Group 1: Recipe Database Final Submission

Haya Ahmed & Jessica Aten

http://flip3.engr.oregonstate.edu:19951/

Group Feedback (in reverse chronological order):

Feedback by the reviewers on Steps 5 + 6:

“As I mentioned in step 5, you have your author delete working so I would just focus on a many-to-many delete. You mentioned you were having problems with the delete a category from a recipe. When you try to delete "category from a recipe" is the problem in getting the drop down menus to populate? In the javascript file you'll need to run a getRecipe() function and a getCategory() to be able to populate them. You can show the name but have the selected values be the id, like in the example videos.

If there's any specific errors you are getting we can help figure it out. Good luck” -Matthew

“It looks like you are getting close except for the items you mentioned above. I was able to delete an author and the results cascaded to recipes created by the author as expected. I also did a few general adds and everything seemed to act as expected.

When I clicked Browse I received a 404 error, so I couldn't test this part. I also noticed one other behavior that I wasn't sure about. When I add a Category in the Category page, it does not add a selection box for the new Category in the Add a New Recipe section.

You mentioned that you are having problems with Deleting a Category from Recipe. Are you observing some specific behavior? Thanks, John”

“Hello Haya, From the project guidelines: "You need to include one DELETE and one UPDATE function in your website, for any one of the entities." You have the author's page able to delete so I wouldn't worry about deleting for the other pages aside from the many-to-many delete.

For Step 5, the update function, I would only focus on one of the tables. Are there any certain problems you have having with the update? I saw that your edit button on the recipe page has an onclick for the updateCategory() with the correct category ID, so that looks good. If you let us know what kind of errors you are getting we can help figure it out. Good luck! -Matthew”

“I was able to add and then delete an author. The pop up verifying if we are sure is a nice touch! Great job!” - Savannah Loberger

“Deleted an author and everything look a-ok on my end. Nice work!” - Christopher Matian

“If you need help with update functions in the coming week I can lend some help and review your code and see what's up. Hopefully you manage to get it up and running, though!” - Christopher Matian

“great start! i was able to delete an author, and i really like the confirmation pop up to prevent errors.” - Kirsten Wollam

Feedback by the reviewers on Step 4:

“Your website looks really good! The first thing I noticed is that the Browse page gives a 404 error. All of your add and delete buttons seem to work well. One suggestion would be to change your UnitID to actual units on the ingredients page.

The only other thing I can find is that you might want to add a way to prevent negative or empty submissions to your database. Also, the delete recipe dropdown doesn't include any actual recipes.

Other than those two minor issues, everything looks great. Keep up the good work!”

“This is a really cool project. I am enjoying seeing how it is turning out.

I did not find any Create or Read functionality that was not working. Most of the Update/Delete did not seem to be functional when I tested, which I assume is intentional. I also received a 404 error when I selected the Browse tab, so not sure if that is intentional at this point or not.

Tables appear to be correct, but it would be clearer for a human user if you had the actual item name (course, ingredient, author... whatever) rather than showing integer indexes. Also, I was able to add duplicate entries to some tables where you might not want them (such as creating duplicate entries for an ingredient "cup(s) of water." Not a huge deal, but I would suggest just making a requirement in DDQs for entries to be unique.

Anyway, that is all I could find to suggest. Hope it is helpful and good luck!” - John

“Your webpage looks great. I was able to add things to all tables and everything is displayed, which is what was required for this week. When I added something to the course and authors they show up correctly in the drop down menus in the Recipe section. The delete button worked well for the authors table as well.

The only thing I can advise is for the Recipe page, you have a few check boxes for the categories. If people add more categories will more check boxes be added? It might be better to use a drop down menu for that if you get a lot of category submissions. Also, when I added a new recipe there's no place in the table for the category.

Other than that, great job with everything that was due this week.” -Matthew

“Hi Jessica. I poked around your site a little and made a few submissions (Persian, Dingo Berry Author, etc) and it seems to work quite well. Nice work so far! I have no major feedback to provide as it seems like you and your teammate are on the right track.” - Christopher Matian

“This is looking great! i was able to see and add to all the categories. The only small thing is that i might make the recipe list the default page, rather than just having it go to a blank index page.” - Kirsten Wollam

Actions based on the feedback - Steps 5-6: Update and delete functionality

List briefly the actions that you chose to take based on the above feedback. If you decided not to act on a specific suggestion, you need to describe your reasoning in detail.

