

Data Modeling Brainstorming for grocery list app

- users can sign into the app with their email and password
- users can create recipes with ingredients and instructions
- recipes can be marked as public or private
- users can view other people's recipes
- ingredients from recipes can be added to user's grocery lists
- users can create their own occasions and assign recipes to occasions

Tables:

Users	Posts	Recipes	Products
user_id SERIAL PRIMARY KEY	Post_id SERIAL PRIMARY KEY	Recipes_id SERIAL PRIMARY KEY	Recipes_id SERIAL PRIMARY KEY
email	user_id INTEGER NOT NULL REFERENCES users(user_id)	user_id INTEGER NOT NULL REFERENCES users(user_id)	reviews
password	type (image or text)	followers_id INTEGER NOT NULL REFERENCES posts(post_id)	user_id INTEGER NOT NULL REFERENCES users(user_id)
dob	followers_id INTEGER NOT NULL REFERENCES users(followers_id)	ingredients	
username		type (image or text)	
location		cooking_time	
		grocery_list	

special_occasions	special_occasions_list (Associate table)
special_occasions SERIAL PRIMARY KEY	special_occasions_list SERIAL PRIMARY KEY
locations	user_id INTEGER NOT NULL REFERENCES users(user_id)
dates	
user_id INTEGER NOT NULL REFERENCES	Recipes_id INTEGER NOT NULL REFERENCES

users(user_id)	posts(Recipes_id)
	Special_occasions_list INTEGER NOT NULL REFERENCES posts(Special_occasions_list)

3 Relationships:

(Many to many) A post can have many users and users can have many post

(One to many) A user can have many special occasions and a special occasion can belong to a user

(One to many) A user can have many recipes and a recipe can belong to a user.

- **explain why you chose that relationship.**

I chose the relationship where a user can have a lot of posts and vice-versa.

I chose one to many because relationship because a special occasion and recipes column should belong to that individual user it wouldn't have made sense to use the many to many relationship.

Part 2 Tables:

relationships	columns
relationships SERIAL PRIMARY KEY	columns SERIAL PRIMARY KEY
followers=VARCHAR(n)	Special_occasions_list INTEGER NOT NULL REFERENCES posts(Special_occasions_list)
user_id INTEGER NOT NULL REFERENCES users(user_id)	Recipes_id INTEGER NOT NULL REFERENCES posts(Recipes_id)
recipes_id INTEGER NOT NULL REFERENCES users(user_id)	user_id INTEGER NOT NULL REFERENCES users(user_id)

- why you'll be storing that data

Because the data i chose relates is to the user in some way, the user might have a follower the user will need recipes for the app

- and why you chose the data type you did

I choose varchar because of the you can throw in strings number etc

Tables:

```
CREATE TABLE user(  
  user_id SERIAL PRIMARY KEY,  
  user_name VARCHAR(40),  
  name VARCHAR(40),  
  email VARCHAR(40),  
  password VARCHAR(20),  
  dob date  
);
```

```
INSERT INTO user(user_name, name, password, date_of_birth)  
VALUES ('ddbad', 'diana', 'secret', 10-15-1962);
```

Tables:

```
CREATE TABLE post(  
  post SERIAL PRIMARY KEY,  
  type VARCHAR(40),  
);
```

```
INSERT INTO post(type)  
VALUES ('I like cats');
```

Tables:

```
CREATE TABLE recipes(  
  Recipes_id SERIAL PRIMARY KEY,  
  ingredients VARCHAR(40),  
  type VARCHAR(40),  
  cooking_time INTEGER(25),  
  grocery_lists VARCHAR(200)  
);
```

```
INSERT INTO recipes(ingredients, type, cooking_time, grocery_lists)  
VALUES ('onion and flour', 'onion rosepedal', '30', 'onion, flour, cooking oil, seasonings');
```

Tables:

```
CREATE TABLE special_occasion(  
  special_occasion SERIAL PRIMARY KEY,
```