## **Logical Data Model Submission**

**Course:** DMDD  
 **Team Name:** Group 1  
 **Submission Date:** October 31, 2025

## **TEAM MEMBERS**

1. Riyanshi Kedia - 002038686
2. Prerana Gireesha - 002088156
3. Hardi Atulbhai Virani - 002062268
4. Harini Thirugnanasambandham - 002316667

## **DOCUMENT LINKS**

All documents have been uploaded and set to "public viewable" access:

**Google Drive:** <https://drive.google.com/file/d/1qDcIFnNC3rkqpT1paI_hmqpELD3OyFxY/view>

**GitHub Repository:** <https://github.com/riyanshikedia10/DMDD_Group-1>

# **EXECUTIVE SUMMARY**

## **Project Overview**

This e-commerce database system manages the complete product lifecycle from supplier sourcing through customer orders, inventory management, shipping, payment processing, and compliance reporting. The system includes 23 normalized entities covering all aspects of online retail operations.

## **System Statistics**

* **Total Entities:** 23 entities
* **Core Business Entities:** 10 (CUSTOMER, ORDER, PRODUCT, SUPPLIER, etc.)
* **Junction Tables:** 6 (eliminates all M:N relationships)
* **Supporting Entities:** 7 (ADDRESS, INVENTORY, CATEGORY, etc.)
* **Total Relationships:** 20+ relationships
* **Normalization Level:** Third Normal Form (3NF) achieved throughout

## **Key Changes from Initial ERD**

Based on instructor feedback, we made the following improvements:

1. **Converted from Conceptual to Logical Model** - Added all primary keys (PK) and foreign keys (FK) with proper notation
2. **Fixed Supplier-Product Relationship** - Created clear link between SUPPLIER and PRODUCT entities
3. **Clarified Customer-Order Flow** - Documented complete order process: CUSTOMER → ORDER → ORDERITEM → PRODUCT
4. **Eliminated All M:N Relationships** - Created 6 junction tables
5. **Achieved 3NF** - Removed all transitive dependencies and multivalued attributes
6. **Added 3-Column Format** - PK/FK | Attribute Name | Data Type for all entities
7. **Expanded System** - Added 15 new entities for comprehensive functionality

# **CRITICAL CHANGES FROM INITIAL ERD**

## **CRITICAL FIX #1: Supplier-Product Relationship**

### **Instructor Feedback**

"Materials are not clearly linked to their suppliers — the model doesn't show which supplier provides each material."

### **Problem in Initial ERD**

The relationship between suppliers and products was either missing or unclear, making it impossible to track product sourcing.

### **Solution Implemented**

Created a **one-to-many relationship** from SUPPLIER to PRODUCT:

SUPPLIER (1) ---- supplies ----> (Many) PRODUCT

### **Technical Implementation**

**SUPPLIER Entity:**

PK | SupplierID | INT

| Name | VARCHAR(100)

| ContactPerson | VARCHAR(50)

| Email | VARCHAR(100)

| Phone | VARCHAR(20)

| Address | TEXT

**PRODUCT Entity (updated):**

PK | ProductID | INT

FK | CategoryID | INT

FK | SupplierID | INT ← CRITICAL: Added to link to SUPPLIER

| Name | VARCHAR(100)

| Description | TEXT

| Price | DECIMAL(10,2)

| StockQuantity | INT

| ReorderLevel | INT

### **What We Changed**

* Added SupplierID as Foreign Key in PRODUCT table
* FK references SupplierID Primary Key in SUPPLIER table
* **Cardinality:** One supplier can supply many products (1:N)
* **Business Rule:** Each product must have exactly one primary supplier

### **Diagram Changes**

* Added visible relationship line with diamond shape labeled "Supplies"
* Marked cardinality as 1:N using ER notation (||—<)
* Used 3-column format throughout

## **CRITICAL FIX #2: Customer-Order Relationship**

### **Instructor Feedback**

"It's unclear what the customer orders. Please clarify what the customer is ordering in the model."

### **Problem in Initial ERD**

The ordering process was ambiguous - it was unclear who the customer was and what they were ordering.

