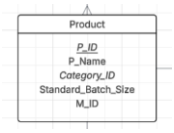
CSC 540
PROJECT FINAL REPORT
GROUP 19

Nico Field (ndfield)
Shaan Chandra (schand33)
Vedantt Koul (vkoul)

NOTE:
The ER Diagram can be found in the glossary (Figure 1).
In the ER Diagram, an italicized attribute means that it's a foreign key.
The table describing the normal forms of the tables is also located in the glossary (Table 1)

EXPLANATION FOR ALL TABLES (19 total):

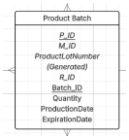
1. Product:



This table keeps the record of products.
CONSTRAINTS: Every attribute is not null. Standard_Batch_Size should be greater than 0.

FUNCTIONAL DEPENDENCIES:
P_ID -> P_Name,Category_ID,Standard_Batch_Size,M_ID

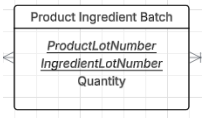
2. Product Batch:



The table keeps the track of the different batches of the products that a manufacturer makes. Quantity here refers to the in hand quantity of the product batch. Each product batch is created using a specific recipe. The ProductLotNumber is an auto generated field and thus we aren't accounting it in the functional dependencies or the verification of normal forms.
CONSTRAINTS: Every attribute is not null. Quantity should be greater than 0.

FUNCTIONAL DEPENDENCIES:
P_ID,Batch_ID->M_ID,ProductLotNumber,R_ID,Quantity

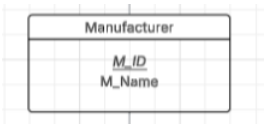
3. Product Ingredient Batch:



The sole purpose of this table is to keep a track of what ingredient batch (IB_ID) was ordered to make a specific product batch along with its quantity.
CONSTRAINTS: Every attribute is not null and quantity should be greater than 0. The Ingredient lot should not be expired. ProductLotNumber must be unique

FUNCTIONAL DEPENDENCY:
ProductLotNumber, IngredientLotNumber -> Quantity

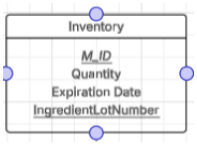
4. Manufacturer:



The Manufacturer table keeps a track of all the manufacturer's manufacturer IDs and names. The manufacturer creates products in batches and maintains an inventory.
CONSTRAINTS: Every attribute is not null.

FUNCTIONAL DEPENDENCIES:
M_ID->M_Name

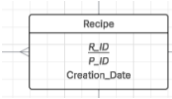
5. Inventory:



The inventory keeps track of what and how many Ingredient Lots were ordered (IngredientLotNumber and Quantity) along with the expiration dates (Expiration Date) of these set ingredients. It also keeps the track of the Manufacturer whose inventory it is.
CONSTRAINTS: Every attribute is not null. Quantity should be greater than 0. The expiration date must be at least 90 days from today.

FUNCTIONAL DEPENDENCIES:
M_ID,IngredientLotNumber -> Quantity, ExperationDate

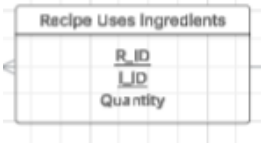
6. Recipe:



Every product that's made follows a specific recipe that's specified by the manufacturer. The recipe table keeps a track of the ID of the recipe (R_ID) and the ID of the product that it makes (P_ID) along with the Creation_Date. The recipe with the latest creation date is always chosen in the case of multiple recipes for the same product. The recipe also refers to the do not combine table to check if it's not containing any pair of ingredients that are not allowed.
CONSTRAINTS: Every attribute is not null.

FUNCTIONAL DEPENDENCIES:
R_ID->P_ID,Creation_Date

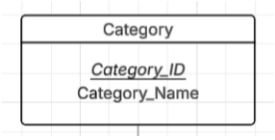
7. Recipe Uses Ingredients:



This table tells what quantity of what ingredient a specific recipe needs.
CONSTRAINTS: Every attribute is not null. Quantity should be greater than 0.

FUNCTIONAL DEPENDENCIES:
R_ID,I_ID->Quantity

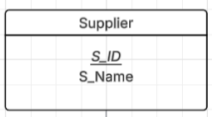
8. Category:



The category table tells what category ID refers to what category (Dinner, Slides or Dessert)
CONSTRAINTS: Every attribute is not null. Category_Name should be either "Desserts", "Dinner" or "Sides".

FUNCTIONAL DEPENDENCIES:
Category_ID->Category_Name

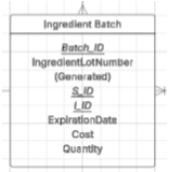
9. Supplier:



The supplier table, as the name suggests, keeps track of the IDs of all the suppliers (S_ID) along with the names of the supplier (S_Name). All a supplier does is maintain an Ingredient Batch.
CONSTRAINTS: Every attribute is not null.

