matches previous data types

Instructions text varchar or text

We chose text for a blurb of instructions

Ingredients

Id integer serial primary key

Name varchar(50) (unique)

Nutrition facts textbox or image or RDA table (allow null)

Price float

Got_it boolean, do i have, that, (cross it off the list)

ID is an integer because that's the scheme we have chosen. Name will be a varchar 30 because that will cover most names. Nutrition facts will be a text box, because facts like that are often a paragraph size chunk of text. Price is float because that's the smallest decimal size. And got it is a boolean that lets you cross something off your list.

Grocery list

Id: serial primary key

User_id f.k. the owner of the list

recipe_id f.k to the ingredients table

Has the recipes it uses and gets the ingredients from them

I'm using integer for id's as normal practice, and the foreign keys are

Occassions

Id serial primary key

Recipes f.k link to the recipes we're using

Date text date box

Budget float how much you can spend on this

These are the parties you planned.

RECIPE-INGREDIENTS table

Id serial primary key

Recipe_id foreign key pointing to the recipe that uses this

Ingredients_id foreign key pointing to the ingredient

quantity : varchar how much of this you need.

Association table connecting recipe and ingredients

Occasion_recipe table

Id serial primary key

Recipe_id foreign key pointing to the recipe that is used on the occasion

Occasion_id foreign key pointing to the occasion that will be using this recipe

Quantity integer how many times we will make this recipe.

Association tables

User_recipe table

Id serial primary key

Recipe_id f.k pointing to the recipe that the user has

User_id f.k. Pointing to the user who like this recipe

```
CREATE TABLE ingredients (  
  id integer SERIAL PRIMARY KEY,  
  name varchar (50) NOT NULL,  
  nutrition text,  
  price float NOT NULL ,  
  got_it boolean NOT NULL  
);
```

```
CREATE TABLE occasion (  
  id SERIAL PRIMARY KEY,  
  name varchar(50) UNIQUE,  
  occasion_date DATE,  
  budget float  
);
```

```
CREATE TABLE recipe(  
  id SERIAL PRIMARY KEY,  
  instructions text,  
  public BOOLEAN  
);
```

```
CREATE TABLE occasion_recipe(  
  id SERIAL PRIMARY KEY,
```

```

    occasion_id integer references occasion(id),
    recipe_id integer REFERENCES recipe(id)
);
CREATE TABLE grocery_list(
    id SERIAL PRIMARY KEY,
    recipe_id INT REFERENCES recipe(id)
);
CREATE TABLE user_info (
    id SERIAL PRIMARY KEY,
    email VARCHAR(50) UNIQUE,
    password VARCHAR(20),
    grocery_id INT REFERENCES grocery_list(id),
    occasion_id INT REFERENCES occasion(id)
);

CREATE TABLE recipe_ingredients(
    id SERIAL PRIMARY KEY,
    recipe_id INT REFERENCES recipe(id),
    ingredient_id INT REFERENCES ingredients(id),
    quantity VARCHAR(50)
);
CREATE TABLE user_recipe(
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES user_info(id),
    recipe_id INT REFERENCES recipe(id)
);

```

On reflection after the lab we would change some of our 1 to many dependency directions