### **Solution Implemented**

Created a **clear order flow** with proper junction table:

CUSTOMER → places → ORDER → contains → ORDERITEM → for → PRODUCT

**Flow Details:**

1. CUSTOMER (1) ---- places -----> (N) ORDER
2. ORDER (1) ---- contains -----> (N) ORDERITEM (junction table)
3. ORDERITEM (N) ---- for -----> (1) PRODUCT

### **Entity Structures**

**CUSTOMER Entity:**

PK | CustomerID | INT

| FirstName | VARCHAR(50)

| LastName | VARCHAR(50)

| Email | VARCHAR(100)

| Phone | VARCHAR(20)

| DateRegistered | DATE

| Address | VARCHAR(200)

| Status | VARCHAR(20)

**ORDER Entity:**

PK | OrderID | INT

FK | CustomerID | INT ← Links to CUSTOMER

FK | AddressID | INT

| OrderDate | DATE

| RequiredDate | DATE

| Status | VARCHAR(20)

| TotalPrice | DECIMAL(10,2)

| PaymentStatus | VARCHAR(20)

**ORDERITEM Entity (Junction Table):**

PK | OrderItemID | INT

FK | OrderID | INT ← Links to ORDER

FK | ProductID | INT ← Links to PRODUCT

| Quantity | INT

| Price | DECIMAL(10,2)

### **What We Clarified**

1. **Customers** are the primary users who place orders
2. Customers order **PRODUCTS** (not materials or other items)
3. Each order can contain **multiple products** (via ORDERITEM junction)
4. Each order item references **exactly one product**
5. Order total is **calculated from order items** (sum of quantity × price)
6. **ORDERITEM eliminates M:N** relationship between ORDER and PRODUCT

## **CRITICAL FIX #3: Conceptual to Logical Conversion**

### **Instructor Feedback**

"The model is a mix of conceptual and logical elements."

### **Changes Made**

**A. Added All Primary Keys (PK)**

Every entity now has a clearly marked primary key:

* CUSTOMER → CustomerID (PK)
* ORDER → OrderID (PK)
* ORDERITEM → OrderItemID (PK)
* PRODUCT → ProductID (PK)
* SUPPLIER → SupplierID (PK)
* INVOICE → InvoiceID (PK)
* SHIPMENT → ShipmentID (PK)
* PAYMENT → PaymentID (PK)
* And 15 more entities...

**Format Used:** PK | AttributeName | DataType

**B. Added All Foreign Keys (FK)**

Every relationship now has corresponding foreign keys:

* **ORDER:** CustomerID (FK), AddressID (FK)
* **ORDERITEM:** OrderID (FK), ProductID (FK)
* **PRODUCT:** SupplierID (FK), CategoryID (FK)
* **SHIPMENT:** OrderID (FK)
* **INVOICE:** OrderID (FK)
* And many more...

**C. Specified All Data Types**

Every attribute has explicit data type:

* **INT** - IDs, quantities, counts
* **VARCHAR(n)** - Names, emails, short text
* **DATE** - Dates without time
* **DATETIME** - Dates with time
* **DECIMAL(10,2)** - Money, prices
* **TEXT** - Long descriptions

**D. Added Cardinality Notation**

All relationships show ER cardinality:

* || = Exactly one (mandatory)
* ○| = Zero or one (optional)
* |< = One to many
* ○< = Zero to many

# **ELIMINATED ALL M:N RELATIONSHIPS**

## **Junction Tables Created**

We created **6 junction tables** to eliminate all many-to-many relationships:

### **1. ORDERITEM Junction**

**Eliminates:** ORDER ↔ PRODUCT (M:N)

**Before:** Direct M:N relationship between ORDER and PRODUCT

**After:** ORDER (1) → ORDERITEM (N) → (1) PRODUCT

**Structure:**

PK | OrderItemID | INT

FK | OrderID | INT

FK | ProductID | INT

| Quantity | INT

| Price | DECIMAL(10,2)

**Why:** Stores quantity and price per order item, maintains historical pricing

### **2. CARTITEM Junction**

**Eliminates:** CART ↔ PRODUCT (M:N)

**Before:** Cart could contain many products directly

**After:** CART (1) → CARTITEM (N) → (1) PRODUCT

**Structure:**

PK | CartItemID | INT

FK | CartID | INT

FK | ProductID | INT

| Quantity | INT

**Why:** Allows multiple products in cart with individual quantities

### **3. REPORT\_ITEM Junction**

**Eliminates:** TRACEABILITY\_REPORT ↔ FOODBATCH (M:N)

**Structure:**

PK | ReportItemID | INT

FK | ReportID | INT

FK | BatchID | INT

| Notes | VARCHAR(200)

### **4. REPORT\_SHIPMENT Junction**

**Eliminates:** TRACEABILITY\_REPORT ↔ SHIPMENT (M:N)

**Structure:**

PK | ReportShipmentID | INT

FK | ReportID | INT

FK | ShipmentID | INT

| Notes | VARCHAR(200)