You mentioned you were having problems with the delete a category from a recipe. When you try to delete "category from a recipe" is the problem in getting the drop down menus to populate? In the javascript file you'll need to run a getRecipe() function and a getCategory() to be able to populate them. You can show the name but have the selected values be the id, like in the example videos	We had some syntax issues that we were able to troubleshoot and fix to get the delete functionality working.
When I clicked Browse I received a 404 error, so I couldn't test this part. I also noticed one other behavior that I wasn't sure about.	The Browse page was not available because it was under construction. We ended up doing away with it and going a different route for providing filtering functionality.
When I add a Category in the Category page, it does not add a selection box for the new Category in the Add a New Recipe section.	We eliminated the checkboxes, implemented a dropdown menu and tested this to ensure that it does populate with a newly added category.
The delete recipe dropdown doesn't include any actual recipes.	We had not finished working on that part of the website and we were aware it wasn’t working yet.
Jessica’s Step 5 - 6 feedback mainly involved confirmation that deletions and updates were working.	We continued to implement the functionality across other parts of the website with confidence that user tests were successful.

Actions based on the feedback - Steps 4: Create and Read Functionality

Tables appear to be correct, but it would be clearer for a human user if you had the actual item name (course, ingredient, author... whatever) rather than showing integer indexes.	Changed the SQL queries to select names instead of id's for these values.
Some edit and delete buttons that weren't functional	We eliminated the extra ones that weren't working and were not required for the project specs.
I was able to add duplicate entries to some tables where you might not want them (such as creating duplicate entries for an ingredient "cup(s) of water." Not a huge deal, but I would suggest just making a requirement in DDQs for entries to be unique.	Added UNIQUE constraints to name attributes for tables such as category and courses.
The only thing I can advise is for the Recipe page, you have a few check boxes for the categories. If people add more categories will more check boxes be added? It might be better to use a drop down menu for that if you get a lot of category submissions.	We decided to use a dropdown menu and changed this on the form.
When I added a new recipe there's no place in the table for the category.	The Category-Recipe relationship is many to many, so this can be seen by viewing the separate relationship table.
Make the recipe list the default page, rather than a blank index page	We decided to add meaningful content to the index page and use this space to explain each section of the site in detail.

Feedback by the Reviewers - Step 3

"Hi Haya, Your website looks great so far. Everything is really well laid out and easy to follow. I just had one question: when you browse by Author, Category, Course, or Ingredients, does it take you to a separate page for each link or is it all the same page? I was not sure from looking at the site.

As far as your DDQ is concerned, the main thing I noticed was that you start inserting objects into your database before you declare all of your primary and foreign keys. As far as I know, this is not considered good practice and can lead to unpredictable behavior. Again, I am not very experience in SQL and I am not positive this is the wrong way to do it; it is just what I have read. Also, I think it would be easier to declare the primary keys when you create the table instead of adding an ALTER later on, but that is just me.

I think the DMQ looks good. The only thing I noticed is that you have the AuthorID insert instead of the FirstName and LastName. I do not know if this is intentional or not, but it is something to consider fixing. Other than that, I think it looks great.

Overall, I think you are off to a great start. Keep up the good work!"

"Hello Haya, As with another project, the data dump is showing the keys being added after the tables are created and the sample data is input. It may be a problem with the data dump but the keys should be assigned when the table is created.

Most everything in the html looks good. I didn't notice a submit button for the Add a new recipe field. I didn't know if the submit button with the Add ingredients to your recipe was part of that or just for add ingredients. For the ingredient quantity field I would maybe make that a number entry just to keep it uniform with the other entries.

Going through the dmq file everything looks good and it looks like every requirement is met.

Overall, great job! -Matthew"

"Hi Haya, I see your project is really progressing. I like the way it is turning out.

