

## Project Step 3 - Draft | CS 340

Justin Boyer | Ahmad El-Bobou

URL for list of pages: <http://web.engr.oregonstate.edu/~boyerju/New%20folder/>

URL for main page: [http://web.engr.oregonstate.edu/~boyerju/New%20folder/recipe\\_main.html](http://web.engr.oregonstate.edu/~boyerju/New%20folder/recipe_main.html)

### B.) Project idea:

The basic idea for the project is a recipe website. On this website, you can find the recipe for many different kinds of dishes. The main page will present the user with a search bar and some recent additions to the recipe base. The user can search for the title of a recipe, a type of recipe, and food the user can make with certain ingredients. Users will also have the opportunity to add recipes to the database.

### The basic break down of the entities and their attributes in the database:

- Recipe:
  - id INT AUTO\_INCREMENT PRIMARY KEY NOT NULL
  - name VARCHAR(25) NOT NULL
  - serving\_size INT
  - cook\_time INT
  - source\_id INT
  - instructions VARCHAR(max)
- Ingredient:
  - id INT AUTO\_INCREMENT PRIMARY KEY NOT NULL
  - food\_id INT NOT NULL
  - recipe\_id INT NOT NULL

- amount INT
- Food:
  - id INT AUTO\_INCREMENT PRIMARY KEY NOT NULL
  - name VARCHAR(25) NOT NULL
  - type VARCHAR(25) NOT NULL
- Type
  - id INT AUTO\_INCREMENT NOT NULL
  - name VARCHAR(25)
- Source
  - id INT AUTO\_INCREMENT NOT NULL,
  - name VARCHAR(25) NOT NULL,
  - year\_published INT,
  - author VARCHAR(25)

### **Description of entities, their attributes, and their relationships:**

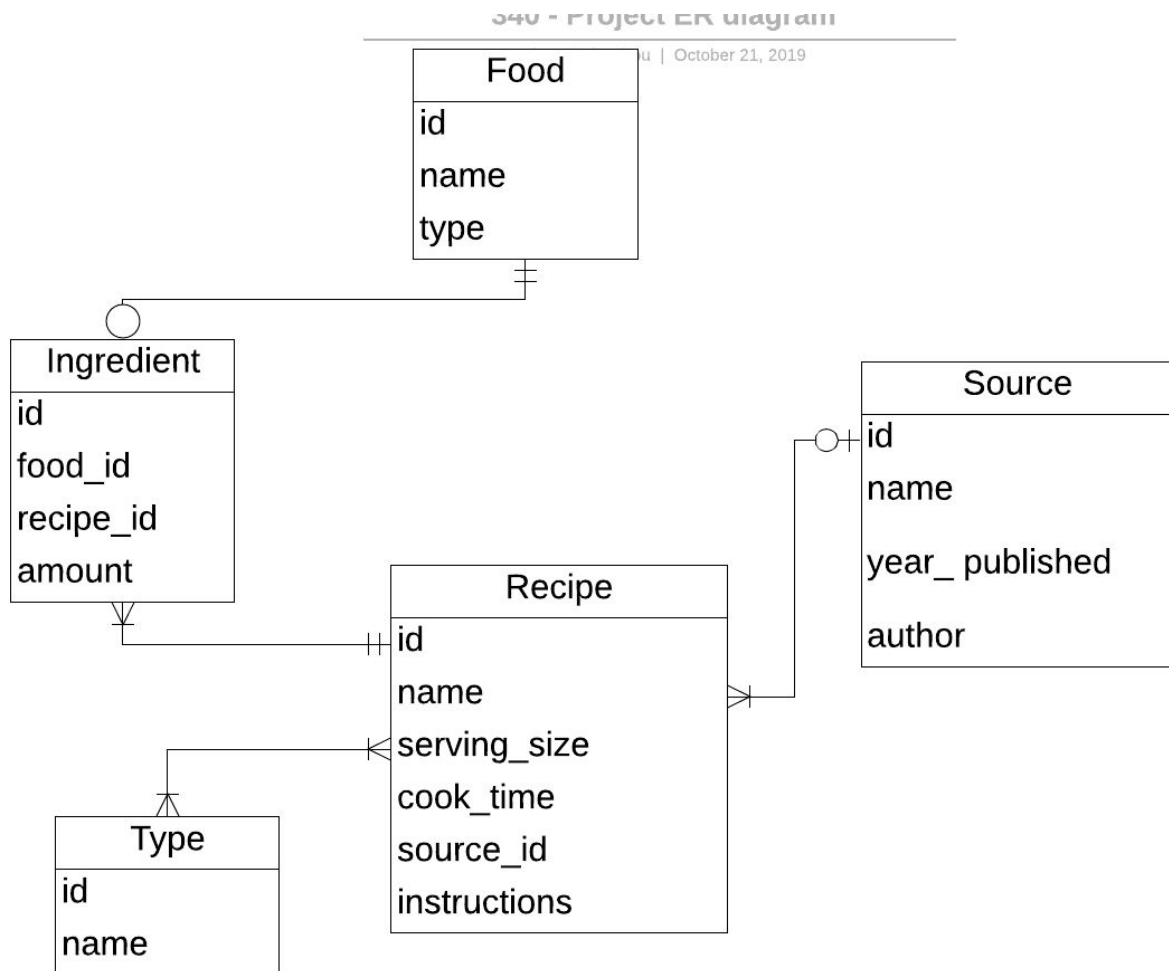
The recipe is the center entity in the database. Every other entity is somehow related to it. The attributes of a recipe will be id, name, serving\_size, and cook\_time, source and instructions.