### **5. REPORT\_REVIEW Junction**

**Eliminates:** TRACEABILITY\_REPORT ↔ REVIEW (M:N)

**Structure:**

PK | ReportReviewID | INT

FK | ReportID | INT

FK | ReviewID | INT

| CreatedAt | DATETIME

### **6. ORDER\_ALLOCATION Junction**

**Eliminates:** ORDER ↔ INVENTORY (M:N)

**Structure:**

PK | AllocationID | INT

FK | OrderID | INT

FK | InventoryID | INT

| AllocatedQty | INT

| AllocationDate | DATE

## **Summary of Junction Tables**

| **Junction Table** | **Eliminates M:N Between** | **Purpose** |
| --- | --- | --- |
| ORDERITEM | ORDER ↔ PRODUCT | Order line items with pricing |
| CARTITEM | CART ↔ PRODUCT | Shopping cart items |
| REPORT\_ITEM | REPORT ↔ FOODBATCH | Batch traceability |
| REPORT\_SHIPMENT | REPORT ↔ SHIPMENT | Shipment tracking |
| REPORT\_REVIEW | REPORT ↔ REVIEW | Review analysis |
| ORDER\_ALLOCATION | ORDER ↔ INVENTORY | Inventory reservation |

**Result:** All M:N relationships eliminated

# **NORMALIZATION TO 3NF**

## **First Normal Form (1NF) - ACHIEVED**

### **Requirement**

All attributes must contain atomic values with no repeating groups.

### **Violations Fixed**

**1. No Multivalued Attributes**

* **Before:** Phone: "555-1234, 555-5678" (multiple values)
* **After:** Phone: "555-1234" (single value only)

**2. No Repeating Groups**

* **Before:** Supplier1, Supplier2, Supplier3 (repeating columns)
* **After:** Single SupplierID foreign key

**3. No Composite Attributes**

* **Before:** Name: "John Smith" (composite)
* **After:** FirstName: "John", LastName: "Smith" (separated)

**1NF:** All attributes contain single, atomic values  
 No comma-separated values in any field  
 No repeating column patterns  
 Each column contains one type of data

## **Second Normal Form (2NF) - ACHIEVED**

### **Requirement**

Meets 1NF + no partial dependencies (relevant for composite keys).

### **How We Achieved 2NF**

We used **surrogate keys** (single-column primary keys) in all junction tables:

**ORDERITEM Example:**

PK | OrderItemID | INT ← Surrogate key (not composite)

FK | OrderID | INT

FK | ProductID | INT

| Quantity | INT ← Depends on entire PK (OrderItemID)

| Price | DECIMAL(10,2) ← Depends on entire PK

**Why Surrogate Keys:**

* Eliminates partial dependency concerns
* All attributes depend on the entire primary key
* More flexible for future changes
* Industry best practice

**2NF:** All tables meet 1NF  
 No partial dependencies exist  
 All non-key attributes fully depend on primary key

## **Third Normal Form (3NF) - ACHIEVED**

### **Requirement**

Meets 2NF + no transitive dependencies (no non-key attribute depends on another non-key attribute).

### **Transitive Dependencies Removed**

**1. City/State/Zipcode Dependency**

**Violation:**

* Customer has Zipcode, City, State
* City depends on Zipcode (transitive dependency)
* State depends on Zipcode (transitive dependency)

**Fixed:**

* Combined into single Address field: VARCHAR(200)
* Alternatively, could create separate ZIPCODE table

**2. Calculated Fields**

**Violations Removed:**

* Age (calculated from DateOfBirth) - REMOVED
* ProductName in ORDERITEM (depends on ProductID) - REMOVED

**Kept with Justification:**

* TotalPrice in ORDER (see justification below)

### **Denormalization Justification**

We **kept TotalPrice in ORDER** even though it's calculable because:

1. **Historical Accuracy** - Preserves total even if item prices change later
2. **Performance** - Avoids summing ORDERITEM for every query
3. **Audit Trail** - Business requirement for fixed totals on completed orders
4. **Industry Standard** - Common practice in e-commerce systems

**This is acceptable denormalization with documented business justification.**

**3NF:** All tables meet 2NF  
 No transitive dependencies  
 All calculated fields removed (except justified)  
 No redundant data across tables

# **COMPLETE ENTITY LIST**

## **Core Business Entities (10)**

1. **CUSTOMER** - Customer account information
2. **ORDER** - Order headers and totals
3. **ORDERITEM** - Order line items (Junction table)
4. **PRODUCT** - Product catalog
5. **SUPPLIER** - Vendor/supplier information
6. **CART** - Shopping cart
7. **CARTITEM** - Cart items (Junction table)
8. **INVOICE** - Order invoices
9. **PAYMENT** - Payment transactions
10. **SHIPMENT** - Shipping and delivery