I just reviewed the tables and attributes against the schema and did not find any discrepancies. I noticed that the table structures are created, then the tables are populated with sample data, then primary keys and foreign keys are declared. I just mention that to point out that the behavior of keys is not really tested by the sample data in this case. If you wanted to confirm that this aspect works properly it would work better to have queries after the key declarations which would rely upon the key logic to ensure that it is working correctly.

A couple points with the key declarations also caught my eye, but may simply be a result of the tool used for developing these tables. One of these is the following…

ALTER TABLE `Recipe_Category`

ADD CONSTRAINT `Recipe_Category_ibfk_1` FOREIGN KEY (`CategoryID`) REFERENCES `Category` (`CategoryID`);

In the Recipe_Category table I see that Category_ID is defined to be a foreign key, but I do not see a similar definition for Recipe_ID. If I understand correctly both of these should be foreign keys. Also, I believe these two fields combined should probably constitute a primary key so that redundant rows will not be accepted. Instead I see an ADD KEY for CategoryID. As a quick

check I just inserted a row (1, 8), which is redundant with an existing entry an SQL accepted it. I THINK the more correct approach would be something along the lines of ADD PRIMARY KEY(RecipeID, CategoryID).

I also noticed one thing in the html/DMQs which may be worth your attention. As an example case, on the page entitled Publish a Recipe I see a text box to enter an author's name, however the accompanying INSERT references the author's ID. In most cases the ID would be transparent to the user, hence I would suggest either using a dropdown box (or table) with a list of all authors in the database (in which case the code would have access to the accompanying author ID) or allowing the user to enter the author's name and then relying on a SELECT statement to fetch the corresponding ID from the Author table.

That's all that really caught my eye. I'm glad to see how the project is turning out and hope that some of this feedback is helpful. Cheers, John"

Actions based on the feedback - Step 3

Feedback	**Action**
"You mention that a recipe will be required to have at least one ingredient, but i do not see where or how that will be enforced."	Because of the order of steps required to create entities, we will revise this relationship. A recipe can be created with 0 ingredients.
"Also, when adding an ingredient to a recipe, how does the form know which recipe it is added to? does the form add it to both the ingredient table and the bridging table to connect it to a particular recipe?"	Revised the form to correct this issue.
"One thing that I would add to the front end is a navigation back to the home page on the other pages or maybe a navigation bar for all the pages?"	Addressed this by adding a navbar to the top of all pages
"On your author page I don't see a space setup for deleting an author?"	We added an option to edit or delete authors (and all other entities) in the sample tables
"the data dump is showing the keys being added after the tables are created and the sample data is input...the keys should be assigned when the table is created.	Verified that the primary keys are assigned when the tables are created - we don't know why the dump would indicate otherwise or if this is relevant. We will look into this at office hours.
"In the Recipe_Category table I see that Category_ID is defined to be a foreign key,	Revised this so that the primary key is comprised of two foreign keys.

but I do not see a similar definition for Recipe_ID. If I understand correctly both of these should be foreign keys. Also, I believe these two fields combined should probably constitute a primary key so that redundant rows will not be accepted. Instead I see an ADD KEY for CategoryID. As a quick check I just inserted a row (1, 8), which is redundant with an existing entry an SQL accepted it. I THINK the more correct approach would be something along the lines of ADD PRIMARY KEY(RecipeID, CategoryID)."	
"As an example case, on the page entitled Publish a Recipe I see a text box to enter an author's name, however the accompanying INSERT references the author's ID. In most cases the ID would be transparent to the user, hence I would suggest either using a dropdown box (or table) with a list of all authors in the database (in which case the code would have access to the accompanying author ID) or allowing the user to enter the author's name and then relying on a SELECT statement to fetch the corresponding ID from the Author table."	Revised the form so that a dropdown menu is utilized to add an author to a recipe.
"I didn't notice a submit button for the Add a new recipe field."	There is a submit button to add a new recipe.
"For the ingredient quantity field I would maybe make that a number entry just to keep it uniform with the other entries."	Changed this to a numeric input
"when you browse by Author, Category, Course, or Ingredients, does it take you to a separate page for each link or is it all the same page? "	Browsing by an entity will generate a table of that entity joined with recipe. These table will be displayed on the same page.
"have the AuthorID insert instead of the FirstName and LastName."	This was referring to how authors were referencing when creating a new recipe. It's a design decision consistent with other references to foreign keys within our database

Actions based on the feedback - Previous Step 2

“Haya, I like your idea for a recipe database. It is a great real-life example to use. Overall, I think your draft is well written.

