BRAINSTORMING

- User
- Private Recipe
- Public Recipe
- Post
- Follow
- Grocery list
- Products-ingredients & cookware/utensils
- Occasions
- Photo
- Comments
- Profile
- Instructions

TABLE IDEAS

User- info about user, each row is a different user, store email & password

- User ID
- User name
- User email
- User password
- User location
- User birthday

Private Recipe-info about recipe, each row is a different recipe, store ingredients, cookware needed & instructions

- Recipe ID
- User ID-REFERENCE-User -User ID
- Product ID-REFERENCE -Product-Product ID
- Photo ID-REFERENCE-Photo-Photo-ID

Public Recipe-info about recipe, each row is a different recipe, store ingredients, cookware needed & instructions

- Recipe ID
- User ID-REFERENCE-User -User ID
- Product ID-REFERENCE -Product-Product ID
- Photo ID-REFERENCE-Photo-Photo ID
- Post ID

Follow -info about followers

- Follower ID-REFERENCE-User -User ID (user being followed)
- Following ID-REFERENCE-User-User ID (user doing the following)

Grocery List - info about grocery list, products needed, user who has one, quantity of products needed, timestamp so we don’t make duplicates

- Grocery-ID
- User ID -REFERENCE-User-User ID

Product ID-REFERENCE-Product-Product ID
Quantity of Products
Timestamp

Products-info about products, each row will be a different product, contains product name, product information, product description

Product ID
Product name
Product info
Photo ID

Post-info about the post, who posted it, when they posted it.

Post ID
User ID-REFERENCE-User-User ID
Timestamp

Occasions-info about occasions created, user who created the occasion and the recipe they are referencing back to for that particular occasion

Occasions ID
User ID-REFERENCE-User-User ID
Public Recipe ID-REFERENCE-Public Recipe- Public Recipe ID
Private Recipe ID-REFERENCE-Private Recipe-Private Recipe ID
Products ID -REFERENCE-Products-Product ID
Post ID-REFERENCE-Post-Post ID

Profile-info about user, their profile photo

Profile ID
Photo ID-REFERENCE-Photo-Photo ID
User ID-REFERENCE User-User ID

Photos-relate to the user, photo of the products, photo of the dish, comments

Photo ID
User ID-REFERENCE-User-User ID

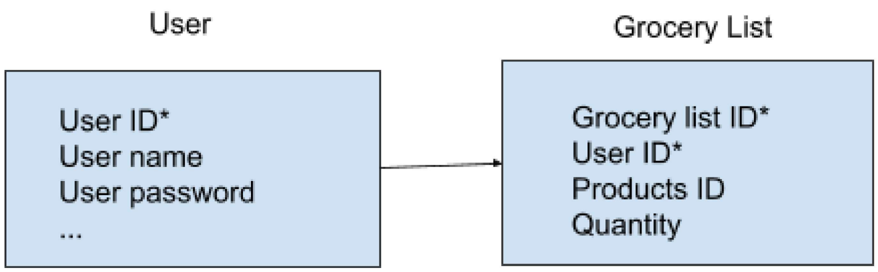
Comments

Comment ID
User ID-REFERENCE-User-User ID
Timestamp

RELATIONSHIPS

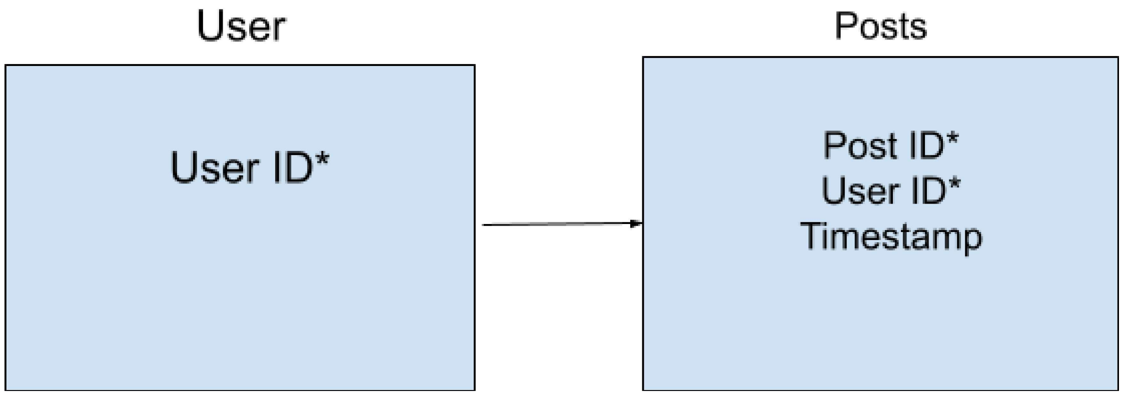
ONE TO ONE

The Grocery List table can only relate to the User table because it references back to the person who created the list. Without the user, the Grocery List would be voided.



