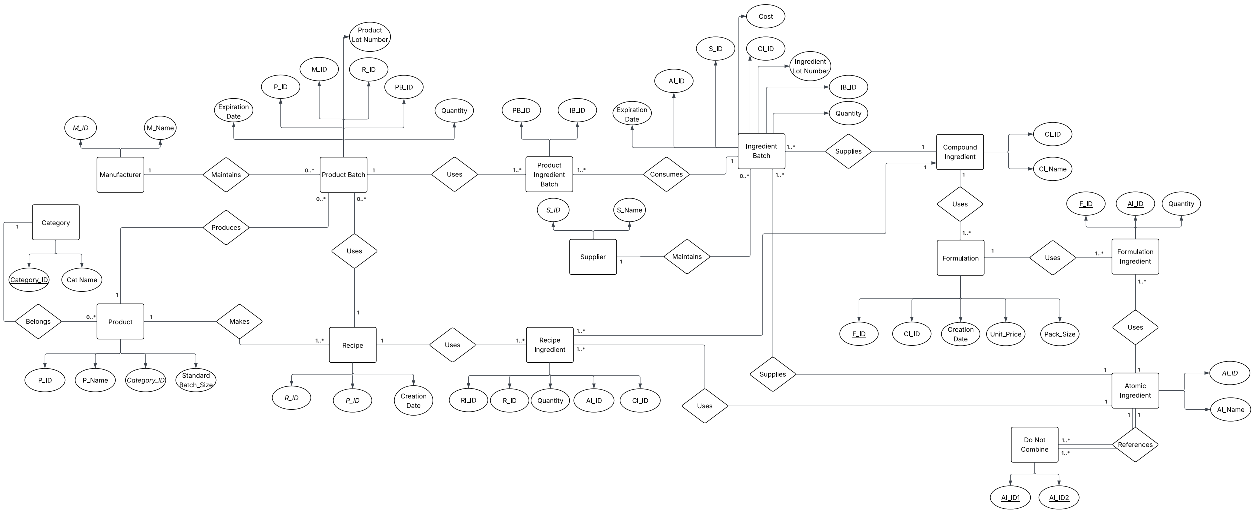
DBMS
PROJECT DELIVERABLE 1

CONTRIBUTORS:

SHAAN CHANDRA (schand33)
NICO FIELD (ndfield)
VEDANTT KOUL (vkoul)

ER DIAGRAM:



(Zoom to view the details)

The ER diagram contains the following tables:

1. **Manufacturer:**
   a. This table has 2 attributes, M_ID and M_Name corresponding to the ID and name of the manufacturer respectively.
   b. The primary key of the table is M_ID
   c. Every manufacturer maintains its inventory by the help of the table ‘Product Batch’. Each product batch is a batch of exactly one product.
   d. The foreign key for the manufacturer maintaining product lot is ‘M_ID’

2. **Product Batch:**
   a. This table includes the M_ID, P_ID, PB_ID, R_ID (referring to manufacturer ID, product ID, product batch ID and recipe ID respectively), Expiration Date, Quantity and Lot ID that’s essentially M_ID+P_ID+PB_ID..
   b. PB_ID is the primary key.
   c. **The quantity refers to the number of packs of the product that the manufacturer has in stock. It could be 0 or a multiple of the standard pack size or a multiple of the attribute ‘Quantity’ in the table ‘Recipe Ingredient’.**
   d. Since each product batch corresponds to a specific product, the table is linked to the ‘Product’ table by the foreign key, ‘P_ID’.
   e. Since each product batch uses a specific recipe for the product that’s decided by the manufacturer, it is linked to the table ‘Recipe’ by the relation ‘Uses’ by the foreign key, ‘R_ID’ that represents the unique recipe ID.
   f. The table is also linked to the table ‘Product Ingredients Batch’ by the relation ‘Uses’ and the foreign key, ‘PB_ID’. This relation is used to track the Ingredient batches that were used to make a certain product based on its recipe.

3. **Recipe:**
    a. This table keeps a track of all the recipe IDs along with their product that it makes and creation dates.
    b. Manufacturers might version their recipes. The recipe ID of each version would be unique. The different versions of the recipes of the same product can be listed by simply querying the recipe_IDs that correspond to a specific product ID.
    c. The table is linked with the product table with the foreign key, 'P_ID' and is linked with the 'Recipe Ingredients' table by the relation 'Uses' with the foreign key, 'R_ID'.

