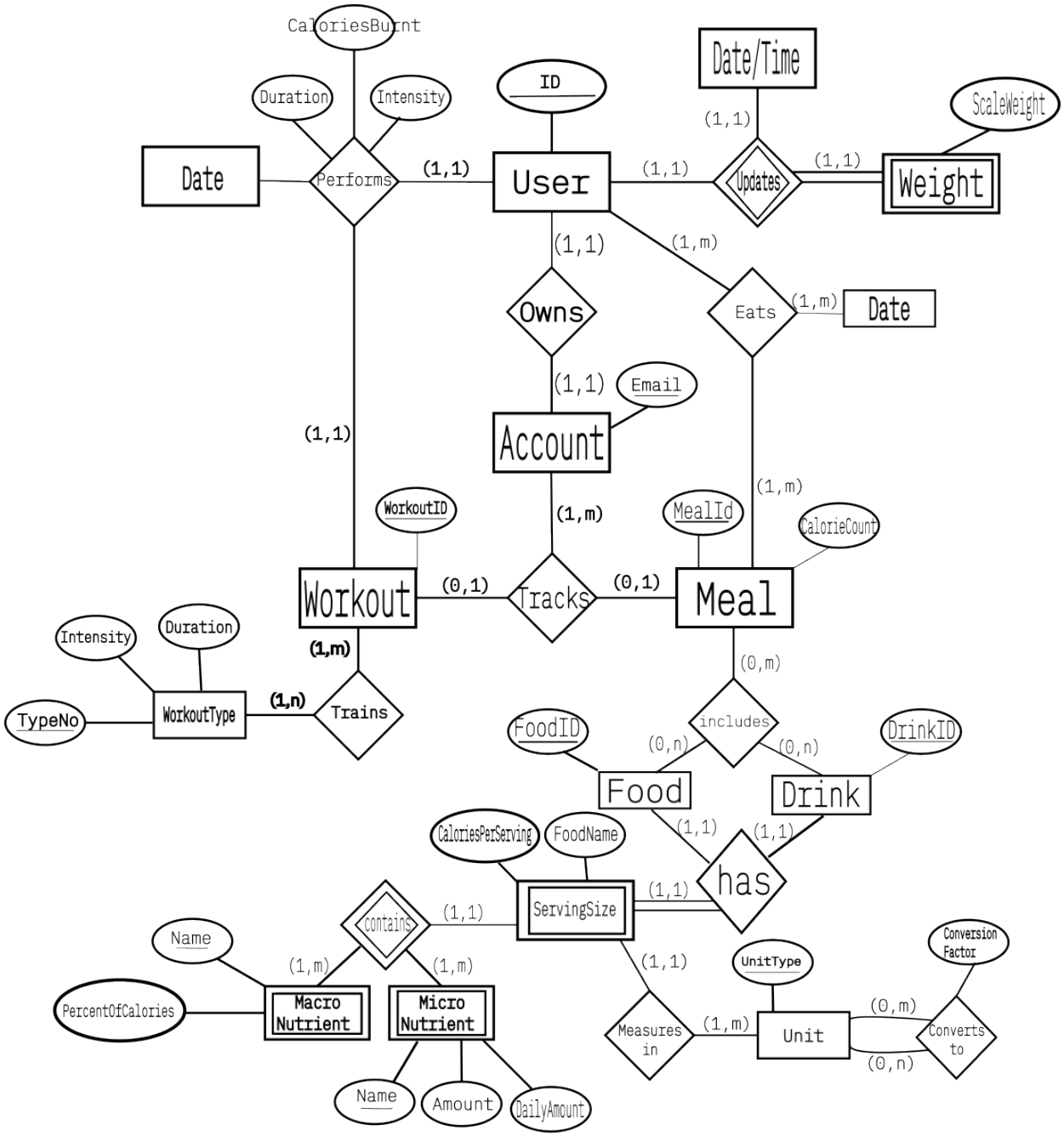
Matt Warner


CaloriesBurnt
Duration
Intensity
ID
Date/Time
ScaleWeight
Date
Performs
(1,1)
User
(1,1)
Updates
(1,1)
Weight
(1,1)
(1,m)
Owns
Eats
(1,m)
Date
(1,1)
Email
(1,1)
Account
(1,m)
WorkoutID
(1,m)
MealId
CalorieCount
Workout
(0,1)
Tracks
(0,1)
Meal
Intensity
Duration
(1,m)
(0,m)
TypeNo
WorkoutType
(1,n)
Trains
includes
FoodID
(0,n)
(0,n)
DrinkID
Food
Drink
CaloriesPerServing
FoodName
(1,1)
(1,1)
has
contains
(1,1)
ServingSize
(1,1)
Name
(1,m)
(1,m)
(1,1)
UnitType
Conversion Factor
PercentOfCalories
Macro Nutrient
Micro Nutrient
Measures in
(1,m)
Unit
(0,m)
Converts to
(0,n)
Name
Amount
DailyAmount

1 Entities & Atributes

- **User**
  - `ID` ***(Identifier)***. - Unique identification number used a surrogate key to identify a user.
  - `PhoneNumber` - The users phone number.
  - `Name` - The users legal name.
  - `Age` - The users age.
  - `DateOfBirth` - The users DOB.
  - `City` - Location of the user.
- `Weight` (Weak Entitiy)
  - `ScaleWeight` - Weight of the user.
- `Account`
  - `Email` ***(Identifier)***. The users email address. Serves as a natural key to identify accounts.
  - `Username` - Identity of the user.
  - `CreationDate` - The date in which the account was created.
- `Meal`
  - MealID ***(identifier)*** - surrogate key used to identify a meal.
  - `CalorieCount` - Amount of calories present in a meal.
- `Food`
  - `FoodID` ***Identifier*** - Surrogate key used to identify a food item.
  - `Name` - The name of the food.
- `Drink`
  - `DrinkID` ***(Identifier)*** - Surrogate key used to identify a drink item.
  - `Name` - The name of the drink.
- `ServingSize` ***(Weak entitiy)*** - Serving size depends on the existence of `Food` or `Drink`.
  - `FoodName` - The name of the food (or drink).
  - `CaloriesPerServing` - The amount of calories present. This data is tracked in order to...
- `Unit`
  - `UnitType` ***(Identifier)*** - Natural key that is the type of unit (since unit names are unique).
- `MacroNutrient`
  - `Name` ***(identifier)*** - The name of the macronutirent. Since all names are unique, this works as a natural key to identify a macronutriant.
- `MicroNutrient`
  - `Name` ***(identifier)*** - The name of the micronutrient. Since all names are unique, this works as a natural key to identify a micronutrient.
- `Workout`
  - `WorkoutID` ***(identifier)*** - Unique ID given to a workout to identify it.
- `WorkoutType`

- `TypeNo` ***(identifier)*** - Surrogate key used to indentify a `WorkoutType`.
- `Type` - Specific type of workout that describes `WorkoutType`.
- `Duration` - Amount of time taken to complete the workout.
- `Intensity` - The level of intensity for the type of workout.

-

2 Relationships & Cardinality

- ***performs*** (binary)
  - `User` ***performs*** `Workout`

  one-to-one - If we know the `User`, and we know the `Date`, it must belong to a single instance of a `workout` . Additionally, if we know the `Workout` and the `date`, then there can only be ***one*** `user` associated with that workout at that specific date.

- ***owns*** (binary)
  - `User` ***owns*** `Account`

  one-to-one - Only one `account` can be owned by a `User`. And only one `User` can ***own*** a given account `Account`.

- ***Updates*** (binary)
  `User` ***updates*** `Weight`

  one-to-one - A user can update one instance of his/her weight at a specific `Time`.

- ***Tracks*** (ternary)
  - `Account` ***tracks*** `Meal` (**one-to-many**) - An `Account` can track many meals. But a meal can be tracked by one `Account`
  - `Account` ***tracks*** `Workout`. **(**one-to-many) Same logic applied here.

- ***includes*** (ternary)
  - `Meal` ***includes*** `Food`
  - `Meal` ***includes*** `Drink`

- ***Trains*** (binary)
  - `Workout` ***Trains*** `WorkoutType`

- ***contains*** (ternary)
  - `ServingSize` ***contains*** `Macronutrient`
  - `ServingSize` ***contains*** `micronutrient`

- ***has*** (ternary)
  - `Food` ***has*** `ServingSize`
  - `Drink` ***has*** `ServingSize`

- ***eats*** (binary)
  - `User` ***eats*** `Meal`

- ***Measures in*** (binary)
  - `ServingSize` ***Measures in*** `Unit`