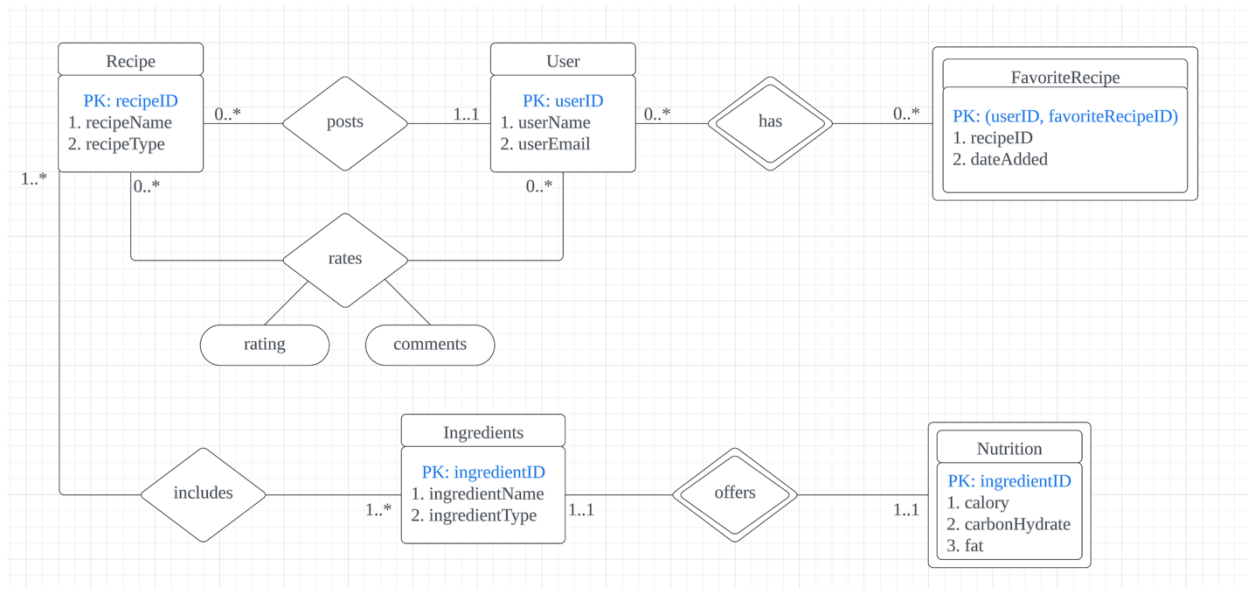


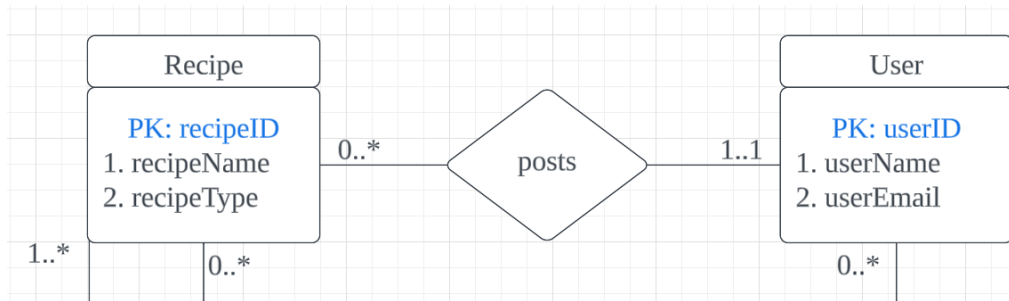
ER diagram:



Entities:

