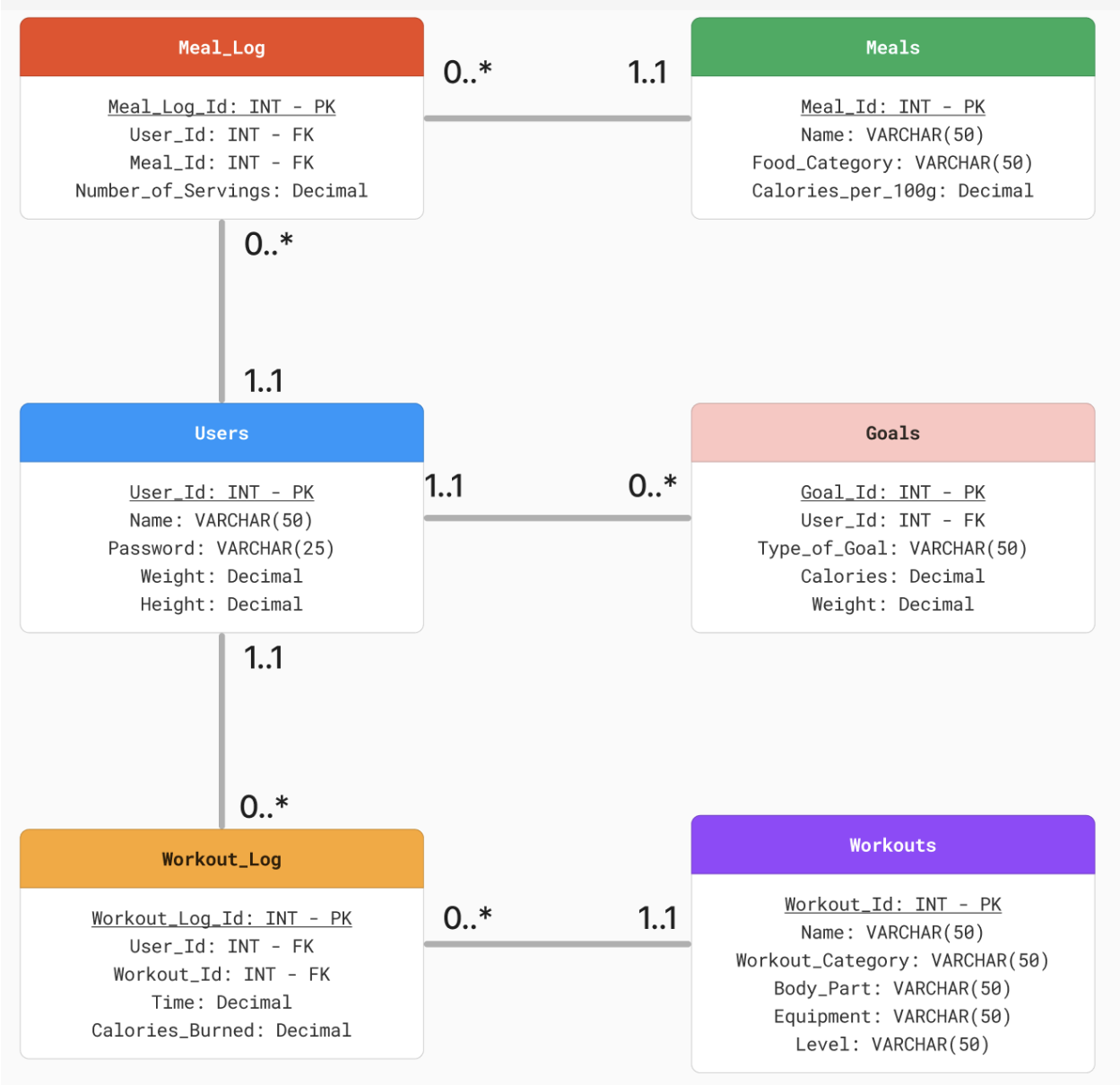
UML Diagram


Meal_Log
Meal_Log_Id: INT - PK
User_Id: INT - FK
Meal_Id: INT - FK
Number_of_Servings: Decimal
0..*
1..1
Meals
Meal_Id: INT - PK
Name: VARCHAR(50)
Food_Category: VARCHAR(50)
Calories_per_100g: Decimal
0..*
1..1
Users
User_Id: INT - PK
Name: VARCHAR(50)
Password: VARCHAR(25)
Weight: Decimal
Height: Decimal
1..1
0..*
Goals
Goal_Id: INT - PK
User_Id: INT - FK
Type_of_Goal: VARCHAR(50)
Calories: Decimal
Weight: Decimal
1..1
0..*
Workout_Log
Workout_Log_Id: INT - PK
User_Id: INT - FK
Workout_Id: INT - FK
Time: Decimal
Calories_Burned: Decimal
0..*
1..1
Workouts
Workout_Id: INT - PK
Name: VARCHAR(50)
Workout_Category: VARCHAR(50)
Body_Part: VARCHAR(50)
Equipment: VARCHAR(50)
Level: VARCHAR(50)

Assumptions Made

Meals and Meal_Log: We are assuming that each meal can have multiple meal logs, but each meal log can only have one meal. Therefore, it is a one to many relationship.

Workout and Workout_Log: We are assuming that each workout can have multiple Workout_Logs, but each Workout_Log can only have one workout. Therefore, it is a one to many relationship.

Users and Meal_Log: We are assuming that each user can have multiple meal logs, but each meal log can only have one user. Therefore, it is a one to many relationship.

Workout_Log and Users: We are assuming that each user can have multiple Workout_Logs, but each Workout_Log can only have one user. Therefore, it is a one to many relationship.