## **Product Management (4)**

1. **CATEGORY** - Product categories
2. **INVENTORY** - Stock tracking by location
3. **FOODBATCH** - Food batch tracking with expiry
4. **PRODUCTION\_BATCH** - Manufacturing batches

## **Supporting Entities (3)**

1. **ADDRESS** - Customer addresses
2. **BOOKING\_SYSTEM** - Resource bookings
3. **ALERT\_LOG** - System notifications

## **Reporting & Traceability (6)**

1. **TRACEABILITY\_REPORT** - Compliance reports
2. **REPORT\_ITEM** - Report batches (Junction)
3. **REPORT\_SHIPMENT** - Report shipments (Junction)
4. **REPORT\_REVIEW** - Report reviews (Junction)
5. **REVIEW** - Product reviews/ratings
6. **ORDER\_ALLOCATION** - Order fulfillment

**Total:** 23 Entities, including 6 Junction Tables

# **KEY BUSINESS RULES**

## **Customer and Order Rules**

1. A customer can place zero or many orders (0:N)
2. An order must belong to exactly one customer (1:1)
3. An order must contain at least one order item (1:N)
4. Each order item references exactly one product (N:1)
5. Order cannot be modified once status is "Completed"
6. Customer must have unique email address

## **Product and Supplier Rules**

1. Each product must have exactly one primary supplier (N:1) **← CRITICAL**
2. A supplier can supply zero or many products (1:N) **← CRITICAL**
3. Products cannot be deleted if in any order
4. Stock updated when orders processed
5. Alert when stock < reorder level

## **Order Processing Rules**

1. Order total = sum of (OrderItem quantity × price)
2. Each order generates exactly one invoice (1:1)
3. Invoice auto-generated when order status = "Completed"
4. Order can have multiple payments (split payments)
5. Order cannot ship until payment = "Paid"

## **Shopping Cart Rules**

1. Each customer has exactly one cart (1:1)
2. Cart converted to order on checkout
3. Cart items → Order items on checkout
4. Cart emptied after successful checkout

## **Food Safety Rules**

1. Food products must have FOODBATCH records
2. Each batch has manufacture and expiry dates
3. Products cannot be sold past expiry
4. Batches enable product recall tracking

# **RELATIONSHIP SUMMARY**

| **Parent Entity** | **Relationship** | **Child Entity** | **Cardinality** | **Description** |
| --- | --- | --- | --- | --- |
| CUSTOMER | Has | ADDRESS | 1:N | Multiple addresses |
| CUSTOMER | Places | ORDER | 1:N | Customer orders |
| CUSTOMER | Owns | CART | 1:1 | One cart per customer |
| CUSTOMER | Writes | REVIEW | 1:N | Product reviews |
| ORDER | Contains | ORDERITEM | 1:N | Order line items |
| ORDER | Generates | INVOICE | 1:1 | One invoice per order |
| ORDER | Pays | PAYMENT | 1:N | Multiple payments allowed |
| ORDER | Ships | SHIPMENT | 1:0..1 | Optional shipment |
| PRODUCT | For | ORDERITEM | 1:N | Product in orders |
| SUPPLIER | Supplies | PRODUCT | 1:N | **Supplier products (CRITICAL)** |
| PRODUCT | Categorizes | CATEGORY | N:1 | Product category |
| PRODUCT | Tracks | INVENTORY | 1:N | Multiple locations |
| PRODUCT | Batches | FOODBATCH | 1:N | Food batches |
| CART | Has | CARTITEM | 1:N | Items in cart |

# **DATA DICTIONARY SUMMARY**

## **CUSTOMER Table**

| **Column** | **PK/FK** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| CustomerID | PK | INT | AUTO\_INCREMENT | Unique ID |
| FirstName |  | VARCHAR(50) | NOT NULL | First name |
| LastName |  | VARCHAR(50) | NOT NULL | Last name |
| Email |  | VARCHAR(100) | UNIQUE, NOT NULL | Email |
| Phone |  | VARCHAR(20) |  | Phone number |
| DateRegistered |  | DATE | NOT NULL | Registration date |
| Address |  | VARCHAR(200) |  | Address |
| Status |  | VARCHAR(20) | NOT NULL | Account status |

