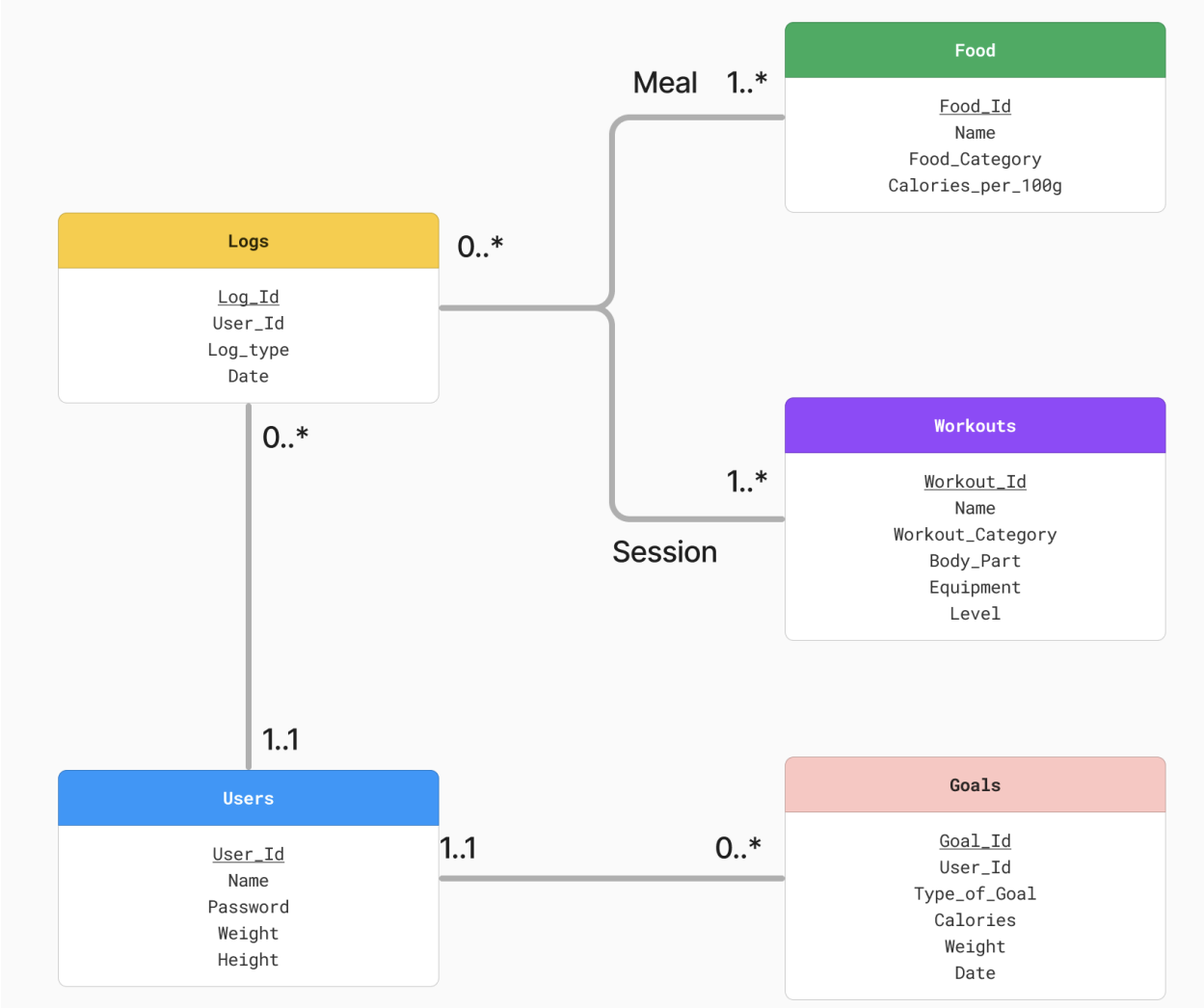
UML Diagram


Food
Food_Id
Name
Food_Category
Calories_per_100g
Meal
1..*
Logs
Log_Id
User_Id
Log_type
Date
0..*
0..*
1..*
Session
Workouts
Workout_Id
Name
Workout_Category
Body_Part
Equipment
Level
1..1
Users
User_Id
Name
Password
Weight
Height
1..1
0..*
Goals
Goal_Id
User_Id
Type_of_Goal
Calories
Weight
Date

Assumptions Made

Food and Log: We are assuming that each food item can be a part of multiple logs, and that each log item can have 1 or more food items. Therefore, it is a many to many relationship.

Workout and Log: We are assuming that each workout item can be a part of multiple logs, and that each log item can have 1 or more workout items. Therefore, it is a many to many relationship.

Users and Log: We are assuming that each user can have multiple logs, but each 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.

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

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, and date.

Logs: The log table contains information about each meal or session that the user completes and inputs into the database. The primary key for this table is the unique Log_Id. This table includes information about the user, type of log (meal or a workout), and the date that it was completed. The attributes of this table are the User_Id, Log_type, and Date.

Meal: To represent the many-many relationship between Logs and Food, this table logs the foods consumed per meal log using Meal_Id. The other attribute is Log_Id to keep track of which log records which meal is being logged, which determines the foods in the meal.

Session: To represent the many-many relationship between Logs and Workouts, this table logs the workouts per workout log using Workout_Id. The other attribute is Log_Id to keep track of which log records which workout session is being logged, which determines the workouts logged.

3NF Normalization

Log_Id → User_Id, Log_type, Date
User_Id → Name, Password, Weight, Height
Food_Id → Name, Food_Category, Calories_per_100g
Goal_Id → User_Id, Type_Of_Goal, Calories, Weight, Date
Workout_Id → Name, Workout_Category, Body_Part, Equipment, Level
(Log_Id, Food_Id) →
(Log_Id, Workout_Id) →

Create Minimal Basis

RHS Singletons:

Log_Id → User_Id
Log_Id → Log_type
Log_Id → Date

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

Food_Id → Name
Food_Id → Food_Category
Food_Id → Calories_per_100g

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

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

Log_Id →
Food_Id →

Log_Id →
Workout_Id →

Remove unnecessary LHS Attribute removal:

Removing:
Log_Id →
Food_Id →
Log_Id →
Workout_Id →

Remove inferred rules:

N/A

Minimal Basis

G = { **User_Id** → Name, **User_Id** → Password, **User_Id** → Weight, **User_Id** → Height, **Log_Id** → Log_type, **Log_Id** → Time, **Workout_Log_Id** → Calories, **Food_Id** → Name, **Food_Id** → Food_Category, **Food_Id** → Calories_per_100g, **Goal_Id** → User_Id, **Goal_Id** → Type_Of_Goal, **Goal_Id** → Calories, **Goal_Id** → Weight, **Goal_Id** → Date, **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 and Log_Id can reach all the other attributes, making it a super key.

Therefore our scheme adheres to the normal form 3NF.

Relational Schema

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

Food(Food_Id:INT [PK], Name:VARCHAR(50), Food_Category:VARCHAR(50), Calories_Per_100g:Decimal)

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

Logs(Log_Id:INT [PK], User_Id:INT [FK to Users.User_Id], Log_type:VARCHAR(50), Date:DATE)

Meal(Log_Id: INT [FK to Logs.Log_id] [PK], Food_id [FK to Food.Food_id] [PK])

Session(Log_Id: INT [FK to Logs.Log_id] [PK], Workout_id [FK to Workouts.Workout_id] [PK])

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