Summary

We want to create a recipe creating/sharing a grocery list app. You'll be planning out what tables we'll need, what information they'll store, and how the data will relate to each other.

Features

- Users can sign into the app with their email and password
  - Users table with columns email and password.
- Users can create recipes with ingredients and instructions
  - (One to Many) User can create multiple recipes, each recipe linked to 1 user
  - (Many to Many) Each recipe can have multiple ingredients, ingredients can be linked to multiple recipes
- Recipes can be marked as public or private
  - Extra column in Recipes table, data type BOOLEAN
- Users can view other people's recipes
  - SELECT recipes linked to that specific user_id?
- Ingredients from recipes can be added to user's grocery lists
  - (Many to Many) ingredients can be apart of multiple grocery lists and multiple grocery lists can use the same ingredients
- Users can create their own occasions and assign recipes to occasions
  - (One to Many) User can have multiple occasions but occasions are assigned to one user

Part 1 - Step 1 - Brainstorming

- *Need to keep track of:*
  - user_id (email, password)
  - recipe_id (user_id foreign key, instructions TEXT, public/private BOOLEAN)
  - ingredients_id (ingredient)
  - recipe_ingredients_id (recipe_id foreign key, ingredients_id foreign key)
  - grocery_list_id (user_id foreign key, recipe_id foreign key)
  - Grocery_list_ingredients_id (grocery_list_id foreign key, ingredients_id foreign key, ingredient)
  - occasions_id (user_id foreign key, recipe_id foreign key)

- *High Level needs to keep track of*
  - User account id
  - User email
  - User password encryption
  - If a user exists with an email
  - Account sign up for a new user
  - Creation of individual recipes

- Creation of grocery lists
- Previously saved recipes
- Previously saved grocery lists
- Individual ingredients
- Individual grocery list items
- Quantity of grocery items (ingredients)
- Image of ingredients?
- Brand of ingredient/items
- Status of grocery item (checked or unchecked) (optional)

Part 1 - Step 2 - Table Ideas

- *Users*
  - This table will hold information about every user account in the app
  - Each row will represent an individual user
- *Recipes*
  - This table will hold information about every recipe created by a specific user
  - Each row will be an individual recipe
- *Ingredients*
  - This table hold information about every different ingredient
  - Each row will be an individual ingredient
- *RecipesIngredients Table (Association Table)*
  - This table will associate which ingredients are used in each recipe
  - Each row will be 1 ingredient associated with 1 recipe
    - These can be grouped/ordered by recipe_id?
- *Grocery Lists*
  - This table will hold information about every grocery list created by a specific user
  - Each row will be an individual grocery list
- *GroceryListsIngredients (Association Table)*
  - This table will associate which ingredients are included in a specific grocery list
  - Each row will be an individual grocery list ingredient
- *Occasions*
  - This table will hold information about every occasion created by specific users
  - Each row will represent an occasion created by a specific user
- *OccasionsRecipes (Association Table)*
  - This table will associate which recipes are included in each occasion
  - Each row will be 1 recipe associated with 1 occasion

Part 1 - Step 3 - Relationships

- One-to-One:
  - N/a because we have no special cases/types
- One-to-Many:
  - Users table and Recipes table because individual users can create multiple recipes, but a recipe is only linked with one specific user

  - Each individual user (from Users Table) can create many occasions in the Occasion Table.
  - Each individual user (from Users Table) can create many grocery lists in the Grocery Lists Table.

- Many-to-Many:
  - RecipeIngredients Table is an Association Table between the Recipe Table and Ingredients Table.
  - GroceryListsIngredients Table is an Association Table between the Grocery Lists Table and Ingredients Table.
  - OccasionsRecipes Table is an Association Table between the Occasions Table and Recipes Table.

Part 2 - Step 1 - PDF from DB Designer

- Michael's pdf
  - Organized for easy reference arrow readability

Part 2 - Step 2 - Columns

- Users Table
  - user_id (type SERIAL PRIMARY KEY)
    - Primary key for the table
    - Important for identifying specific users
  - username (type VARCHAR(60))
    - Characters selected for the user's username will have a limit of 30
    - Will be useful for logging in + to display on public recipes
  - email (type VARCHAR(100))
    - Characters selected for the user's email will have a limit of 100
    - Will be useful for if they forgot their password and need to reset it + if they want to subscribe for weekly emails about the hottest new recipes.
  - password (type VARCHAR (60))
    - Characters selected for the user's password will have a limit of 100
    - Will be useful for verifying login credentials, so only that user will be able to log in to that account.
- Recipes Table
  - recipe_id (type SERIAL PRIMARY KEY)
    - The primary key for the table
    - Important for identifying specific recipes
  - user_id (type INT FOREIGN KEY)
    - Data type being referenced here is of type INT
    - A reference to the specific user who created the recipe

