# **Food Supply Chain — Traceability & Recall Data Hub**

## **Database Design Document**

### **1) Business Problem & Goals**

#### **1.1 Overview**

**Food manufacturing must track materials from intake to delivery while safeguarding quality and responding rapidly to recalls. The data model provides a single source of truth for suppliers, materials, lots, QA results, storage, shipments, and recalls so that operations, QA, and compliance teams can work from consistent, auditable data.**

#### **1.2 Objectives**

* **Maintain end‑to‑end traceability of every lot across transformations and movements.**
* **Enforce data quality for identifiers, units of measure, statuses, and temperature limits.**
* **Distinguish on‑hand vs reserved inventory to prevent double allocation.**
* **Gate outbound shipments on QualityTest results and capture cold‑chain details.**
* **Produce clear, regulator‑ready reports with a full AuditLog of changes.**

#### **1.3 Key Use Cases**

* **Identify all customers and shipments affected by a specific lot recall.**
* **Trace a shipped lot back to its supplier inputs (upstream provenance).**
* **Monitor inventory aging, dwell time, and lots nearing expiry.**
* **Compare supplier lead times, pricing, and QA pass rates over time.**

### **2) Scope & Assumptions**

#### **2.1 In Scope**

* **Master data: Supplier, Material, Warehouse, Customer.**
* **Transactions: Lot, QualityTest, Inventory, Shipment, Recall.**
* **Linkages: SupplierMaterial, ShipmentLot, RecallLot, Genealogy.**
* **Governance: AuditLog, statuses, effective dating, and basic temperature handling.**

#### **2.2 Out of Scope**

* **Detailed manufacturing routing/operations beyond lot genealogy.**
* **Advanced IoT telemetry warehousing; only summarized/attached logs are stored.**
* **Pricing contracts beyond unit price at the supplier‑material level.**

#### **2.3 Assumptions**

* **Timestamps are stored in UTC. Natural business codes (e.g., lot\_number) are unique.**
* **"Soft delete" via is\_active where needed; transactional history is never removed.**
* **Controlled vocabulary for statuses and test types is managed application‑side.**

### **3) Entities & Attributes (ERD‑Aligned)**

#### **3.1 Master Data Entities**

* **Supplier — vendor profile (code, name, country, contact, rating).**
* **Material — item definition (name, UoM, shelf life, temperature bounds).**
* **Warehouse — storage site (code, name, address, capacity, temperature zones).**
* **Customer — sold‑to account (code, name, type, ship‑to address, contact).**

#### **3.2 Transactional Entities**

* **Lot — batch instance linked to a supplier and material; tracks qty, dates (production/expiry/receipt), and status.**
* **QualityTest — test record for a lot (type, method, result value, pass flag, tester/device, timestamp).**
* **Inventory — stock position per lot and warehouse/location; separates on‑hand vs reserved, status, last movement, dwell days.**
* **Shipment — outbound movement from a warehouse (number, carrier, ETD/ATD, tracking, status).**
* **Recall — recall event metadata (number, date, reason, severity, status, regulator, initiator, resolution date).**

#### **3.3 Associative/Link Entities**

* **SupplierMaterial — approved sourcing (supplier ↔ material) with lead time, unit price, effective dates.**
* **ShipmentLot — what lots/quantities were placed on a shipment; line price and optional temperature log.**
* **RecallLot — mapping of recalls to lots with affected quantity/customers notes.**
* **Genealogy — parent/child lot links created by splitting/combining/processing; includes qty used, yield %, waste amount, and timestamp.**

### **4) Relationships & Cardinalities**

#### **4.1 One‑to‑Many Relationships**

* **Supplier → Lot; Material → Lot; Lot → QualityTest; Warehouse → Inventory; Warehouse → Shipment; Customer → Shipment (as destination).**

#### **4.2 Many‑to‑Many via Associative Tables**

* **Supplier ↔ Material via SupplierMaterial.**
* **Shipment ↔ Lot via ShipmentLot.**
* **Recall ↔ Lot via RecallLot.**