All recipes are made of at least one ingredient and recipes can have more than one ingredient. An ingredient entity will be composed of the following attributes: id, food\_id, recipe\_id and amount. This means that there is a many-to-one relationship between ingredients and recipes. The food\_id attribute refers to another entity called food which has the following attributes: id, name, type (vegetable, fruit, meat, etc.). Food is related to ingredients in a one-to-many relationship. All ingredients have one food and many ingredients can have the same food, but no food has any ingredient.

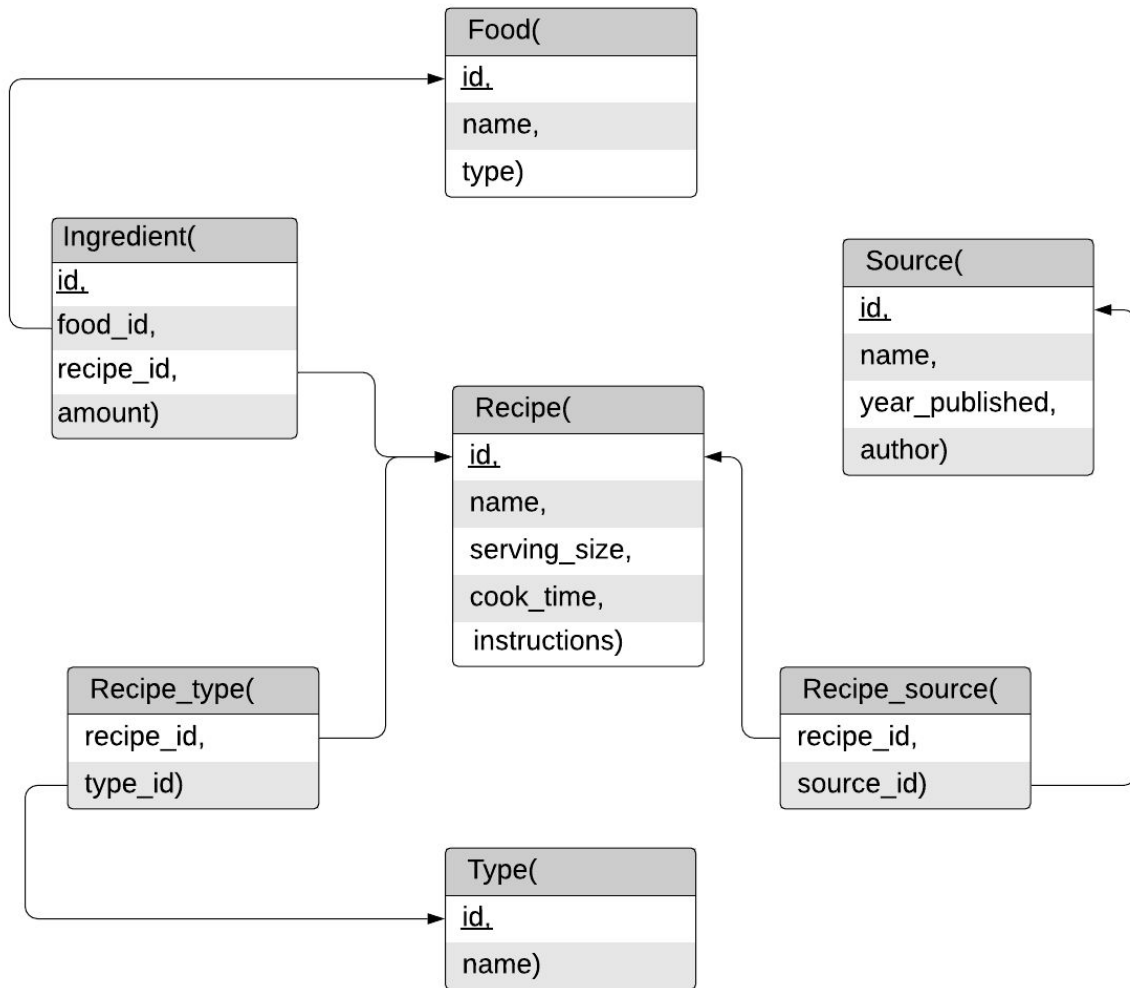
We need to be able to organize food into different types of food (breakfast, lunch, dinner, snack, dessert, winter, summer, diabetic, vegetarian, keto, etc.). The attributes of type will be: id and name. The relationship between type and recipe will be many-to-many. Each type can have more than one recipe and each recipe can have be more one type.

Lastly, recipes can come from a publication source. Sources have the following attributes: id, name, author, year\_published. The relationship between a source to a recipe is a one-to-many. Each source can have many recipes, but each recipe may be from zero or one source.