4. **Recipe Ingredient:**
    a. This table keeps a track of the ingredients that a specific recipe would need to make a product.
    b. The table contains the R_ID, RI_ID, the AI_ID, the CI_ID and the Quantity corresponding to the Recipe ID, Recipe Indexing ID and the Ingredient ID of the ingredient that's required and the quantity of the ingredient that's required respectively.
    c. The RI_ID is used to maintain indexing.
    d. CI_ID refers to the ID of the composite ingredients.
    e. AI_ID refers to the ID of the atomic ingredients.
    f. This table is linked to the 'Atomic Ingredients' table by the relation 'Uses' and by the foreign key, 'AI_ID'.
    g. This table is linked to the 'Composite Ingredients' table by the relation 'Uses' and by the foreign key, 'CI_ID'.
    h. **The constraint here is that in every entry, one of the attributes, AI_ID or CI_ID should be NULL. Both of the attributes can not contain not NULL values in AI_ID and CI_ID**

5. **Product:**
    a. This table keeps a track of all the available products that are manufactured by all the manufacturers.
    b. It contains P_ID, P_Name, Category_ID and Standard Batch Size. The P_category refers to the product category and it could either be Dinner, Sides or Desserts.
    c. This table is linked with the 'Category' table using Category_ID as the foreign key

6. **Category:**
    a. This table contains the name of the category (Name) that corresponds to the Category_ID (which is the primary key for the table).
7. **Product Ingredient Batch:**
    a. This table keeps track of what ingredient batches were used to order the ingredients to make a specific batch of product based on the recipe.

   b. It contains PB_ID and IB_ID referring to the product batch ID and ingredient batch ID respectively. This is linked to the ‘Ingredient Batch’ table by the relation ‘Consumes’ and the foreign key, ‘IB_ID’

8. **Supplier:**
   a. This table has 2 attributes, S_ID and S_Name corresponding to the ID and name of the supplier respectively.
   b. The primary key of the table is S_ID
   c. Every supplier maintains its inventory by the help of the table ‘Ingredients Batch’. Each ingredient batch corresponds to a batch of exactly one ingredient.
   d. The foreign key for the supplier maintaining ingredient batch is ‘S_ID’

9. **Ingredient Batch:**
   a. This table includes the S_ID, AI_ID, IB_ID, CI_ID (referring to supplier ID, atomic ingredient ID, ingredient batch ID and composite ingredient ID respectively), Expiration Date, Quantity, Cost and Lot ID which is essentially S_ID+AI_ID/CI_ID+IB_ID.
   b. The constraint here is that in every entry, one of the attributes, AI_ID or CI_ID should be NULL. Both of the attributes can not contain not NULL values in AI_ID and CI_ID.
   c. IB_ID is the primary key.
   d. The quantity refers to the number of packs of the product that the manufacturer has in stock. It could be 0 or a multiple of the standard pack size or a multiple of the attribute ‘Pack Size’ in the table ‘Composition’.
   e. Since each ingredient batch corresponds to a specific ingredient, the table is linked to the ‘Atomic Ingredients’ table by the foreign key, ‘AI_ID’.
   f. For each composite ingredient, the ingredient batch uses a specific composition for the ingredient that’s decided by the supplier. Thus, the table is linked to the table ‘Compound Ingredients’ by the relation ‘Uses’ by the foreign key, ‘CI_ID’ that represents the unique Composite Ingredient ID.

10. **Compound Ingredient:**
    a. This table keeps a track of the composite ingredients. It contains CI_ID and C_Name (corresponding to the composite ingredient ID and composite Ingredient Name respectively).
    b. The CI_ID is an identifier for the composite ingredients.
    c. The table is linked to the ‘Formulation’ table by CI_ID as the foreign key.

11. **Formulation:**
    a. This table keeps a track of the formulation that a supplier is using to make the composite ingredient. It contains CI_ID, Creation Date, Unit Price, Pack Size and F_ID that refers to the formulation ID.
    b. This table is furthermore connected to the ‘Formulation Ingredient’ table by ‘F_ID’ as the foreign key.

12. **Composition Ingredient:**
    - **a.** This table keeps a track of the ingredients that a specific recipe would need to make a product.
    - **b.** The table contains the F_ID, the AI_ID and the Quantity corresponding to the Formulation ID, the Atomic Ingredient ID of the ingredient that's required and the quantity of the ingredient that's required respectively.
    - **c.** This table is linked to the 'Ingredients' table by the relation 'Uses' and by the foreign key, 'AI_ID'.

13. **Atomic Ingredients:**
    - **a.** This table keeps a track of all the atomic ingredients. It includes the AI_ID and I_Name corresponding to Atomic Ingredient ID and Atomic Ingredient Name respectively.
    - **b.** The primary key for the table is AI_ID.

14. **Do Not Combine:**
    - **a.** This table keeps a track of the Atomic Ingredients that can't be combined in a recipe. It contains 2 AI_IDs, that act as a foreign key to the 'Atomic Ingredients' table.
    - **b.** We could furthermore warn the manufacturer if they use any of the forbidden combinations of ingredients.