

Link to website:

<http://flip3.engr.oregonstate.edu:4576/>

Feedback by the peer reviewer:

Michele Larson:

This was a cool idea for a database! First, as a suggestion, I got confused with your table names. You are using “type” to describe both a menu type and a meal type. It would be clearer to be more verbose and specify “meal_type” and “menu_type” if you want to use “type” for both. Your ERD shows items has a many relationship with Type, but I think the participation is reversed. An item can only have one type, but a type can have many items. The meal type table is missing attributes (you don’t have them listed in your outline either, but they are in your schema). For “Menu” you have “number of items” as a primary key (underlined) but it is not one. I think you may want to add cardinality between items and menu (as far as I can tell, an item doesn’t exist without a menu, correct?) Same thing for type, types don’t exist without items, right? Your schema format doesn’t follow the class symbol key that we’re supposed to use. Next I looked at your sql file and imported it. I think it would be nice if the attributes were not a mix of uppercase, lowercase, and both upper and lowercase and followed the same conventions. I think you might have autogenerated this schema but there are some issues with it. When importing, I get this error: -- Constraints for table `items` ALTER TABLE `items` ADD CONSTRAINT `fk_items_Type1` FOREIGN KEY (`ID`) REFERENCES `Type` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, ADD CONSTRAINT `fk_items_menus` FOREIGN KEY (`ID`) REFERENCES `menus` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, ADD CONSTRAINT `fk_items_primary_ingredient1` FOREIGN KEY (`ID`) REFERENCES `primary_ingredient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION You can define the PRIMARY KEY in the initial table creation instead of altering the tables later. Also, you can insert the data all at the end after each table is created. All of your tables in the schema are missing auto increment and unique for the id. For “menus”, do you also want name to be unique, so someone doesn’t put “breakfast” twice, for example? Also for num_of_items”, the default value you gave it is a string, but you have defined it as an int. For “primary_ingredient”, do you also want PI_name to be unique, so someone doesn’t put the same ingredient twice in the table? Same thing for “type_name” in “Type”? Thanks!

David Sahni

ERD Peer Review: The attributes for the Item and Primary Ingredient seem fine, just need to add Name on Type. The Menu entity has both ID and Number of Items underlined as primary keys, so that is probably a typo that needs to be corrected. The Item and Primary Ingredient relationship looks right, with each item having one ingredient but an ingredient can be a primary for many items. The items to type relationship seems off, as your ERD suggests each Type can only be related to one item. The item/type relationship seems to be more many-to-many, where each item can have many types and each type can describe many items. The participation of the item and menu relationship is unclear from the outline as well, can items be on 0 menus and can menus have 0 items? Schema Peer Review: The attributes and variable types on the schema all look right. You are missing the extra table for the items/menu many-to-many relationship however. I would highly recommend looking at the syntax file in the Project 2 description/the walkthrough of schema/ERD on the modules page, and use that as reference for building the schema. I think they are going to grade based on those guidelines. DDQ Peer Review: It loads so that’s good! In your SQL you have your Items/Type relationship as many to 1, same as Items to Primary Ingredient, which makes sense so that might have just been a typo on the ERD. All the attributes are present, however the Items are on Menus table is still missing. So far, looks good! Just finish up your schema and SQL with the correct relationships, and you are on the right track!

Actions based on the feedback:

- Specify meal_type and menu_type
- Change all attributes to lowercase

- Change items and type relationship to match
- Meal type table is missing attributes
- Add total participation to item and menus
- Change cardinality to types and items
- Remove underline from number of items
- Change num_of_items to INT
- Make primary ingredient name unique
- Make primary ingredient name unique
- Add items to menus table relationship
- Add name to type

Upgrades to the Draft version:

- ERD
 - Removed number of items
 - Changed type to meal
 - Add name to meal
 - Add name to primary ingredient
 - Made items and menu total participation
 - Added meal <-> menu relationship
 - Removed name from menu (redundant)
- Schema
 - Changed completely to be in line with assignment
 - Changed to match ERD
- DDQ
 - Completely redid the SQL file
 - Added changes inline with ERD and Schema
 - Created restaurant name attribute in menu table

Restaurant Menu Database

The outline for this database represents the many variations that a restaurant menu can present. Restaurants have the potential of laying out their menus in any number of ways and data can be represented in a myriad of forms. This database will attempt to control the complexity and ease the maintenance of menus by highlighting common characteristics. Restaurants will often change menus based upon price, season, or local availability of ingredients. These changes require constant edits which can incur a high maintenance cost. Complexity can also arise when there are multiple locations with varied menus. Being able to correlate this data and display it in different forms will simplify the upkeep of menus. This database could be used on any restaurant website or to collect data for printed menus. The data could be further disseminated to make decisions about pricing or menu layout.