The database outline section looks good. I do have a question to consider though. Usually, cooking time is a range of times. Would it make more sense to have a minTime and maxTime to represent that? Again, that's just a thought. There is nothing wrong with the way it is now. Also, would it be possible to not repeat the total calories, fat content, carbohydrates, and protein content attributes in both the ingredients and nutritional information label entities. It seems like any relationship that could be made with the nutritional information label entity could be made with those same categories in the ingredients label.

The ERD looks good, as well. My only question is with the recipe and ingredients relationship. Wouldn't a recipe HAVE to have at least one ingredient in it for it to be considered a recipe? I do not see a situation where you could have a recipe with no ingredients, or even just one ingredient. You might want to consider fixing that.

Other than those two minor issues, this looks like a great start!”

“Hello Haya! This is a very clean definition. I did not spot any issues in the outline sections, but one thing did catch my attention in the ERD and Schemas.

I see that you have relationship tables included in the Schema for all relationships. There is no reason that this wouldn’t work, but separate relationship tables aren’t really required for relationships which are not many-to-many. For example, the Author to Recipe relationship is one-to-many, so it would be cleaner and simpler to just have a foreign key in the Recipe table which would relate to the key in the Author table directly.

Under the “Relationships in Schemas and Foreign keys” Section of the following there are subsections which discuss One-to-Many and Many-to-Many relationships specifically.

https://oregonstate.instructure.com/courses/1727186/pages/week-3-learn?module_item_id=18520960

Also, I THINK the participation portion of the relationship between Recipe and Ingredients may be backwards in the ERD. As it is shown, every ingredient would have to appear in at least recipe, but it would be permissible for a recipe to have no ingredients. I don’t see a specific statement to the contrary, but I suspect you meant for this to work the other way around. (I.e. it is ok for an ingredient to not be used in any recipe, but every recipe should have at least one ingredient.)

I don’t see any other problems per se, but just to call out one potential issue to keep in mind in future steps… if I am reading correctly, attributes in the Nutrition Info Label entity may be mathematically derived from attributes in the Recipe and Ingredients entities. This means that any time there is an update to one set of attributes the other must be updated as well in order to avoid stale data being out of synch. Just something to keep in mind in future steps.

This looks like an interesting project. Good luck! Cheers, John”

“Hello Haya, I read over your Recipe database and it seems like an interesting topic. The outline is well written and the ERD and schema make it easy to see the relationships.

For the database outline under the recipe entity you don't mention a "category" attribute but there is one in the ERD and schema. If there is a "category" attribute in the recipe entity will it be a foreign key for the Category entity, or will the Category entity use the CategoryName attribute? In the schema for the Recipe_Category table it has a Category from the Recipe entity and CategoryID from the Category entity both as keys for the CategoryID attribute.

Other than being confused by that, everything else looks great.”

“Great idea! Here are some things that i was thinking as i looked it over:

1) You state that Recipe has relationship with all the other entities, but in the listing for a Recipe in the outline you do not list an attribute for Author, Ingredient, or Category. Ingredient is also not on the ERD box for Recipe.

2) for Ingredient and Category, the key does not say it auto increments. Seems like that is likely how it is created but if it is a different method that could be indicated.

3) for ingredient you have calories, fat, carbs, and protein per serving. It might be nice to have an attribute that says what a serving is so that a user can know how much that is for.

4) the relationship between ingredient - recipe - and nutrition label is unclear to me. where is it indicated how much of something is in a particular recipe. Is there a separate listing for the ingredient each time a different amount is needed? maybe ingredient ID 3 is one egg and ID 4 is 6 eggs. Some clarification on how this is going to work would be good.

