Nelson Moreno 3/29/2022 DATA MODELING LAB

Brainstorming Section

usergrocerymain

userID int

username varchar(50)

userpassword varchar(50)

recipespublic Boolean

useringredients text

userInstructions text

userFollowingID integer

userOccasions

Brain storming section END

Table Ideas

NOTE: ALL PRIMARY AND FOREIGN KEYS WILL HAVE TYPE INTEGER

UserGroceryMain Table (One to many table)

This table will hold information about the users and each row is a separate user

name varchar

holds name information

password varchar

email varchar

holds email information

password varchar

holds password information

recipePublic boolean

lets the user choose whenether they want their information public (TRUE) or private (FALSE)

userFollowingID

Contains information of a user another user may be following

GroceryIngredients Table (Many-to-many table)

This table sustains a many to many relationship between ingredients and a users grocery list

grocerIngredientsID

Primary Key

groceryID

Foreign key

ingredientsID

Foreign Key

Ingredients Table

Ingredients table: this table will hold unique rows about different ingredients needed for a recipe

ingredientsID

Primary key for ingredients

ingredientAmount float

Self-explanatory

ingredientName varchar

Self-explanatory

ingredientPrice float

Self-explanatory

Groceries Table (Many-to Many relationship table)

Groceries: This table will hold information about the groceries people need to create ingredients each row will be an individual product

userGrocerID

primary key that is an integer

groceriesID

foreign key that is creating a one to many relationship between user and grocery list

groceryItemName

Self-explanatory

userSavedRecipes (Many-to Many Table)

Recipe: This table will hold information about the different recipes people will have listed. Each row will have a unique recipe

RecipeID

Primary Key for recipes

usergrocerID

Recipe created by specific user

ingredientsID

Specific ingredient needed by this recipe

recipeName

Self-explanatory

instructionsText

Self-explanatory

Occasions table (One to many table)

Each row is separate event that the user will or has attended with the specific recipe

OccasionsID

Primary key for the Occasions table

eventType

Details exact kind of event (Wedding, Birthday party, Sweet 16, Cinco de Mayo)

eventDate

Self-explanatory

userGrocerID

Creates one to many relationship; One grocer can have many events

Relationships

Recipes table is linked to the Ingredients table and it is also linked to the Users table

Recipes can have many ingredients and Users can have many recipes Many-to-many

Users can have multiple recipes One-to-many

Recipes can have multiple users

One user can have many occasions – One to many

One user can have many grocery lists – One to many

GroceryIngredients is a junction table that links the ingredients to a grocery list that the user will use to shop at the supermarket

Create Tables in SQL Code

```
create table userGroceryMain (
        userGrocerID SERIAL PRIMARY KEY,
        name VARCHAR(55),
                email VARCHAR(55),
                password VARCHAR(55),
                recipePublic boolean
);
CREATE TABLE groceries (
 groceriesID SERIAL PRIMARY KEY,
 usergrocerID INT REFERENCES userGroceryMain(userGrocerID),
 groceryItemName VarChar(255)
 );
CREATE TABLE occasions (
 occasionsID SERIAL PRIMARY KEY,
 eventType VARCHAR(255),
 eventDate VARCHAR(255),
 userGroceryID INT REFERENCES userGroceryMain(userGroceryID)
 );

CREATE TABLE ingredients (
 ingredientsID SERIAL PRIMARY KEY,
 ingredientAmount FLOAT,
 ingredientName VARCHAR(255),
 ingredientPrice FLOAT
 );
CREATE TABLE groceryingredients (
        groceryIngredientsID SERIAL PRIMARY KEY,
        groceriesID INT REFERENCES groceries(groceriesID),
```

```
        ingredientsID INT REFERENCES ingredients(ingredientsID),
);
CREATE TABLE userSavedRecipies (
        recipeID SERIAL PRIMARY KEY,
        ingredientsID INT REFERENCES ingredients(ingredientsID),
        userGrocerID INT REFERENCES userGroceryMain(userGrocerID),
        recipeName VARCHAR(255),
        instructionsText text
);
```