FUNCTIONAL DEPENDENCIES:
S_ID->S_Name

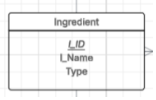
10. Ingredient Batch:



The ingredients are supplied by the supplier in batches called ingredient batches, which are then consumed by the manufacturer to produce the product batches. Every ingredient is made in a batch. Based on the expiration date and the quantity of the batch, the manufacturer can find out which supplier creates the suitable batches that would meet the needs of the respective recipe.
CONSTRAINTS: Every attribute is not null. Quantity and cost should be greater than 0. The expiration date must be at least 90 days from today. IngredientLotNumber must be unique

FUNCTIONAL DEPENDENCIES:
Batch_ID,S_ID,I_ID -> IngredientLotNumber, ExpirationDate, Cost, Quantity

11. Ingredient:



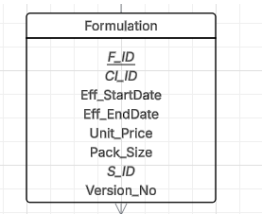
The ingredient table keeps track of the unique identifiers of all the ingredients that exist (I_ID), the name of the ingredient (I_Name) and the type. Every ingredient could be either an atomic ingredient or a compound ingredient. This distinction is made by using an ISA generalization. The ingredients are generalized into Compound Ingredients and Atomic Ingredients.
CONSTRAINTS: Every attribute is not null. Type should be either "Atomic" or "Compound". I_Name must be unique.

FUNCTIONAL DEPENDENCIES:
I_ID->I_Name,Type

12. Compound Ingredient:

Every compound ingredient has a formulation that's essentially an analog of the recipe table on the manufacturer's side. A compound ingredient is formed by atomic ingredients.
CONSTRAINT: CI_ID should be a valid I_ID from the Ingredient table of the type "Compound".

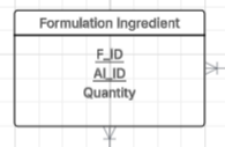
13. Formulation:



The formulation keeps track of the identifier of the formula (F_ID), the ID of the compound ingredient that it's making along with the other details. It has the S_ID which allows every supplier to have their own formulation of a specific compound product.
CONSTRAINTS: Every attribute is not null. Unit price and pack size should be greater than 0. The effective periods should not overlap. (CI_ID, Version_No) must be unique

FUNCTIONAL DEPENDENCIES:
FI_D -> CI_ID,Eff_StartDate, Eff_EndDate, Unit_Price, Pack_Size, S_ID, Version_No
S_ID,CI_ID,Version_No -> F_ID,Eff_StartDate, Eff_EndDate, Unit_Price, Pack_Size
S_ID,CI_ID,Eff_StartDate,Eff_EndDate -> F_ID, Version_No, Unit_Price, Pack_Size

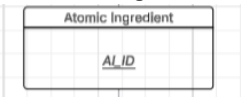
14. Formulation Ingredient



This table keeps a track of what atomic ingredients and how many of them are needed to make a specific compound ingredient.
CONSTRAINTS: Every attribute is not null. Quantity should be greater than 0.

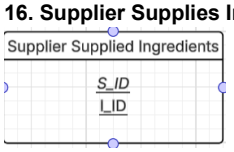
FUNCTIONAL DEPENDENCIES:
F_ID,AI_ID->Quantity

15. Atomic Ingredient



The table keeps track of all the atomic ingredients and their abstracted identifiers (AI_ID).
CONSTRAINT: AI_ID should be a valid I_ID from the Ingredient table of the type "Atomic".

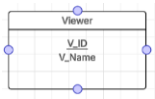
16. Supplier Supplies Ingredients:



The table keeps a track of what ingredients are supplied by a supplier.
CONSTRAINTS: Every attribute is not null.

AUXILIARY TABLES:

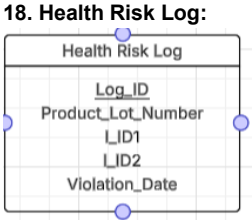
17. Viewer:



This table keeps a track of all the viewers that view the database.
CONSTRAINTS: Every attribute is not null.

FUNCTIONAL DEPENDENCIES:
V_ID->V_Name

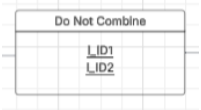
18. Health Risk Log:



Any recipe that violates the do not combine list is added here as a risk log.
CONSTRAINTS: Every attribute is not null.

FUNCTIONAL DEPENDENCIES:
Log_ID->ProductLotNumber,I_ID1, I_ID2, Violation_Date

19. Do Not Combine:



This table keeps a track of all the pairs of ingredients that we can't combine. Doing so raises a trigger and adds the recipe that does so in the health risk log.
CONSTRAINTS: Every attribute is not null. I_ID1 and I_ID2 should be a valid I_ID from the Ingredient table.

