

NutriChoice - Stage 2

Elizabeth Binkina, Simon Hu, Nynika Badam

ER Model

Entities and Relationships

Entities:

1. *User info*: This includes user specific data such as name, height, weight, and age. These are attributes that are unique to the user so they are maintained within this specific entity.
2. *Food items*: This entity stores all food items in our database that will be recommended to each user. It has its own entity because it keeps track of the nutrients and macros as well as their corresponding volume, allowing for the entity to have relationships with the other entities. For example, Kraft Cheese would count as a Food Item and have its own FoodID.
3. *Food Group*: This entity consists of the food group that the food item corresponds to. For example, Kraft Cheese would belong to the dairy category.
4. *Vitamins/Macros*: These are entities rather than an attribute to provide ease in navigating between food characteristics and granular control. This also allows for specifying how much of each nutrient is present in the specific serving size of each food item. These two are separate entities and aren't meant to be counted together.
5. *Factors*: This is an entity that acts as a tool to specify what a user wants to track within the progress entity. It consists of foreign keys to connect the nutrient entities to the user.

Relationships:

1. Every food item can have many vitamins (many to many)
2. Every macro can have many food items (many to many)
3. Every food item belongs to one food group (ex. Cheese belongs to dairy) (many to one)
4. A user can be tracking multiple diet goals (one to many)
5. Each progress goal will be tracking one specific vitamin/macro/food type (one to many)

Progress entity is directly connected to macros and vitamins because we're tracking progress at the nutrient level rather than by how many food items are consumed.

The amounts of each vitamin/mineral and macro correspond to a serving size specified in the relationship set before the FoodItems entity. The amounts contribute to the overall GoalIntake attribute in the Progress entity.

Normalized Model

1. UserInfo (UserID, Customer_Name, Weight, Age, Height)
UserID → Customer_Name, Weight, Age, Height
A → B, C, D, E (Converted relation into terms where A = UserID, B = Customer_Name...)

R(A, B, C, D, E) (3NF decomposition)

UserID is the primary key and has to correspond to the rest of the keys

2. FoodItems (FoodID, FoodName, ServingSize, GroupID)
FoodID \rightarrow FoodName, ServingSize, GroupID
A \rightarrow B, C, D (Converted relation into terms in order where A = FoodID, B = FoodName...)
R(A, B, C, D) (3NF decomposition)

FoodID has to correspond to food names and serving sizes since it is a primary key in the relation.

3. Vitamins (VitaminID, VitaminName, VitAmount)
VitaminID \rightarrow VitaminName, VitAmount
A \rightarrow B, C (Converted relation into terms in order where A = VitaminID, B = VitaminName,...)
R(A, B, C) (3NF decomposition)

VitaminID is the primary key since it dictates the table. VitaminID corresponds to a VitaminName and VitAmount (mg) for each serving size.

4. Macros (MacroID, MacroName, MacAmount)
MacroID \rightarrow MacroName, MacAmount
A \rightarrow B, C (Converted relation into terms in order where A = MacroID, B = MacroName...)
R(A, B, C) (3NF decomposition)

MacroID is the primary key since it dictates the table. MacroID corresponds to MacroName and MacAmount (g) for each serving size.

5. FoodGroup (GroupID, GroupName)
GroupID \rightarrow GroupName
A \rightarrow B (Converted relation into terms in order where A = GroupID, B = GroupName...)
R(A, B) (3NF decomposition)

GroupID is the primary key of this table and it is in relation to GroupName. GroupID specifies which food category group it is in and GroupName the name of that group.

6. Factors (ProgressID, UserID, GoalIntake, CurrentIntake)
ProgressID \rightarrow UserID, GoalIntake, CurrentIntake

$A \rightarrow B, C, D$ (Converted relation into terms in order where $A = \text{UserID}$, $B = \text{GoalIntake}, \dots$)
 $R(A, B, C, D)$ (3NF decomposition)

ProgressID is the primary key of this table and tracks the current progress information for the User. The table also has foreign keys to UserID because the Progress table must be in relation to the user and what food they consumed.

Logical Design

1. UserInfo (UserID: INT [PK], Customer_Name: VARCHAR(100), Height: DECIMAL, Age: INT, Weight: DECIMAL)

UserID	INT {PK}
Customer_Name	VARCHAR(100)
Height	DECIMAL
Age	INT
Weight	DECIMAL

2. FoodItems (FoodID: INT [PK], FoodName: VARCHAR(100), ServingSize: INT, GroupID: INT [FK to FoodGroup.GroupID])

FoodID	INT {PK}
FoodName	VARCHAR(100)
Serving_Size	INT
GroupID	INT {FK to FoodGroup.GroupID}

3. Vitamins (VitaminID: INT [PK], VitaminName: VARCHAR(100), VitAmount: DECIMAL)

VitaminID	INT {PK}
VitaminName	VARCHAR(100)
VitAmount	DECIMAL

4. Macros (MacroID: INT [PK], MacroName: VARCHAR(100), MacAmount: DECIMAL)

MacroID	INT {PK}
MacroName	VARCHAR(25)
MacAmount	DECIMAL

5. FoodGroup (GroupID: INT [PK], GroupName: VARCHAR(100))

GroupID	INT {PK}
GroupName	VARCHAR(100)

6. Factors (ProgressID: INT [PK], Goal_Intake: DECIMAL, Current_Intake: DECIMAL, UserID: INT [FK to UserInfo.UserID])

ProgressID	INT {PK}
UserID	INT {FK to UserInfo.UserID}
Goal_Intake	DECIMAL
Current_Intake	DECIMAL