5) I am not sure why you have the tables recipe_nutrition and recipe_author as these are 1 to 1 and 1 to many relationships. I believe they can simply reference the correct ID and an intermediary table is only needed for many to many relationships.

6) because you have an intermediary table i don't believe you list category as a column on recipe, the recipe ID just points to the intermediary table.” - Kirsten Wollam

“Hi Jessica, I love this idea of creating a recipe database! Overall I think it looks great and really thought out. I think it was a smart idea to limit the nutritional content that is associated with each recipe as that can get quite long.

A few questions/suggestions:

- I like the idea of having categories. Have you thought about dividing the two categories for the recipes so there is a separate dietary and a region category? Maybe by adding another row in

your category table or by splitting the two? I think the concept of categories would work either way, depending on the audience.

- For the ingredients entity, from what I can tell the nutritional values are for one serving. Would a head of lettuce be a separate ingredient from romaine lettuce? It is a little unclear and you might want to state that as a parameter for your ingredient entity. Great job so far!" - Savannah

Updates from Step 3

We made some significant changes to our original outline that simplifies our database and better conforms our database to the third normal form. Here are the changes:

- Removed Nutritional info label as an entity, which had a one to one relationship with recipes. The Nutritional info label entity had attributes that required calculations, the attributes total calories, fat content, ect were dependant on the respective attributes in the Ingredients entity
- Removed the attributes calories, fat content, carb content, and protein content from the Ingredients entity. Since we no longer had a Nutritional info entity, these attributes were no longer meaningful. Ingredients now has IngredientName, IngredientID and UnitID as its attributes.
- Added Courses as an entity, to replace nutritional info label. Courses are separate from Category.
- Added another column to the RecipeID_Ingredients relationship table to our schema, which is QuantityOfIngredient. Adding this additional column means that we do not need to classify the same ingredient as two different ingredients if the quantity is different (for example, 2 eggs and 4 eggs as different ingredients), as we previously had to do.
- Added an additional entity called Units to reference the units of measure for ingredients.
- The above changes are reflected in our written outline, ERD and schema

Feedback	**Action**
A question about why there is a "category" attribute listed under the Recipe entity on the ER diagram and Schema but not in the outline.	Determined that this was done in error. Deleted the category attribute from the Recipe entity on the ER diagram and Schema
Relationship tables included in the schema for all relationships, even one to many and one to one	Removed the relationship tables that were not many to many relationships

Question about the relationship between recipes and ingredients – ER diagram showed that it is possible for a recipe to have zero ingredients.	Decided that this is incorrect and a recipe should have at least one ingredient. That said, we decided that we would allow a recipe to consist of one ingredient, (for cases when the recipe was focused on a method).
“Attributes in the Nutrition Info Label entity may be mathematically derived from attributes in the Recipe and Ingredients entities. This means that any time there is an update to one set of attributes the other must be updated as well in order to avoid stale data being out of synch. “	Did not make any changes but will consider this in the future steps of the project.
Cooking Time should be a range of times like minTime to maxTime	Decided this was unnecessary and will use an approximation of the total cooking time
Don’t repeat nutrition attributes in Ingredients and Nutrition Information	Decided that this information should be included in both categories, since the nutritional label is derived from the ingredients.
“You state that Recipe has relationship with all the other entities, but in the listing for a Recipe in the outline you do not list an attribute for Author, Ingredient, or Category. You state that Recipe has relationship with all the other entities, but in the listing for a Recipe in the outline you do not list an attribute for Author, Ingredient, or Category. Ingredient is also not on the ERD box for Recipe”	Determined that the relationship tables in the schema demonstrate the relationship between recipe and all other entities.
Ingredient and Category keys are not specified as auto incrementing.	Added “auto incrementing” to the outline for clarity.
There needs to be a serving size for ingredients.	Added a quantity attribute to Ingredient
Question about how ingredient, recipe and nutrition label are related, particularly how quantities affect the relationship between ingredient and nutrition label.	This should be clarified by the addition of a quantity attribute to Ingredient.
Divide the two categories for recipes so there is a separate dietary and region category.	Decided this was unnecessary and overcomplicating, since a recipe can be both

	region and diet specific, or region specific but not diet specific ect.
Question about the nutritional values listed as attributes of the Ingredient entity.	Clarified this by adding the quantity attribute to Ingredient.