## **ORDER Table**

| **Column** | **PK/FK** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| OrderID | PK | INT | AUTO\_INCREMENT | Unique ID |
| CustomerID | FK | INT | NOT NULL | Customer reference |
| AddressID | FK | INT | NOT NULL | Shipping address |
| OrderDate |  | DATE | NOT NULL | Order date |
| RequiredDate |  | DATE |  | Delivery date |
| Status |  | VARCHAR(20) | NOT NULL | Order status |
| TotalPrice |  | DECIMAL(10,2) | NOT NULL | Total amount |
| PaymentStatus |  | VARCHAR(20) | NOT NULL | Payment status |

## **ORDERITEM Table (Junction)**

| **Column** | **PK/FK** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| OrderItemID | PK | INT | AUTO\_INCREMENT | Unique ID |
| OrderID | FK | INT | NOT NULL | Order reference |
| ProductID | FK | INT | NOT NULL | Product reference |
| Quantity |  | INT | NOT NULL, >0 | Quantity |
| Price |  | DECIMAL(10,2) | NOT NULL | Price at order |

## **PRODUCT Table**

| **Column** | **PK/FK** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| ProductID | PK | INT | AUTO\_INCREMENT | Unique ID |
| CategoryID | FK | INT | NOT NULL | Category reference |
| SupplierID | FK | INT | NOT NULL | **Supplier reference (CRITICAL)** |
| Name |  | VARCHAR(100) | NOT NULL | Product name |
| Description |  | TEXT |  | Description |
| Price |  | DECIMAL(10,2) | NOT NULL | Current price |
| StockQuantity |  | INT | NOT NULL | Stock level |
| ReorderLevel |  | INT | NOT NULL | Reorder threshold |

## **SUPPLIER Table**

| **Column** | **PK/FK** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| SupplierID | PK | INT | AUTO\_INCREMENT | Unique ID |
| Name |  | VARCHAR(100) | NOT NULL | Supplier name |
| ContactPerson |  | VARCHAR(50) |  | Contact person |
| Email |  | VARCHAR(100) |  | Email |
| Phone |  | VARCHAR(20) |  | Phone |
| Address |  | TEXT |  | Address |

## **Additional Entities**

The following entities follow the same 3-column format (PK/FK | Name | Type):

* CART, CARTITEM, INVOICE, PAYMENT, SHIPMENT
* CATEGORY, INVENTORY, FOODBATCH, PRODUCTION\_BATCH
* REVIEW, TRACEABILITY\_REPORT, REPORT\_ITEM, REPORT\_SHIPMENT
* ORDER\_ALLOCATION, ADDRESS, BOOKING\_SYSTEM, ALERT\_LOG

# **REQUIREMENTS VERIFICATION**

## **Logical Data Model Checklist**

All entities have primary keys (PK marked)  
 All relationships have foreign keys (FK marked)  
 All attributes have data types (3-column format)  
 Cardinality shown on all relationships  
 No many-to-many relationships (6 junction tables)  
 No multivalued attributes (atomic values only)  
 No composite attributes (decomposed)  
 Third normal form achieved (3NF verified)  
 Denormalization justified (TotalPrice documented)

## **Instructor Feedback Addressed**

**Supplier-Product link clear and visible**

* Added SupplierID FK in PRODUCT
* Diamond relationship with cardinality
* Documented in 3-column format

**Customer-order process documented**

* Clear flow: CUSTOMER → ORDER → ORDERITEM → PRODUCT
* Junction table eliminates M:N
* Business rules documented

**Converted to logical model**

* All PKs and FKs marked
* All data types specified
* ER notation with cardinality

## **Additional Requirements**

**Data types mandatory** - All attributes have explicit types  
 **3-column format** - PK/FK | Name | Type throughout  
 **Professional presentation** - Clean, consistent  
 **Diamond shapes** - All relationships use diamonds  
 **ER arrows** - Proper cardinality notation  
**This is documented, justified denormalization.**

## 

# **SUMMARY OF CHANGES**

## **What Changed from Initial ERD**

1. Added SupplierID FK to PRODUCT (CRITICAL FIX #1)
2. Created ORDERITEM junction (CRITICAL FIX #2)
3. Created 5 additional junction tables
4. Added all PK/FK markers
5. Specified all data types
6. Achieved 3NF normalization
7. Added ER cardinality notation
8. Added diamond relationship shapes
9. Expanded from ~8 to 23 entities
10. Added 3-column format throughout

## **Statistics**

* **Total Entities:** 23
* **Junction Tables:** 6
* **M:N Eliminated:**
* **3NF Achieved:**
* **All PK/FK Marked:**
* **All Data Types:**

Google Drive: <https://drive.google.com/drive/folders/18pCWMVJXZpMPIlY-O03WkaBKskoBnN2j>

GitHub: <https://github.com/riyanshikedia10/DMDD_Group-1>