#### **4.3 Special Rules (Genealogy, Recalls, QA)**

* **Genealogy models lot transformations (M:N); parents can produce multiple children and children can consume multiple parents.**
* **Quality gate: a lot cannot be shipped unless its latest QualityTest is a pass (or a documented override exists).**
* **Recall impact uses RecallLot to discover affected Inventory and historical ShipmentLot rows.**

### **5) Key Database Design Decisions**

#### **5.1 Keys & Indexing Strategy**

* **Surrogate integer PKs on all tables; natural business identifiers (e.g., supplier\_code, lot\_number, shipment\_number, warehouse\_code) are UNIQUE.**
* **Index all FKs; add composites for common filters (e.g., Inventory(lot\_id, warehouse\_id, status)).**

#### **5.2 Status/Soft‑Delete Strategy**

* **Use status enums (e.g., Draft/Active/Hold/Closed) and is\_active flags where appropriate. No hard deletes for transactional data.**

#### **5.3 Quality Gating & Cold‑Chain Handling**

* **Enforce QA gate before shipment. Store min/max temperature thresholds on Material/Warehouse; attach summarized temperature logs on ShipmentLot when available.**

#### **5.4 Audit & Effective‑Dating Approach**

* **AuditLog captures table name, record id, action, user, timestamp, and old/new JSON payloads.**
* **SupplierMaterial supports effective dating to track historical price/lead‑time changes.**

### **6) Data Quality & Governance**

#### **6.1 Controlled Vocabularies & Validation**

* **Validate codes (regex/length), units of measure, and status sets. Enforce positive quantities and realistic temperatures.**

#### **6.2 Constraints & Checks**

* **Inventory.quantity\_on\_hand ≥ quantity\_reserved ≥ 0 via CHECKs/triggers.**
* **Genealogy balance: sum(child.qty\_used) ≤ parent available quantity at the time of processing.**

#### **6.3 Change Management & Audit Trail**

* **All critical updates generate an AuditLog entry; sensitive updates require elevated roles and justification fields captured in the log.**

### **7) Reporting & Analytics**

#### **7.1 Recall Impact & Downstream Exposure**

* **Starting from Recall, join RecallLot → Inventory for what remains and ShipmentLot for who received affected lots, grouped by customer and shipment.**

#### **7.2 Upstream Provenance/Lineage**

* **For any shipped lot, walk Genealogy recursively to list all contributing parent lots, their suppliers, and QA outcomes.**
* **For any shipped lot, walk Genealogy recursively to list all contributing parent lots, their suppliers, and QA outcomes.**

#### 

#### 

#### 

#### 

#### **7.3 Inventory Aging & Dwell Time**

**Calculate days‑on‑hand and highlight lots near expiry; report dwell time by warehouse and location to optimize rotation.**

### **8) Risks & Future Extensions**

#### **8.1 Known Risks**

* **Temperature telemetry may outgrow summarized storage; consider a dedicated time‑series table if granularity increases.**
* **Complex recalls require performant recursive lineage queries; plan for materialized views or graph extensions if needed.**

#### **8.2 Potential Extensions**

* **Add ProductionOrder/Operation entities for detailed manufacturing steps.**
* **Add Returns/Disposition for reverse logistics and scrap handling.**
* **Add Users/Roles for row‑level security and approval workflows.**

### 

### 

### 

### 

### 

### 

### 

### 

### **9) Submission Summary (Links)**

#### **9.1** [**https://github.com/riyanshikedia10/DMDD\_Group-1**](https://github.com/riyanshikedia10/DMDD_Group-1)

***Add your shared ERD URL here.***

#### **9.2 Design Document (This File —** [**https://drive.google.com/file/d/1KDXv9hpZm3MEjyPB1ga3QTws0VhJCknd/view?usp=sharing**](https://drive.google.com/file/d/1KDXv9hpZm3MEjyPB1ga3QTws0VhJCknd/view?usp=sharing)**)**

## 

## Team Members

1. Member 1: Riyanshi Kedia - 002038686

2. Member 2: Prerana Gireesha - 002088156

3. Member 3: Hardi Atulbhai Virani - 002062268

4. Member 4: Harini Thirugnanasambandham - 002316667