

1. ER modeling procedure based on requirements derived from Stage I documents.

- Similar to Stage 1, this covers the nutritional bases of sugars, fats, carbs, proteins, and calories. This also has a special distinction between gender when calculating the preferred caloric intake, which was not taken into account in my Stage 1 Document.

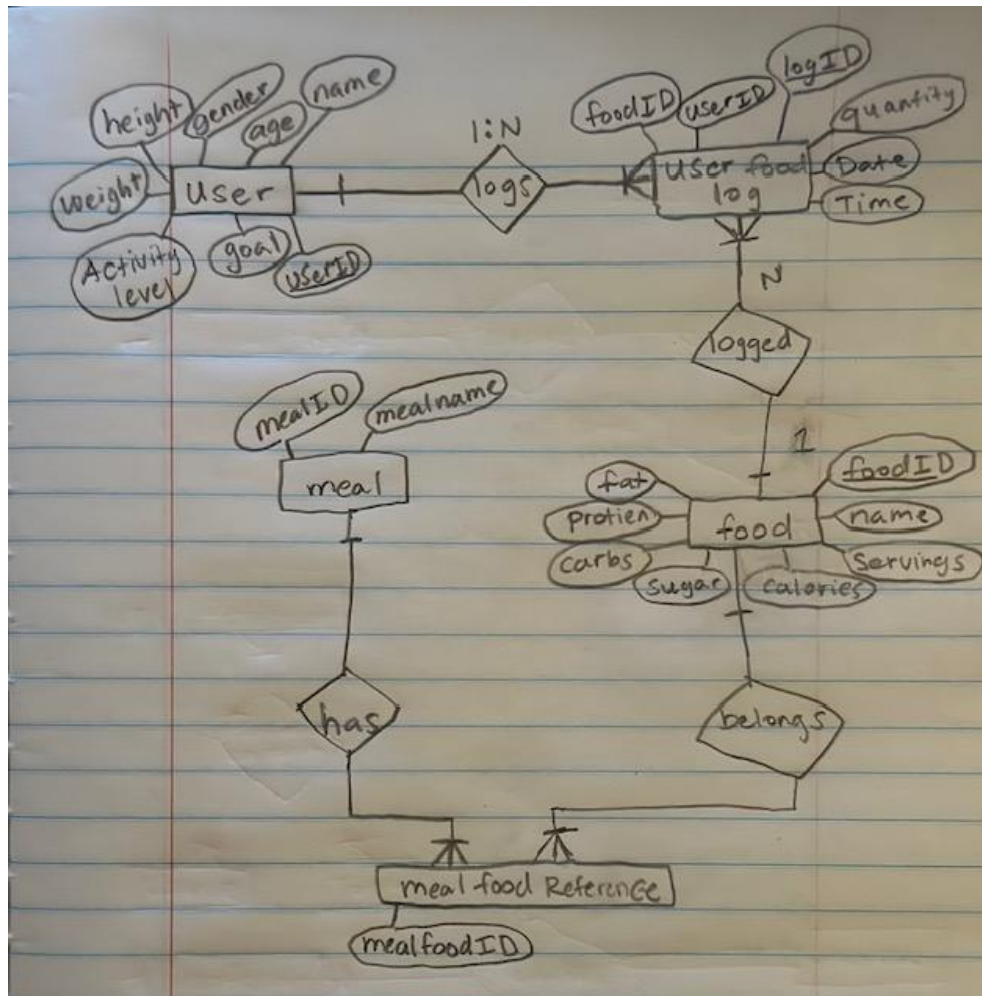
Entities:

1. User Table with attributes as below
 - a. UserID - primary key
 - b. Username
 - c. Age
 - d. Gender
 - e. Height
 - f. Weight
 - g. ActivityLevel - Sedentary, lightly active, Active, Very active
 - h. Goal - Lose weight, maintain weight, gain weight
2. Food Table with attributes as below
 - a. FoodID - primary key
 - b. Foodname
 - c. Servingsize
 - d. Calories
 - e. Carbs
 - f. Fat
 - g. Protein
 - h. Sugar
3. Meal Table with attributes
 - a. MealID
 - b. Mealname
4. UserFoodLog Table with attributes
 - a. LogID
 - b. UserID
 - c. FoodID
 - d. Date
 - e. Time
 - f. Quantity
5. MealFoodReference Table with attributes
 - a. MealFoodID
 - b. MealID
 - c. FoodID

Relationships:

1. User-UserFoodLog - (1:M) user will have multiple food logs
2. Food-UserFoodLog - (1:M) food item can be logged into multiple food logs
3. Meal-Food - (M:M) - A meal can have multiple food items and a Food can be part of multiple meals and the reason behind creating MealFoodReference.

2. Complete ER diagram



3. Translation to relational tables, written in DDL

- a. CREATE TABLE User (
UserID INT PRIMARY KEY,
Username VARCHAR(255),
Age INT,
Gender VARCHAR(10),
Height DECIMAL(5,2), -- height is in cm with 2 decimal places
Weight DECIMAL(5,2), -- weight is in kg with 2 decimal places
ActivityLevel VARCHAR(20),

Goal VARCHAR(20));

- b. CREATE TABLE Food (
 FoodID INT PRIMARY KEY,
 FoodName VARCHAR(255),
 ServingSize DECIMAL(5,2), -- serving size is in grams with 2 decimal places
 Calories INT,
 Carbs DECIMAL(5,2), -- carbs are in grams with 2 decimal places
 Fat DECIMAL(5,2), -- fat is in grams with 2 decimal places
 Protein DECIMAL(5,2), -- protein is in grams with 2 decimal places
 Sugar DECIMAL(5,2)); -- sugar is in grams with 2 decimal places
- c. CREATE TABLE Meal (
 MealID INT PRIMARY KEY,
 MealName VARCHAR(255));
- d. CREATE TABLE UserFoodLog (
 LogID INT PRIMARY KEY,
 UserID INT,
 FoodID INT,
 LogDate DATE,
 LogTime TIME,
 Quantity DECIMAL(8,2), -- quantity is a decimal value
 FOREIGN KEY (UserID) REFERENCES User(UserID),
 FOREIGN KEY (FoodID) REFERENCES Food(FoodID));
- e. CREATE TABLE MealFoodReference (
 MealFoodID INT PRIMARY KEY,
 MealID INT,
 FoodID INT,
 FOREIGN KEY (MealID) REFERENCES Meal(MealID),
 FOREIGN KEY (FoodID) REFERENCES Food(FoodID));

4. Schema refinement to BCNF

- a. User Table: - No non-trivial functional dependencies (FDs) exist other than the UserID being a primary key, so it is in BCNF.
- b. Food Table: - No non-trivial FDs exist other than the FoodID being a primary key, so it is in BCNF.
- c. Meal Table: - No non-trivial FDs exist other than the MealID being a primary key, so it is in BCNF.
- d. UserFoodLog Table: - The table relates UserID and FoodID with the LogID, LogDate, LogTime, and Quantity. There are no non-trivial FDs other than the primary key LogID.
- e. MealFoodReference Table: - The table relates MealID and FoodID with the MealFoodID. There are no non-trivial FDs other than the primary key MealFoodID.

The tables are already in BCNF since there are no non-trivial FDs beyond the primary keys in each table.

5. Supporting SQL queries for all proposed application functionalities.

Queries to calculate the number of calories to consume in order to gain weight or lose weight or maintain weight. And calculate consumed calories in order to inform about the number of calories left from the calories required for the day

- a.

```
SELECT
CASE WHEN Gender = 'Male' AND ActivityLevel = 'Sedentary' THEN 1.2 * ((10
* Weight) + (6.25 * Height) - (5 * Age) + 5)
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Sedentary' THEN 1.2 *
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Lightly Active' THEN 1.375 *
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Lightly Active' THEN 1.375
* ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Moderately Active' THEN
1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Moderately Active' THEN
1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Very Active' THEN 1.725 *
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)
```

```
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Very Active' THEN 1.725 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)  
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Extra Active' THEN 1.9 * ((10  
* Weight) + (6.25 * Height) - (5 * Age) + 5)  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Extra Active' THEN 1.9 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)  
END AS Calories_To_Maintainweight  
FROM User  
WHERE UserID = input_UserID;
```

– Check how many calories are left to eat

```
SELECT Calories_To_Maintainweight - (UFL.Quantity * F.Calories)  
FROM UserFoodLog AS USF  
JOIN Food AS F ON F.FoodID = USF.FoodID  
WHERE USF.UserID = input_UserID  
AND USF.Date = Current_Date;
```

b. SELECT

```
CASE WHEN Gender = 'Male' AND ActivityLevel = 'Sedentary' THEN (1.2 * ((10  
* Weight) + (6.25 * Height) - (5 * Age) + 5)) - 500  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Sedentary' THEN (1.2 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) - 500  
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Lightly Active' THEN (1.375  
* ((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) - 500  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Lightly Active' THEN  
(1.375 * ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)) - 500  
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Moderately Active' THEN  
(1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) - 500  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Moderately Active' THEN  
(1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) - 500  
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Very Active' THEN (1.725 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) - 500  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Very Active' THEN (1.725  
* ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) - 500  
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Extra Active' THEN (1.9 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) - 500  
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Extra Active' THEN (1.9 *  
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) - 500  
END AS Calories_To_Loseweight
```

```
FROM User
WHERE UserID = input_UserID;
```

– Check how many calories are left to eat

```
SELECT Calories_To_Loseweight - (UFL.Quantity * F.Calories)
FROM UserFoodLog AS USF
JOIN Food AS F ON F.FoodID = USF.FoodID
WHERE USF.UserID = input_UserID
      AND USF.Date = Current_Date;
```

a. SELECT

```
CASE WHEN Gender = 'Male' AND ActivityLevel = 'Sedentary' THEN (1.2 * ((10
* Weight) + (6.25 * Height) - (5 * Age) + 5)) + 500
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Sedentary' THEN (1.2 *
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) + 500
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Lightly Active' THEN (1.375
* ((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) + 500
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Lightly Active' THEN
(1.375 * ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161)) + 500
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Moderately Active' THEN
(1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) + 500
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Moderately Active' THEN
(1.55 * ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) + 500
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Very Active' THEN (1.725 *
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) + 500
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Very Active' THEN (1.725
* ((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) + 500
ELSE WHEN Gender = 'Male' AND ActivityLevel = 'Extra Active' THEN (1.9 *
((10 * Weight) + (6.25 * Height) - (5 * Age) + 5)) + 500
ELSE WHEN Gender = 'Female' AND ActivityLevel = 'Extra Active' THEN (1.9 *
((10 * Weight) + (6.25 * Height) - (5 * Age) - 161) ) + 500
END AS Calories_To_Gainweight
FROM User
WHERE UserID = input_UserID;
```

– Check how many calories are left to eat

```
SELECT Calories_To_Gainweight - (UFL.Quantity * F.Calories)
```

Neeraj Thokala
CMPSC 431W

```
FROM UserFoodLog AS USF
JOIN Food AS F ON F.FoodID = USF.FoodID
WHERE USF.UserID = input_UserID
      AND USF.Date = Current_Date;
```