Database Outline, in Words The entities in my database are:

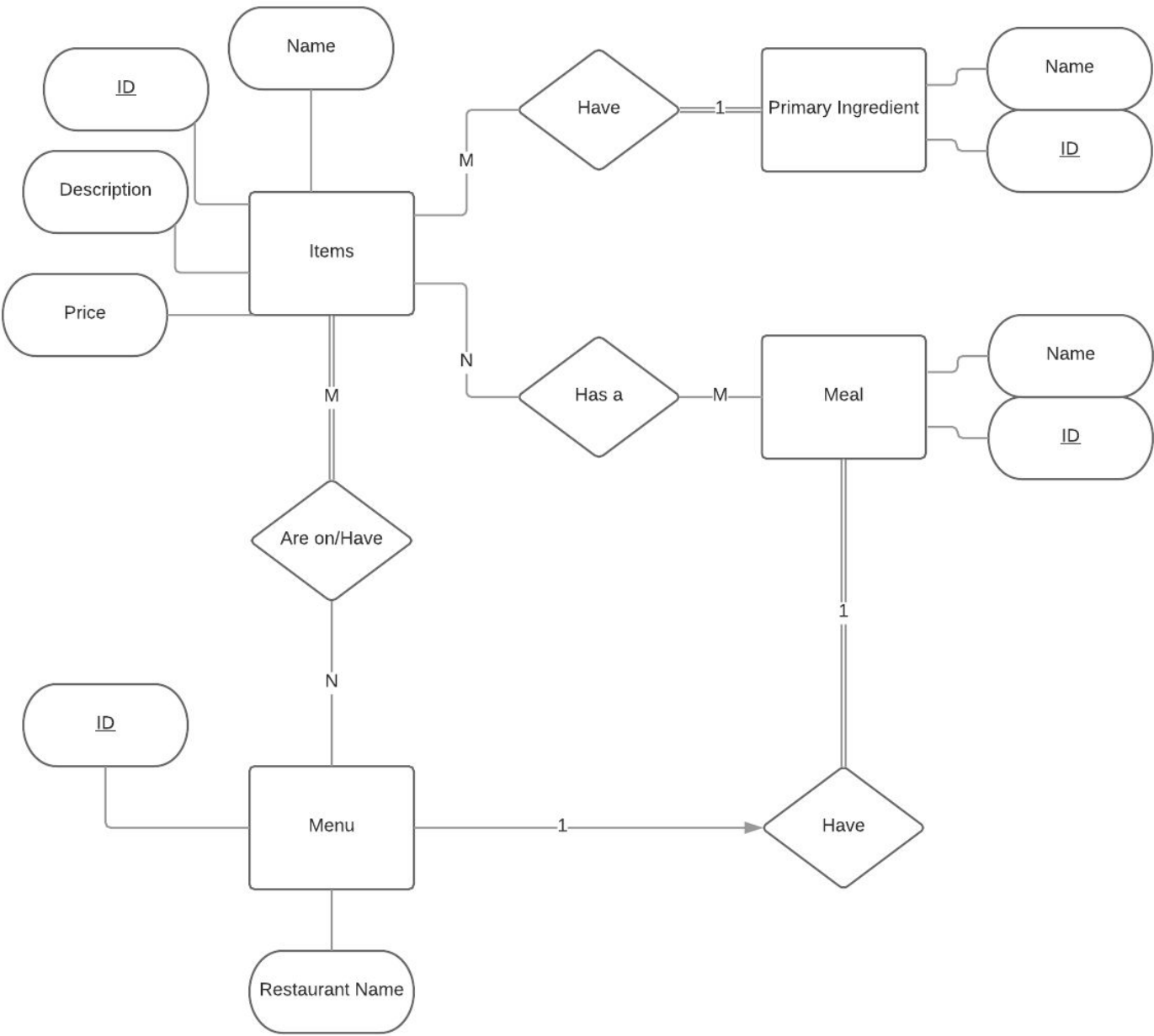
- Items -- This is the primary component of a menu. Items are typically what you would see when looking at a restaurant menu. This consists of all of the attributes that a menu item can contain.
 - ID: This number is automatically assigned to each menu item when added to the database. An auto-incrementing number which is the primary key
 - Name: This is a string name.
 - Description: This is a long string description of the item containing written material to sell the item

- Price: The base price is an 11 FLOAT(5,2) that cannot be less than 0. This value represents the cost of the item if it were in the entree or main section of the menu. Modifiers can be applied to manipulate the price
 - Meal Type: ID of the meal type table
 - Primary Ingredient: An ID of the primary ingredient. For instance, if the primary ingredient is fish, it would fall under the fish category
- Menu Table -- There are different types of menus: breakfast, brunch, lunch and dinner. Some items can be on the dinner menu and lunch menu. Other items can be on the brunch, breakfast or lunch menu. Menus will reference the meal type table.
 - ID: An auto-incrementing number identifying the menu
 - Restaurant Name: Name of restaurant for menu. Allows unique menus for individual restaurants
 - Meal Type: ID of the meal type table
- Primary Ingredient Table -- This is the primary ingredient in the dish such as meat, fish, vegetarian. This is how the meal would be classified under a menu in the main section of the menu.
 - ID: An auto-incrementing number
 - Name: This is a string for the primary ingredient
- Meal Type Table -- This is where the item would fall in the course of a meal. For instance, there might be a dinner salad but also a side salad. They are appetizers, soups, sides, entrees, drinks and desserts.
 - ID: An auto-incrementing number
 - Name: This is a string for the type of meal

The relationships in my database are:

- Items are on menus – This is a many to many relationship. One menu can have many items on it and items can appear on many menus. Items must have menus and menus must have items.
- Menus must have items - A menu cannot exist without items.
- Items must have one primary ingredient – This is a one to one relationship. This is a way of breaking down the menu into easily recognizable categories. Items must have at least one primary ingredient.
- Primary ingredients must be at least one item - A primary ingredient cannot exist without an item therefore that item must exist.
- Items can have many meal types – This is a one to many relationship. Items can be many types. For instance, there can be an appetizer or side of soup but also a main course. This category would also add or subtract a multiplier to the items base price.
- Menus have different meal types -- This is a one to many relationship. Menus will have different meal types. For instance, breakfast will not have dessert options, but dinner will.

Entity-Relationship Diagram:



Schema:

