

CSC540 –Project Final Report

TEAM MEMBERS

NAME	Unity ID
PRANAV VARENYA VENKATESH	Pvareny
CHRISTOPHER DILLON MICHELS	Cdmichels
ANANDTEERTHA RAMESH RAO	Arrao6
EDWARD FENG	Sfeng9

Title: Inventory Management for a Prepared/Frozen Meals Manufacturer

1. Introduction

This report documents the design and implementation of the DB system for Fresh Frost Meals which is a meals manufacturer. This includes supply chain, tracking products, ingredients, suppliers, manufacturers, and batches. The schema consists of declarative constraints, triggers, and stored procedures, fulfilling all requirements for the Supplier, Manufacturer, and Viewer roles.

2. Design Evolution: From Preliminary to Final ER Model

Our initial ER diagram had the core entities and their relationships. The changes to the final implemented schema has a few changes

Key refinements from the initial design are:

- **Composite Ingredient Handling:** Instead of separate Atomic/Composite tables, we use Ingredient.type ENUM with FormulationMaterial for compound composition
- **Formulation Versioning:** Added IngredientFormulation with version control and validity periods for supplier pricing
- **Role-Based User Management:** Consolidated Manufacturer/Supplier into UserDetails with role_code ENUM
- **Batch Separation:** Split into IngredientBatch and ProductBatch with lot-based tracking
- **Recipe Management:** Added RecipePlan for product formulation versioning
- **Safety Rules:** Added IngredientIncompatibility for conflict prevention

3. Functional Dependencies that Influenced Design

FDs that influence the design are:

- $\text{Product.id} \rightarrow \text{Product.name}, \text{Product.number}, \text{Product.category_id}, \text{Product.standard_batch_units}$

- Ingredient.id → Ingredient.name, Ingredient.type
- (Ingredient.id, UserDetails.id, version_number) → unit_price, pack_size, validity dates in IngredientFormulation
- (UserDetails.id, Product.id) → relationship in ManufacturerProduct
- ProductBatch.lot_number → all batch attributes including costs
- (formulation_id, ingredient_id) → quantity in FormulationMaterial
- (product_id, ingredient_id) → quantity in ProductBOM

4. Comments about Normalization

DB schema has been designed to meet at least 3NF although the majority refer to BCNF.

All base tables

ie....(Category, Product, Manufacturer, Supplier, AtomicIngredient, CompositeIngredient, Formulation, Batch) are in BCNF.

The other entities

(AtomicIngredientSupplier, CompositeIngredientSupplier, ManufacturerProduct) are also in BCNF. A summary of the normalization analysis for all tables is provided in Appendix A.

5. Description of Constraints

a) Database-Level Constraints (SQL DDL)

- **Primary & Unique Keys:** All entities have PRIMARY KEY constraints. Uniqueness enforced with constraints like UNIQUE(name, number) on Product and UNIQUE(ingredient_id, supplier_id, version_number) on IngredientFormulation
- **Foreign Keys:** ON UPDATE CASCADE for primary keys and ON DELETE RESTRICT/CASCADE based on business rules
- **Check Constraints:** Quantity ≥ 0 , pack_size > 0 , expiration_date $>$ production_date, produced_quantity > 0
- **ENUM Constraints:** role_code, ingredient_type with restricted values

