Data Modeling Lab

STEP 1

Brainstorm

- Sign in: users can sign into the app with their email and password.
    - Sign in with email and password
- Users can have a “library” of collected recipes.
    - Each recipe will have a list of ingredients and cooking instructions
- Users can have a grocery list to buy the ingredients that they need
- Users can can put their recipe as either private or public
- Users can have a board of events that they can add recipes to for that occasion(like pinterest)
- Users can view other users public recipes
- Users can search for recipes
- Users can post recipes
- Users can search recipes by name or type (appetizers, desserts, dinner, lunch, breakfast, cook time)
- Users can review recipes
    - Users can add comments to recipe reviews

Table Ideas

User: User ID(PK), email, password
Ingredients: ingredient _id, ingredient_name, ingredient_amount.
Recipes: recipe-_id, title, ingredient_id, ingredient_amount, instructions, User ID(foreign key), private(boolean), cook_time
Groceries: list_id, ingredient id, User ID(foreign key)

Category: category_id, recipe_id, category type
Board: board_id, name, recipe_id, User ID(foreign key)
Reviews: review_id, body, rating, User ID(foreign key)
Comments: comment_id, review_id, body, User ID(foreign key)

Relationships

One-to-One	One-to-Many	Many-to-Many
Grocery	User	Recipe table
Category	board	Ingredients table
	review	Comments

STEP 2

Columns

User sign table:
Columns & datatype:

- User ID(PK) serial
- email varchar
- password varchar

Why store data: To track user
Why datatype: User ID should be serial PK so that it increments and it's a PK. Email and password are varchar because they could include numbers and letters

Ingredients table:
Columns & data types:

- ingredient_id int
- ingredient_name varchar
- ingredient_amount int

Why store data: ingredients should be stored in order to pull them into grocery lists
Why data type: ingredient id should be unique to the table, ingredient name should be entered as text and amount is int because it should be entered as a number

Recipe Table:
Columns & datatype:

- ID serial
- title varchar
- ingredients_id int
- ingredient_amount int

- instructions varchar
- userID int
- private boolean
- cook_time integer

Why store data: To track the recipes that are created and that we will create
Why datatype: *Recipe ID* should be serial PK so that it increments and it's a PK. *Title* and *Instructions* are varchar because they could include numbers and letters *Ingredients ID* is an Int because it is auto-generated and referenced in other tables, *Ingredient amount* is an int because it is a number, *UserID* is int because it is referenced in the user table and doesn't have to be generated/incremented again, Private/public is boolean because there can only be two options & cook time is an in because it should be entered as a number.

Grocery List:
Columns & datatypes:

- list_id serial
- user_id int
- ingredient_id int

Why store data: to make a grocery list that users can add ingredients to.
Why data type: ID is serial so each one can have a unique ID to refer to. User ID is integer so that each user continues to have a unique ID. Ingredient ID is int so each different ingredient will have a unique ID.

Category:
Columns & datatypes:

- category_id serial
- recipe_id integer
- type varchar

Why store data: so that recipes can be searched by category
Why data type: Category id needs to be a unique id for the table. Recipe ID is referenced from the recipe table and category type is varchar because it should be letters

Board:
Columns & data types:

- Id serial
- user ID int
- Name varchar
- recipe id int

Why store data: each user can have a board for different events they can save recipes to.
Why data type: board id is serial so it can generate a new ID for each board made. User ID is integer so it can refer back to the user table. Name is varchar so you can add what you want to name the board. Recipe ID is int so it can refer back to a pre made recipe.

Reviews:
Columns & data types:

- ID serial
- user id int
- body varchar
- rating int

Why store data: reviews will be stored so that users can see how recipes are rated
Why data type: Review ID has to be unique to the table, user ID is referenced in the user table, body should be varchar so that numbers and letters can be used and the rating is int because users should enter it as a number

Comments:
Columns & data type:

- Id serial
- review id int
- Body varchar
- userID int

Why store data: Comments will be stored so that other people can view them and not just the person who types it.
Why data type: the comment ID is serial so it can generate a unique ID for each comment. Review ID is integer so it can pull the unique ID from the review table. Body is varchar so the user can type in what they want for the comment. User ID is integer so it can pull the users unique ID from the users table.

PART 3

```
CREATE TABLE users(
  user_id SERIAL PRIMARY KEY,
  email VARCHAR(255),
  password VARCHAR (255)
```

```
  );

CREATE TABLE ingredients(
  ingredient_id SERIAL PRIMARY KEY,
  name VARCHAR(1000),
  amount INT
  );

CREATE TABLE recipe(
  recipe_id SERIAL PRIMARY KEY,
  title VARCHAR(1000),
  user_id INT REFERENCES users(user_id),
  ingredient_id INT REFERENCES ingredients(ingredient_id),
  ingredient_amount INT,
  instructions VARCHAR(10000),
  private BOOLEAN DEFAULT False,
  cook_time INT
  );

    CREATE TABLE category(
    category_id SERIAL PRIMARY KEY,
    recipe_id INT REFERENCES recipe(recipe_id),
    type VARCHAR (100)
    );

CREATE TABLE board(
```

```
  board_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(user_id),
  board_name VARCHAR(1000),
  recipe_id INT REFERENCES recipe(recipe_id)
  );

CREATE TABLE groceries(
  list_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(user_id),
  ingredient_id INT REFERENCES ingredients(ingredient_id)
  );

CREATE TABLE reviews(
  review_id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(user_id),
  review VARCHAR(10000)
  );

CREATE TABLE comments(
  comment_id SERIAL PRIMARY KEY,
  review_id INT REFERENCES reviews(review_id),
  user_id INT REFERENCES users(user_id)
  );
```

INTERMEDIATE

```
INSERT INTO users(email, password)
```

```
VALUES
        ('123@hotmail.com', 12345),
  ('lilguy83@gmail', 'abc123'),
  ('hottamale@gmail', 'mydogiscool');


INSERT INTO ingredients(name, amount)
VALUES
        ('velveeta', 1),
        ('macaroni', 3),
        ('tuna', 2),
        ('water', 1),
        ('broccoli', 4),
        ('salt', 1);
```