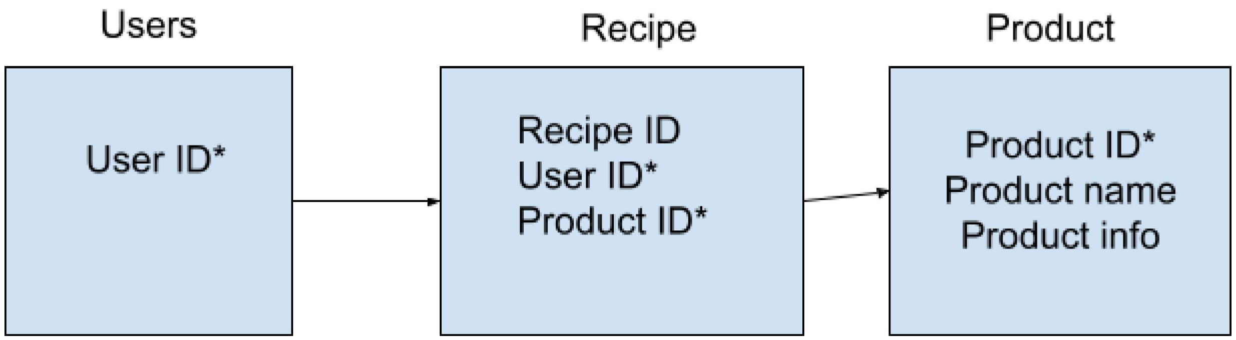
ONE TO MANY

In one to many relationships, one row on the User table can have many related rows on the Posts table. And rows on the Posts table can only relate to one row on the User's table.



MANY TO MANY

Users can have many to many relationships because one user can have multiple products in their recipe.



COLUMNS

User columns:

- User ID-primary key id for all users
- User name-stores user’s name
- User email-stores user’s email
- User password-stores user password
- User location-stores where user is located
- User birthday-stores user birthday

Recipe-info about recipe, each row is a different recipe, store ingredients, cookware needed & instructions. This one is different than the private recipe table because this one is accessible to public views.

- Recipe ID
- User ID-REFERENCE-User -User ID
- Product ID-REFERENCE -Product-Product ID
- Photo ID-REFERENCE-Photo-Photo ID
- Post ID

Follow -info about followers table stores all the information about the users who are followers or following a certain user and keeps track of the traffic on user gets.

- Follower ID-REFERENCE-User -User ID (user being followed)
- Following ID-REFERENCE-User-User ID (user doing the following)

Grocery List - info about grocery list, products needed, user who has one, quantity of products needed, timestamp so we don’t make duplicates The grocery list allows user to collect all the materials needed to make their recipe while also being able to access the recipes. It also allows them to see when they created the list so they don’t make duplicate lists.

- Grocery-ID
- User ID -REFERENCE-User-User ID
- Product ID-REFERENCE-Product-Product ID
- Quantity of Products
- Timestamp

Products-info about products, each row will be a different product, contains product name, product information, product description. This is important because not only does it have ingredients for the recipes but it also contains products for the cookware as well.

- Product ID
- Product name
- Product info
- Photo ID

Recipe_ingredient

- Recipe_ingredient-ID
- Recipe ID-REFERENCE-Recipe-Recipe ID
- Product ID-REFERENCE-Product-Product ID

Post-info about the post, who posted it, when they posted it but also a timestamp of when they post it so users can see if the recipe is up to date or is an older recipe.

- Post ID
- User ID-REFERENCE-User-User ID

Timestamp

Occasions-info about occasions created, user who created the occasion and the recipe they are referencing back to for that particular occasion

Occasions ID

User ID-REFERENCE-User-User ID

Public Recipe ID-REFERENCE-Public Recipe- Public Recipe ID

Private Recipe ID-REFERENCE-Private Recipe-Private Recipe ID

Products ID -REFERENCE-Products-Product ID

Post ID-REFERENCE-Post-Post ID

Profile-info about user, their profile photo

Profile ID

Photo ID-REFERENCE-Photo-Photo ID

User ID-REFERENCE User-User ID

Photos-relate to the user, photo of the products, photo of the dish, comments

Photo ID

User ID-REFERENCE-User-User ID

Comments

Comment ID

User ID-REFERENCE-User-User ID

Timestamp

```
CREATE TABLE user(
User_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) NOT NULL,
password VARCHAR(100) NOT NULL,
location VARCHAR (50) NOT NULL,
birthday DATE NOT NULL
);

CREATE TABLE recipe(
recipe_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user-user_id,
product_id-REFERENCE-products-product_id,
photo_id-REFERENCE-photo-photo_id,
post_id-REFERENCE-post-post_id
);

CREATE TABLE recipe_ingredient(
recipe_ingredient_id SERIAL PRIMARY KEY,
recipe_id-REFERENCE-recipe-recipe_id,
product_id-REFERENCE-products-product_id,
photo_id-REFERENCE-photos-photo_id
);

CREATE TABLE grocery_list(
grocery_list_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user - user_id,
product_id-REFERENCE-products-product_id,
quantity INTEGER NOT NULL,
recipe_id-REFERENCE-recipe-recipe_id,
timestamp TIMESTAMP NOT NULL
);

CREATE TABLE follow (
follower_id-REFERENCE-user-user_id
following_id-REFERENCE-user-user_id
);

CREATE TABLE photos(
photo_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user-user_id
);

CREATE TABLE products(
product_id SERIAL PRIMARY KEY,
product_name VARCHAR(100) NOT NULL,
product_info        VARCHAR(255) NOT NULL,
photo_id-REFERENCE-photos-photo_id
);
```

```
CREATE TABLE post (
post_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user-user_id,
timestamp  TIMESTAMP
);

CREATE TABLE occasions (
occasions_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user-user_id,
post_id-REFERENCE-post-post_id
);

CREATE TABLE profile(
profile_id SERIAL PRIMARY KEY,
photo_id -REFERENCE-photo-photo_id,
user_id -REFERENCE- user-user_id
);

CREATE TABLE comments(
comment_id SERIAL PRIMARY KEY,
user_id-REFERENCE-user-user_id
timestamp TIMESTAMP
);
```