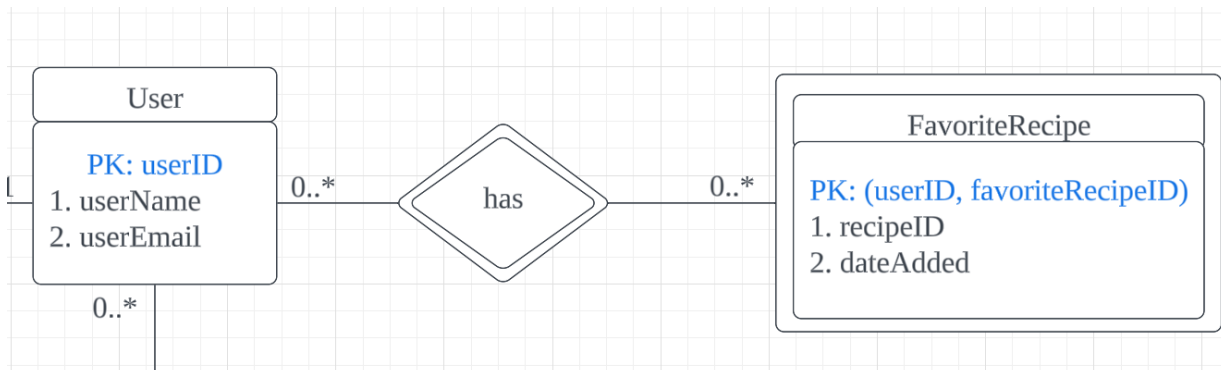
1. **Recipe:** Because Recipe is unique in our project, and contains multiple attributes which makes it too hard to be an attribute of other entities.
2. **User:** User plays a huge role in the relationships as shown in the above ER diagram, so it will be much more efficient to break out to its own entity to handle the relations.
3. **FavoriteRecipe:** Because user can have multiple favorites recipes, keep the array of data in the user will have anomalies, so we decide to break the favoriteRecipe out to its own table.
4. **Ingredients:** Ingredients also suffer the same anomalies like the favoriteRecipes(array of data). Therefore, we decide to make it an entity. Not to mention, there are also attributes within ingredients which is worth make it an entity.
5. **Nutrition:** Originally, the we include the nutrition as an attribute in the Ingredients entity. However, being adviced by TA, we follow his suggestion to make the nutrion an entity to make the data & relations clearer.

Relationship Assumptions:



1. User-Recipe relationship (1-to-many):

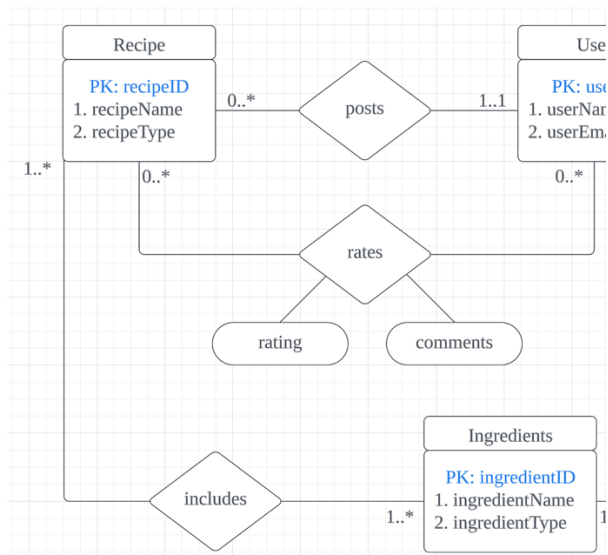
- We assume each user can upload 0-to-many recipes. “0”, in this case, means some of the users only want to learn from the other, and not contributing their recipes, which is normal. Of course, some users like to share 1 or more recipes they know, this is where the “many” comes from.
- We assume there are no redundant recipes (for the sake of data cleaning). Therefore, each recipe can only be posted by one user. EX: if someone posts the recipe for scrambled eggs, then other will not be able to post it.