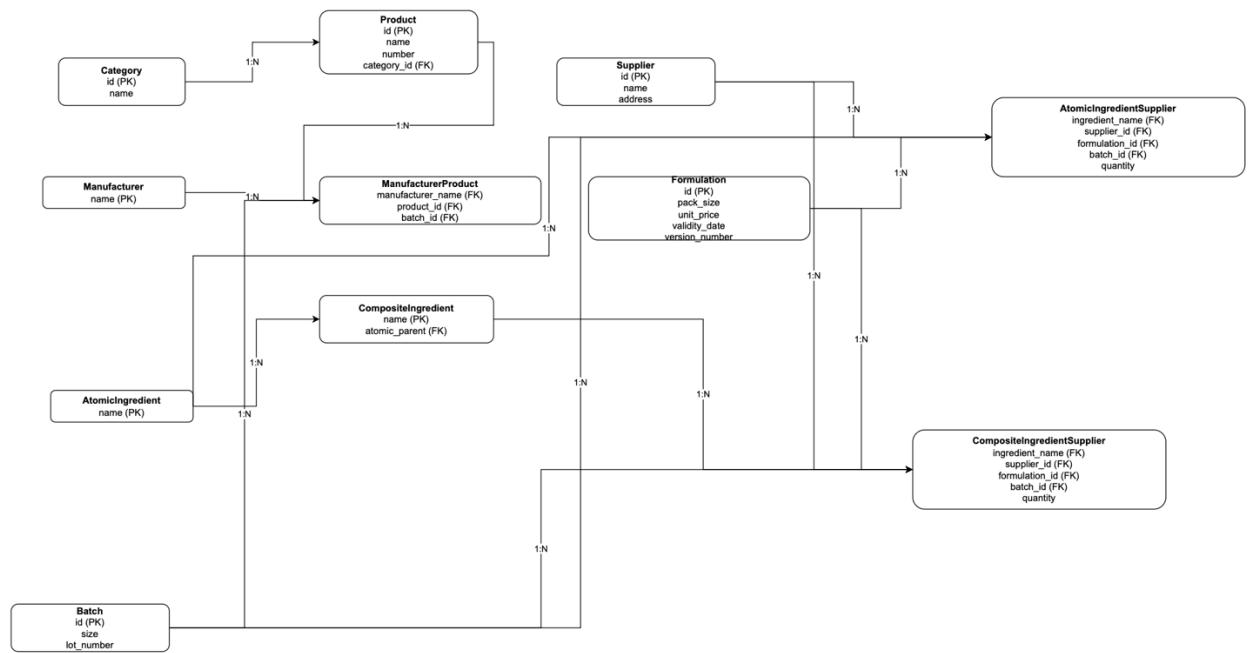
b) Application-Level Constraints (Triggers/Procedures)

- Structured Lot Numbering: BEFORE INSERT triggers generate lot_number as ingredient_id-supplier_id-batch_id and product_id-manufacturer_id-batch_id
- 90-Day Expiration Rule: BEFORE INSERT trigger on IngredientBatch validates expiration_date \geq CURDATE() + 90 days
- Inventory Management: Triggers prevent consumption of expired lots and ensure sufficient quantity
- Incompatibility Enforcement: BEFORE INSERT trigger on IngredientConsumption prevents conflicting ingredients in same product batch
- Role Validation: Triggers ensure ManufacturerProduct only references MANUFACTURER users and IngredientFormulation only references SUPPLIER users
- Cost Recalculation: Stored procedures automatically update batch_total_cost and unit_cost after ingredient consumption

Appendix A: Normalization

UserDetails	id	BCNF	All attributes are dependent on the primary key.
Category	id	BCNF	All attributes are dependent on the primary key.
Product	id	BCNF	name, number, and category_id are fully functionally dependent on id. The UNIQUE (name, number) is an alternate key.
Ingredient	id	BCNF	The key is a single attribute.
IngredientFormulation	id	BCNF	All attributes are dependent on the primary key.
IngredientBatch	lot_number	BCNF	All attributes are dependent on the primary key.
ProductBatch	lot_number	BCNF	All attributes are dependent on the primary key.
RecipePlan	plan_id	BCNF	All attributes are dependent on the primary key.
ManufacturerProduct	(manufacturer_id, product_id)	BCNF	The composite primary key uniquely determines the relationship.
ProductBOM	(product_id, ingredient_id)	BCNF	The composite primary key uniquely determines quantity.
FormulationMaterial	(formulation_id, ingredient_id)	BCNF	The composite primary key uniquely determines quantity.
IngredientConsumption	(product_lot_number, ingredient_lot_number)	BCNF	The composite primary key uniquely determines consumed_quantity_oz.

Appendix B: Final ER Diagram



Zybooks Notation used

Updated ER:

