

## 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.
3. *Food type, Vitamins/Minerals, Macros*: These are entities rather than an attribute to provide ease in navigating between food characteristics and granular control.
4. *Attributes*: 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 vitamin/mineral can have many food items (one to many)
3. Every food item can have many macros (one to many)
4. Every macro can have many food items (many to many)
5. Every food item belongs to one food type (ex. Cheese belongs to dairy) (one to one)
6. Every food type can have many food items (one to many)
7. A user can be tracking multiple diet goals (one to many)
8. Each progress goal will be tracking one specific vitamin/macro/food type

Progress entity is directly connected to macros, food type, 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, macro, and food type correspond to a serving size specified in 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)  
FoodId  $\rightarrow$  FoodName, ServingSize  
A  $\rightarrow$  B, C (Converted relation into terms in order where A = FoodId, B = FoodName...)  
R(A, B, C) (3NF decomposition)

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

3. Vitamins (VitaminId, FoodId, VitaminName, Amount)  
VitaminId, FoodId  $\rightarrow$  VitaminName, Amount  
A, B  $\rightarrow$  C, D (Converted relation into terms in order where A = VitaminId, B = FoodId...)  
R(A, B, C, D) (3NF decomposition)

VitaminId and FoodId is the composite primary key since they both dictate the table. VitaminId and FoodId correspond to a VitaminName and amount (g) for each serving size. It should also be noted that FoodId is a foreign key and part of the composite primary key for this table.

4. Macros (MacroId, FoodId, MacroName, Amount)  
MacroId, FoodId  $\rightarrow$  MacroName, Amount  
A, B  $\rightarrow$  C, D (Converted relation into terms in order where A = MacroId, B = FoodId...)  
R(A, B, C, D) (3NF decomposition)

MacroId and FoodId is the composite primary key and they both correspond to the MacroName and amount (g) for each serving size. It should also be noted that FoodId is a foreign key and part of the composite primary key for this table.

5. Attributes (AttributeId, VitaminId, MacroId, TypeId)  
AttributeId  $\rightarrow$  VitaminId, MacroId, TypeId  
A  $\rightarrow$  B, C, D (Converted relation into terms in order where A = AttributeId, B = VitaminId...)  
R(A, B, C, D) (3NF decomposition)

AttributeId dictates what Vitamin or Macro or food type the food may be so it is the primary key while the other keys are foreign keys.

6. FoodType (TypeId, FoodId, TypeName, Amount)  
TypeId, FoodId  $\rightarrow$  TypeName, Amount

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

$\text{TypeId}$  and  $\text{FoodId}$  is the composite primary key of this table because the  $\text{TypeId}$  is in relation to the  $\text{TypeName}$  of the food and the amount of food type there is in each  $\text{FoodId}$ . It should also be noted that  $\text{FoodId}$  is a foreign key but also part of the composite primary key of this table.

7. Progress (ProgressId, UserId, AttributeId, GoalIntake, CurrentIntake)  
 $\text{ProgressId} \rightarrow \text{UserId}, \text{AttributeId}, \text{GoalIntake}, \text{CurrentIntake}$   
 $A \rightarrow B, C, D, E$  (Converted relation into terms in order where  $A = \text{UserId}$ ,  $B = \text{AttributeId} \dots$ )  
 $R(A, B, C, D, E)$  (3NF decomposition)

$\text{ProgressId}$  is the primary key of this table and tracks the current progress information for the User. The table also has foreign keys to  $\text{UserId}$  and  $\text{AttributeId}$  because the Progress table must be in relation to the user and what food they consumed.

### Logical Design

1. UserInfo (UserId: INT [PK], List\_Name: VARCHAR(25), Customer\_Name: VARCHAR(25), Height: DECIMAL, Age: INT, Weight: DECIMAL)

UserId	INT {PK}
List_Name	VARCHAR(25)
Customer_Name	VARCHAR(25)
Height	DECIMAL
Age	INT
Weight	DECIMAL

2. FoodItems (FoodId: INT [PK], FoodName: VARCHAR(25), ServingSize: INT)

FoodId	INT {PK}
FoodName	VARCHAR(25)
Serving_Size	INT

3. Vitamins (VitaminId: INT [PK], FoodId: INT [FK to FoodItems.FoodId], VitaminName: VARCHAR(25), Amount: DECIMAL)

VitaminId	INT {PK}
FoodId	INT {FK to FoodItems.FoodId}
VitaminName	VARCHAR(25)
Amount	DECIMAL

4. Macros (MacroId: INT [PK], FoodId: INT [FK to FoodItems.FoodId], MacroName: VARCHAR(25), Amount: DECIMAL)

MacroId	INT {PK}
FoodId	INT {FK to FoodItems.FoodId}
MacroName	VARCHAR(25)
Amount	DECIMAL

5. FoodType (TypeId: INT [PK], TypeName: VARCHAR(25), Amount: DECIMAL, FoodId: INT [PK/FK to FoodItems.FoodId])

VitaminId	INT {PK}
FoodId	INT {PK/FK to FoodItems.FoodId}
TypeName	VARCHAR(25)
Amount	DECIMAL

6. Attributes (AttributeId: INT [PK], TypeId: INT [FK to FoodType.TypeId], MacroId: INT [FK to Macros.MacroId], VitaminId: INT [FK to Vitamins.VitaminId])

AttributeId	INT {PK}
TypeId	INT {FK to FoodType.TypeId}

MacroId	INT	{FK to Macros.MacroId}
VitaminId	INT	{FK to Vitamins.VitaminId}

7. Progress (ProgressId: INT [PK], Goal\_Intake: DECIMAL, Current\_Intake: DECIMAL, AttributeId: INT [FK to Attributes.AttributeId], UserId: INT [FK to UserInfo.UserId])

ProgressId	INT	{PK}
AttributeId	INT	{FK to Attributes.AttributeId}
UserId	INT	{FK to UserInfo.UserId}
Goal_Intake	DECIMAL	
Current_Intake	DECIMAL	