## B.) Entity Relationship Diagram



### C.) Schema



#### D.) Fixes based on feedback from previous steps.

##### Feedback from Step 1:

1. (-.25) Inconsistency between breakdown and description. (e.g. No mention of Cookbook source in the breakdown)
2. Ingredient table doesn't store any important information. Example: Ingredient name can be stored.
3. (-.5) Diet tags, Type these tables lack attributes to link them with other entities. You need some more attributes (possibly foreign keys).

##### Responses:

1. Fixed: The source\_id attribute of the recipe entity is now listed in the bullet point attribute section.
2. No changes: The ingredient table holds the very *important* information of **amount**. A recipe without indication of amount for each ingredient would not be very useful.
3. No changes: These entities do not need to be linked to by key. They have a many to many relationship with the recipe entity. Therefore, their relationship will be demonstrated through an auxiliary relation in the schema.

##### Feedback from Step 2 draft:

###### 1. Feedback by the Peer Reviewers

- a. Hello! I think this is a great idea for a database. As someone who loves to cook, I would really like having an easy-to-navigate recipe database to store recipes in and look new ones up.

I do have a few suggestions:

- i. "Type" is a bit confusing. I think either the "Type" entity or the "type" attribute of the "Food" table should be renamed. For example, changing the "Type" entity to "Cuisine" may clear up some confusion. Or changing the "type" attribute of the "Food" table to "food\_group".
  - ii. The id's of the tables that share relationships with the "Recipe" entity, I think, should be included in "Recipe". Otherwise, the user doesn't know how to relate the two entities. For example, each recipe has a set of instructions, but how do you link "Instructions" to "Recipe"? Adding "instructions\_id" to "Recipe" would solve the problem.
  - iii. I'm not sure what the relationship from "Food" to "Ingredient" is meant to be. Is it "one food item CAN be related to many ingredients", or "one food item CAN be related to ONE ingredient"?
- b. Hi, I like the concept you chose for your database. I'm curious, what was the reason behind creating a separate entity for instructions? In my mind, it makes sense to just include the instructions description in the recipe entity. In your ERD I noticed that the "Food" to "Ingredient" relationship is misrepresented here. The diagram suggests that "Food" does not have a relationship with "Ingredients" but your outline says "Food" has a none-to-many relationship with "Ingredients". I also suggest looking into combining "Food" and "Ingredients" into one entity, then make "Food/Ingredients" to "Recipe" a many-to-many relationship. You could then address the relationship between "Food/Ingredients" and "Recipe" in your schema.

- c. Hello! I really like your idea for the database. I cook frequently so I would love to use this in real life. Here's some feedback. I was confused at first about the relationships between food, ingredient, and recipe, but after thinking about it and examining your ERD I think it makes sense. I was going to suggest that you add an ingredient attribute to your recipe entity but then I had to figure where the amount would go. I think the food, ingredient, and recipe relationships are good. Is there a reason that there are diet and type entities? Can you add a diet attribute to your recipe entity as a varchar(25)? Also the same with the type, can you add a type attribute to your recipe entity as a varchar(25)? I also wonder if instead of the instructions entity, you can add an instructions attribute to recipe as a varchar(max)?
- d. Hi! I think this will be a really interesting use of a database, I hope to be able to see it once you all start implementing it! From looking at your entities, it looks like you can definitely cut down on how many you have. For example, the type entity looks like it only holds the name of the type and a primary key int. Is there any reason the name attribute from this table couldn't be added to say, recipe? Another example could be the recipe and instruction entities. At least in my mind, those two are essentially the same thing, but perhaps you have an idea that I'm just not seeing.
- e. **Instructor:** Hey Justin and Ahmad! Here's a few things I was able to catch after a quick read-through:

Schema:

- i. Recipe\_diet points to the name attribute of Diet instead of the primary key

Additional Notes:

- ii. You have a lot of entities - be aware that you'll be making these soon and you only need 4. If you can manage 7, however - great!
- iii. Just looking them, aside from a minor error, your ERD and schema look great! I would always recommend, however, making sure your designs match your outline.
- iv. As long as you can manage the size of your project, I'm looking forward to what your team makes!

## **2. Actions Based on Feedback**

- a. We left type the same as we thought that it should be a general category covering diet, food for special times of day, etc. We did not link the other tables to the recipe table as they are mostly many-to-many relationships. We maintained the food to ingredient distinction in order to ensure that data repetition is not created.
- b. We condensed instructions into an attribute of recipe, but we maintained the food/ingredient distinction.
- c. We did not follow this persons advice to add diet and type into the recipe as attributes as we wanted to ensure that eat recipe could have multiple diets and types.
- d. We did not follow this persons advice as they seemed to lack an understanding as to why many-to-many tables exist.



- e. **Instructor:** Resolved issue with schema attribute pointing to incorrect key and condensed the project into 5 tables out of 7.

### **3. Upgrades to Draft Version**

- a. No additional changes aside from changes outlined in the above section.

**Feedback from Step 2 final:** No feedback was given by grader from Step 2 - Final, so no changes have been made since that submission.