GLOSSARY:

ER DIAGRAM:

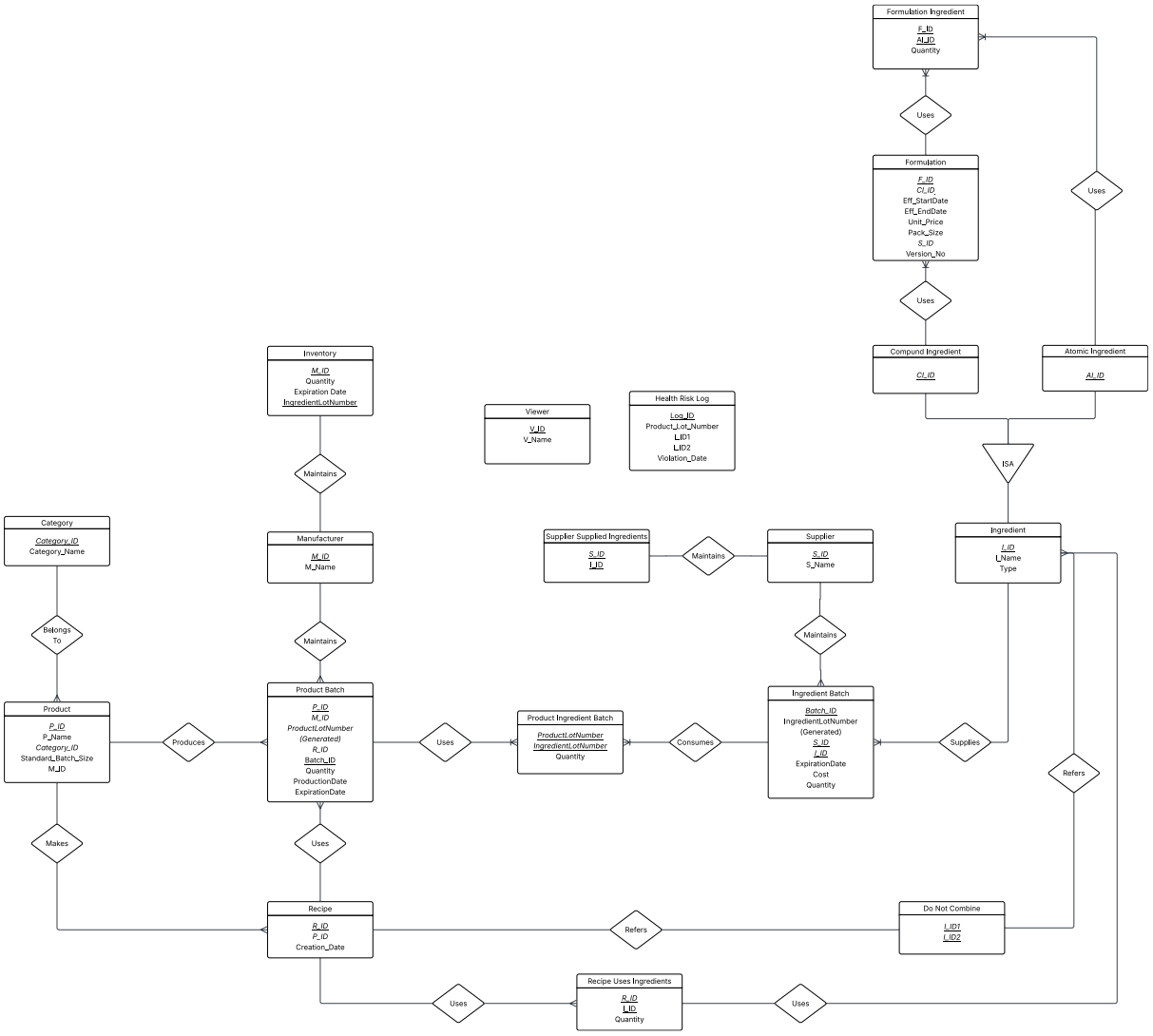


FIGURE 1: ER Diagram

NORMAL FORMS:

TABLE	**NORMAL FORM**
Manufacturer	BCNF
Product Batch	BCNF
Product	BCNF
Recipe	BCNF
Recipe Uses Ingredients	BCNF
Category	BCNF
Product Ingredient Batch	BCNF
Inventory	BCNF
Supplier	BCNF

Ingredient Batch	BCNF
Ingredient	BCNF
Atomic Ingredient	BCNF
Compound Ingredient	BCNF
Formulation	3NF
Formulation Ingredient	BCNF
Do Not Combine	BCNF
Viewer	BCNF
Health Risk Log	BCNF

TABLE 1: Normal Forms of All the Tables