Upgrades to the Draft version for Step 2

Here is a summary of the changes we made to the draft based on the feedback we received:

- Removed one to one and one to many relationship tables from schema. These were the Recipe_Nutrition tables and Recipe_Author tables.
- Removed Category as a column under the schema (since they are a many to many relationship), but kept Author and Nutrition Label as columns since these reference foreign keys that are NOT a many to many relationship.
- Added a Quantity attribute to the ingredient entity.
- Updated IngredientID, CategoryID and NutritionID, specified that these auto increment.
- Revised attribute and column names so they match between the Schema, ER Diagram and Outline.

Fixes Based on Feedback from Step 1

1. Missing data type for the attribute “Email” under the Author entity.
2. Missing data types for the attributes “Fat Content,” “Carbohydrate Content,” and “Protein Content” under both the Ingredient and Nutrition Info Label entities.
3. This was not addressed in the grader’s feedback, but we determined that we should add the following attributes to the Recipe entity to reflect the relationship between Recipe and the other entities: Author, Category, and Nutrition Label.

Recipe Database - Project Outline

Group 1: Haya Ahmed & Jessica Aten

We are making a recipe database which is designed to store details of a user generated collection of recipes. We found this to be an ideal topic for our project because of the interesting, complex nature of the relationships between the entities we defined. For the purpose of simplifying this project, we've established some rules in our database that are not representative of recipe websites in real life, such as a recipe can only have one author in our database.

Database Outline - Final (Unchanged since Step 3 final version)

The entities in this database and their respective attributes are as follows. The key attribute for each entity is denoted by an asterisk.

1. Recipe: Recipe is the most important entity, and a recipe has a relationship with every other entity in this database.
    a. *RecipeID: Each recipe will be assigned a unique identification number when they are recorded into this database. This will be the primary key. This is an auto incrementing number stored as an int to keep track of the number of recipes in our database.
    b. RecipeTitle: The title of each recipe will be a varchar with a maximum of 100 characters. It cannot be blank and and there's no default.
    c. CookingTime: The total cooking time for the recipe will be an int in minutes.
    d. Servings: The number of people a recipe serves, which will be an int.
    e. AuthorID: An int, this lists the ID of the Author that wrote this recipe. It cannot be null as an recipe will always have an Author. A recipe cannot have an Author that doesn't exist in our database.

f. CourseID: An int, this lists the ID of the Course that this recipe belongs to. It cannot be null as an recipe will always have an Course. A recipe cannot have a Course that doesn't exist in our database.

2. Author
    a. *AuthorID: Everytime a user joins our database, they are assigned a unique numerical ID. This is an int and will be auto incrementing to keep track of the number of users in our database.
    b. FirstName: An author's first name is a varchar with a max of 50 characters. It cannot be blank and there is no default
    c. LastName: An author's last name is a varchar with a max of 50 characters. It cannot be blank and there is no default
    d. Email: An author registers for our database with a unique email. An author's email is a varchar with a max of 50 characters. It cannot be blank and there is no default.

3. Ingredients
    a. *IngredientID: An auto incrementing int. Every ingredient in our database has its own unique numerical ID.
    b. IngredientName: Every ingredient in our database has a name. It cannot be blank and there is no default. Ingredient names will be stored as a varchar with a maximum of 50 characters.
    c. Quantity: An int, the quantity of that ingredient that the recipe calls for.
    d. Unit: A varchar, this is the unit of the quantity of the ingredient (ie, in cups, tablespoons, mL, ounces, ect). It cannot be blank or null, but is it possible to have a none value for cases where there are no units (for example, 3 eggs).

4. Category
    a. CategoryID: An auto incrementing int. Every category has a unique numeric ID associated with it.
    b. CategoryName: Every category has a unique name. Category names will be stored as a varchar with a maximum of 50 characters. It cannot be blank and there is no default.