Users and Goals: We are assuming that each user can have multiple goals, but each goal can only have one user. Therefore, it is a one to many relationship.

Description of Relationships:

Users: This table represents all of the information about the user including their login information. The primary key for this table is User_Id, and this is different for every unique user. This table also contains information that the user inputs about their name, password associated with their account, weight, and height. These are the four attributes contained in the Users table.

Meals: This table contains all information regarding different meal logging. The primary key for this table is the Meal_Id which is uniquely associated with each different meal. The next attributes are the Name which is simply the name of the meal. In addition the next attribute is Food_Category which simply associates the meal with the corresponding category like fast food, canned food, etc. The last attribute is Calories within each meal.

Workouts: This table contains all information regarding the workouts completed by the user. The primary key of this table is the Workout_Id. This table includes the name of the user, the Workout_Category, the Body_Part that the workout is targeting, the

equipment used, and the intensity level of the workout. The attributes are listed as Name, Workout_Category, Body_Part, Equipment, and Level.

Goals: This table contains information regarding the different goals for each user. The primary key is a unique Goal_Id. The attributes of this table are a User_Id associating each unique goal with a specific user, the Type_Of_Goal whether it is food aligned or workout aligned, the Calorie goal, and weight goal. The attributes of this table are listed as Goal_Id, User_Id, Type_of_Goal, Calories, Weight.

Meal_Logs: The meal log table contains information about each meal that the user consumes and inputs into the database. The primary key for this table is the unique Meal_Log_Id. This includes information about the type of meal and the Number_of_Servings per meal. The attributes of this table are the User_Id, Meal_Id, and Number_of_Servings.

Workout_Logs: The Workout_Log is a similar table to the meal log and contains information about every workout that the user inputs into the database. The primary key for this table is a unique Workout_Log_Id which identifies logged workout. The table also has a User_Id to associate the specific Workout_Log to a corresponding user. The next attribute is a Workout_Id which defines what workout occurred. The Time attribute logs the length of the workout. The last attribute Calories_Burned logs how many calories were burned within the workout.

3NF Normalization

Meal_Log_Id → User_Id, Meal_Id, Number_Of_Servings
User_Id → Name, Password, Weight, Height
Workout_Log_Id → User, Workout, Time, Calories
Meal_Id → Name, Food_Category, Calories_per_100g
Goal_Id → User_Id, Type_Of_Goal, Calories, Weight
Workout_Id → Name, Workout_Category, Body_Part, Equipment, Level

Create Minimal Basis

RHS Singletons:

Meal_Log_Id → User_Id
Meal_Log_Id → Meal_Id
Meal_Log_Id → Number_Of_Servings

User_Id → Name
User_Id → Password
User_Id → Weight
User_Id → Height

Workout_Log_Id → User_Id
Workout_Log_Id → Workout_Id
Workout_Log_Id → Time
Workout_Log_Id → Calories

Meal_Id → Name
Meal_Id → Food_Category
Meal_Id → Calories_per_100g

Goal_Id → User_Id
Goal_Id → Type_of_Goal
Goal_Id → Calories
Goal_Id → Weight

Workout_Id → Name

Workout_Id → Workout_Category
Workout_Id → Body_Part
Workout_Id → Equipment
Workout_Id → Level

Remove unnecessary LHS Attribute removal:

N/A

Remove inferred rules:

N/A

Minimal Basis

G = { **User_Id** → Name, **User_Id** → Password, **User_Id** → Weight, **User_Id** → Height, **Workout_Log_Id** → User, **Workout_Log_Id** → Workout, **Workout_Log_Id** → Time, **Workout_Log_Id** → Calories, **Meal_Id** → Name, **Meal_Id** → Food_Category, **Meal_Id** → Calories_per_100g, **Goal_Id** → User_Id, **Goal_Id** → Type_Of_Goal, **Goal_Id** → Calories, **Goal_Id** → Weight, **Workout_Id** → Name, **Workout_Id** → Workout_Category, **Workout_Id** → Body_Part, **Workout_Id** → Equipment, **Workout_Id** → Level }

We chose to use 3NF rather than BCNF so that we can preserve any dependencies in addition to simply reducing redundancies.

For each FD A → B in the minimal basis G uses AB as the scheme of a new relationship. If none of the schemas from Step 2 is a superkey, add another relation whose schema is a key for the original relation.

Super Key

Goal_Id, Meal_Log_Id, Workout_Log_Id can reach all the other attributes, making it a super key.

Therefore our scheme adheres to the normal form 3NF.

Relational Schema

Meal_Logs(Meal_Log_Id:INT [PK], User_Id:INT [FK to Users.User_Id], Meal_Id:INT [FK to Meals.Meal_Id, Number_of_Servings:Decimal)

Meals(Meal_Id:INT [PK], Name:VARCHAR(50), Food Category:VARCHAR(50), Calories_Per_100g:Decimal)

Users(User_Id:INT [PK], Name:VARCHAR(50), Password:VARCHAR(50), Weight:DECIMAL, Height:DECIMAL)

Goals(Goal_Id:INT[PK], User_Id:INT[FK to Users.User_Id], Type_of_Goal:VARCHAR(50), Calories:Decimal, Weight:Decimal)

Workout_Logs(Workout_Log_Id:INT [PK], User_Id:INT [FK to Users.User_Id], Workout_Id:INT [FK to Workouts.Workout_Id], Time:Decimal, Calories_Burned:Decimal)

Workouts(Workout_Id:INT [PK], Name:VARCHAR(50), Workout_Category:VARCHAR(50), Body_Part:VARCHAR(50), Equipment:VARCHAR(50), Level:VARCHAR(50))