PS: Weak Identity

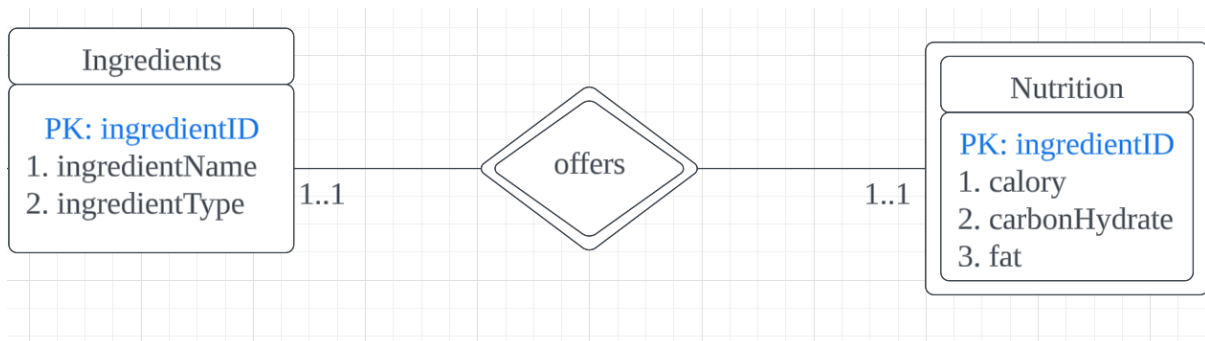
2. User-FavoriteRecipe relationship (1-to-many):

- Each user can have 0 or more favorite recipes that they can mark.
- Each recipe can be mark as favorite by 0 or more users.



3. Recipe-Ingredients relationship (many-to-many):

- Each recipe can contain multiple ingredients.
- Each ingredient might appear in 1 or multiple recipes as well. Why “1” you might ask because user can only be allowed to add ingredients to the database when they post cuisines that contain these ingredients. We do not allow users to add more than they need ingredients to the database, ie we do not allow users to just post raw ingredients without a cuisine recipe.



4. Ingredients-Nutrition relationship (1-to-1):

PS: Weak Identity

- Each ingredient can contain their corresponding nutrition group values.
- Each nutrition group values belong to their corresponding ingredient.

Normalization:

Normalization Decision: We decide to use 3NF for our database design. One of the reason is because BCNF will take more effort to derive, so we decide to cut some corners. Another reason is that BCNF does not preserve FD, therefore for the “safety” concerns, we decide to NOT use BCNF.

1. Recipe

Recipe

Naming $\left\{ \begin{array}{l} A = \text{recipeID} \\ B = \text{recipeName} \\ C = \text{recipeType} \end{array} \right\}$

Relation: $A \rightarrow B, B \rightarrow A, A \rightarrow C$

Min Base: L M R N
A B C

$\textcircled{A} \rightarrow ABC$

FD:

$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow B$
$B \rightarrow A$	$B \rightarrow A$	$B \rightarrow A$
$A \rightarrow C$	$A \rightarrow C$	$A \rightarrow C$
right is clear	left is clear	cannot substitute

2. User

User

Naming $\left\{ \begin{array}{l} A = \text{userID} \\ B = \text{userName} \\ C = \text{userEmail} \end{array} \right\}$

Relation: $A \rightarrow B, A \rightarrow C$

Min Base: L M R N
A BC

$\textcircled{A} \rightarrow ABC$

FD:

$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow B$
$A \rightarrow C$	$A \rightarrow C$	$A \rightarrow C$

3. Ingredients

Ingredients

Naming $\left\{ \begin{array}{l} A = \text{ingredientID} \\ B = \text{ingredientName} \\ C = \text{ingredientType} \end{array} \right.$

Relation: $A \rightarrow B, A \rightarrow C$

Min Base: L M R N
A BC

$A \rightarrow ABC$

FD:

$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow B$
$A \rightarrow C$	$A \rightarrow C$	$A \rightarrow C$
right is clear	left is clear	cannot substitute

4. FavoriteRecipe

FavoriteRecipe

Naming $\left\{ \begin{array}{l} A = \text{userID} \\ B = \text{favoriteRecipeID} \\ C = \text{recipeID} \\ D = \text{dateAdded} \end{array} \right.$

Relation: $AB \rightarrow C, AB \rightarrow D$

Min Base: L M R N
AB CD

$AB \rightarrow ABCD$

FD:

$AB \rightarrow C$	$AB \rightarrow C$	$AB \rightarrow C$	} $AB \rightarrow CD$
$AB \rightarrow D$	$AB \rightarrow D$	$AB \rightarrow D$	
right is clear	left is clear	cannot substitute	