5. Courses
    a. CourseID: A predetermined int. Every course has a unique numeric ID associated with it.
    b. CourseName: Every course has a unique name. Category names will be stored as a varchar with a maximum of 20 characters. It cannot be blank and there is no default.
6. Units
    a. UnitID: An auto incrementing int. Every unit has a unique numeric ID.

b. UnitName: The unit's name is stored in a varchar(50).

Relationships

Recipes have categories: This is a many-to-many relationship since a category can have many recipes, and a recipe can fit into many categories. Both region-specific recipes (ie, Italian, Mexican, Western) and dietary-specific (vegetarian, nut-free, gluten-free) recipes are included under the umbrella of category.

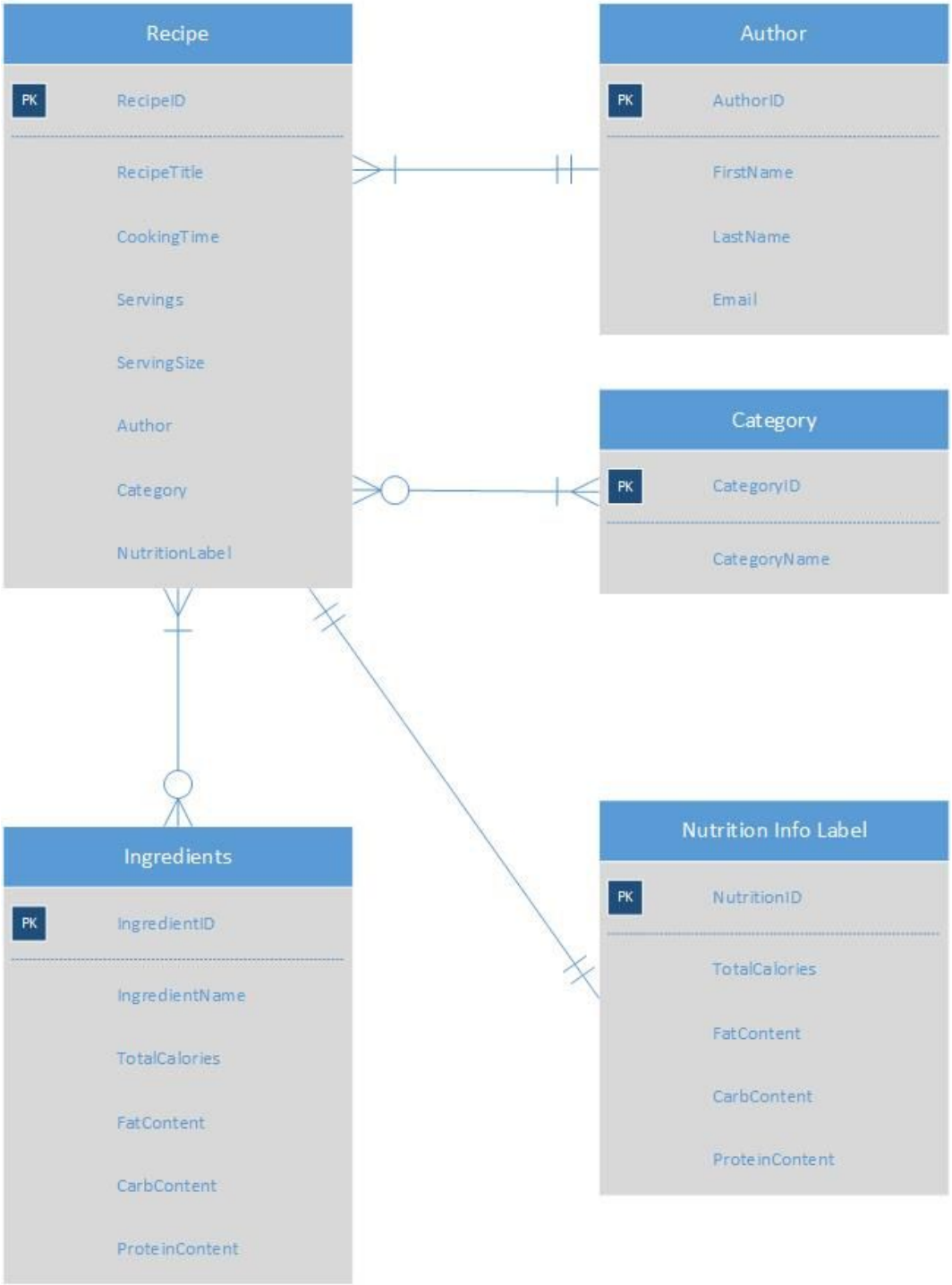
Recipes are created by authors: This is a one-to-many relationship. An author can write many recipes, but for the purpose of our database, a recipe can only have one author. Authors are users of this database.