    - recipe_name (type VARCHAR(60)
        - Characters selected for the recipe_name will have a limit of 60
        - Name of the recipe the ingredient are in
    - ~~username (type VARCHAR(60)~~REDUNDANT
        - ~~Data type being referenced here is of type VARCHAR(60)~~
        - ~~References the username of the recipe creator~~
    - instructions (type TEXT)
        - This data type imposes no character limit for the recipe instructions
        - The step by step instructions for preparing the recipe
    - public (type BOOLEAN with a default of FALSE)
        - Recipes can be kept private or shared publicly, can keep track of this with a BOOLEAN
        - By default, recipes are kept private. But if a user would like to share their recipe publicly, they can hit a button to update the Boolean here to True
- Ingredients Table
    - ingredient_id (type SERIAL PRIMARY KEY)
        - Primary key for the table
        - Important for identifying specific ingredients
    - ingredient_name (type VARCHAR(30))
        - Characters selected for the ingredient_name will have a limit of 30
        - The name of the ingredient
- RecipesIngredients Table
    - recipes_ingredients_id (type SERIAL PRIMARY KEY)
        - Primary Key for the table
        - Ensures uniqueness
    - recipe_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - Reference id number to the specific recipe
    - ingredients_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - Reference id number to the specific ingredient
    - ~~ingredient_name (type VARCHAR(30)~~REDUNDANT
        - ~~Characters selected for the ingredient_name will have a limit of 30~~
        - ~~Name of the ingredient used in the recipe~~
- Grocery Lists Table
    - grocery_list_id (type SERIAL PRIMARY KEY)
        - Primary key for the table
        - Ensures uniqueness
    - grocery_list_name (type VARCHAR(60))
        - Characters selected for the grocery_list_name will have a limit of 60
        - Name of the specific grocery list
    - user_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to the specific user who created each grocery list

    - recipe_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - Reference id number to the specific recipe
- GroceryListsIngredients Table
    - grocery_lists_ingredients_id (type SERIAL PRIMARY KEY)
        - Primary key for the table
        - Ensures uniqueness
    - grocery_list_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to the specific id for each grocery list, which will be repeated for each ingredient to be added to the grocery list
    - ingredients_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to the specific id for each ingredient to be added to any grocery list
    - ~~ingredient_name (type VARCHAR(30) FOREIGN KEY)~~ REDUNDANT
        - ~~Data type being referenced here is of type VARCHAR(30)~~
        - ~~A reference to the ingredient_name associated with the specific ingredients_id~~
- Occasions Table
    - occasions_id (type SERIAL PRIMARY KEY)
        - Primary key for the table
        - Ensures uniqueness
    - occasion_name (type VARCHAR(60))
        - Characters selected for the occasion_name will have a limit of 60
        - Name of the specific occasion
    - user_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to the specific user who created each occasion
- OccasionsRecipes Table
    - occasions_recipes_id (type SERIAL PRIMARY KEY)
        - Primary key for the table
        - Ensures uniqueness
    - occasions_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to the specific id for each occasion, which will be repeated for each recipe to be added to the occasion
    - recipe_id (type INT FOREIGN KEY)
        - Data type being referenced here is of type INT
        - A reference to each recipe_id to be added to a specific occasion_id

Part 3 - Create Table Statements in SQL

- Users Table
    - CREATE TABLE users(user_id SERIAL PRIMARY KEY, username VARCHAR(60), email VARCHAR(100), password VARCHAR(60));

- Recipes Table
    - CREATE TABLE Recipes(recipe_id SERIAL PRIMARY KEY, user_id INTEGER NOT NULL REFERENCES users(user_id), recipe_name VARCHAR(60), instructions TEXT, public BOOLEAN DEFAULT False);
- Ingredients Table
    - CREATE TABLE ingredients(ingredients_id SERIAL PRIMARY KEY, ingredients_name VARCHAR(30));
- RecipesIngredients Table
    - CREATE TABLE recipesIngredients(recipes_ingredients_id SERIAL PRIMARY KEY, recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id), ingredients_id INTEGER NOT NULL REFERENCES ingredients(ingredients_id));
- Grocery Lists Table
    - CREATE TABLE groceryList(grocery_list_id SERIAL PRIMARY KEY, grocery_list_name VARCHAR(60), user_id INTEGER NOT NULL REFERENCES users(user_id), recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id));
- GroceryListsIngredients Table
    - CREATE TABLE groceryListsIngredients(grocery_lists_ingredients_id SERIAL PRIMARY KEY, grocery_list_id INTEGER NOT NULL REFERENCES groceryList(grocery_list_id), ingredients_id INTEGER NOT NULL REFERENCES ingredients(ingredients_id));
- Occasions Table
    - CREATE TABLE occasions(occasions_id SERIAL PRIMARY KEY, occasion_name VARCHAR(60), user_id INTEGER NOT NULL REFERENCES users(user_id));
- OccasionsRecipes Table
    - CREATE TABLE occasionsRecipes(occasions_recipes_id SERIAL PRIMARY KEY, occasions_id INTEGER NOT NULL REFERENCES occasions(occasions_id), recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id));

Intermediate - Inserting Data into our Tables

-