5. Nutrition

Nutrition

Naming $\left\{ \begin{array}{l} A = \text{ingredientID} \\ B = \text{Calory} \\ C = \text{carbohydrate} \\ D = \text{fat} \end{array} \right.$

Relation: $A \rightarrow B, A \rightarrow C, A \rightarrow D$

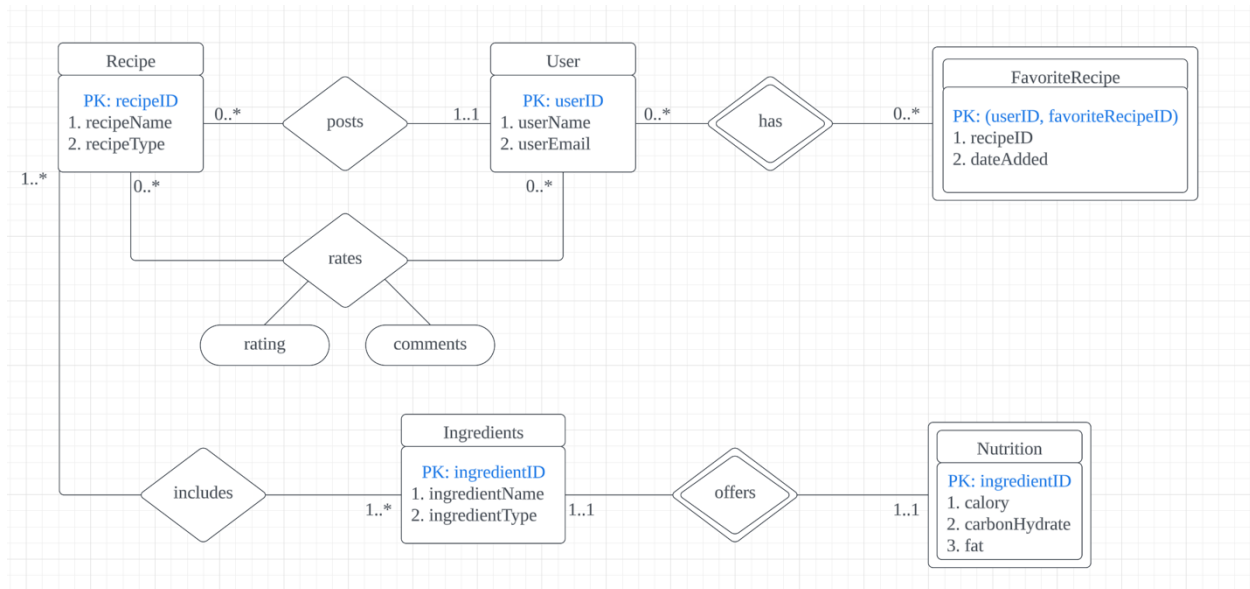
MinBase: L M R N
 A BCD

$A \rightarrow ABCD$

FD:

$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow B$	} $A \rightarrow BCD$
$A \rightarrow C$	$A \rightarrow C$	$A \rightarrow C$	
$A \rightarrow D$	$A \rightarrow D$	$A \rightarrow D$	
right is clear	left is clear	cannot substitute	

Relational Schema:



PS: PK color, FK color, PK & FK

1. Recipe (recipeID: INT [PK], recipeName: VARCHAR(20), recipeType: VARCHAR(10));
2. User (userID: INT [PK], userName: VARCHAR(20), userEmail: VARCHAR(30));
3. FavoriteRecipe (userID: INT [PK, FK to User.userID], favoriteRecipeID: INT [PK], recipeID: INT [FK to Recipe.recipeID], dataAdded: Date);
4. Ingredients (ingredientID: INT [PK], ingredientName: VARCHAR(30), ingredientType: VARCHAR(10));
5. Nutrition (ingredientID: INT [PK, FK to Ingredients.ingredientID], calory: DECIMAL(10, 2), carbonHydrate: INT, fat: DECIMAL(5, 2));