Recipes consist of ingredients: This is a many-to-many relationship. Recipes have many ingredients, and an ingredient can be found in many recipes.

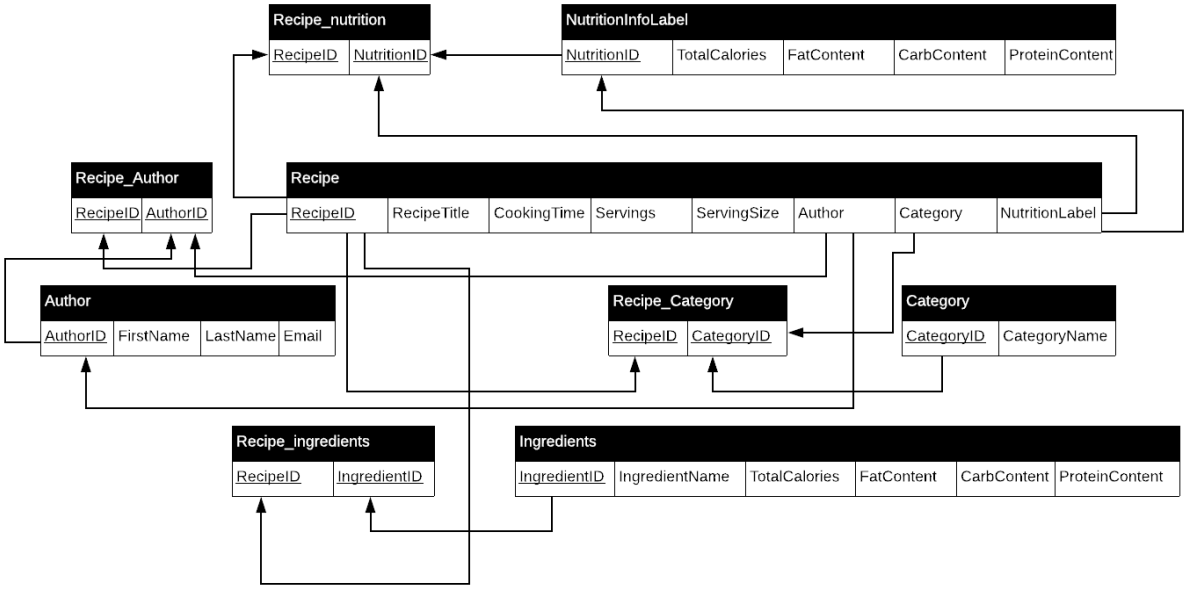
Courses have recipes: This is a one-to-many relationship. A recipe can be categorised as only one course, but a course can have many recipes associated with it. Examples of courses are breakfast, lunch, dinner, ect.

An ingredient has a unit of measure. This can be null to account for quantities that are not measured in units (for example, 3 eggs). A unit can zero or more ingredients associated with it. This is a zero-to-many relationship.

C) Recipe Database Entity-Relationship Diagram



d) Schema


Recipe_nutrition
RecipeID
NutritionID
NutritionInfoLabel
NutritionID
TotalCalories
FatContent
CarbContent
ProteinContent
Recipe_Author
RecipeID
AuthorID
Recipe
RecipeID
RecipeTitle
CookingTime
Servings
ServingSize
Author
Category
NutritionLabel
Author
AuthorID
FirstName
LastName
Email
Recipe_Category
RecipeID
CategoryID
Category
CategoryID
CategoryName
Recipe_ingredients
RecipeID
IngredientID
Ingredients
IngredientID
IngredientName
TotalCalories
FatContent
CarbContent
ProteinContent