

Enhancing the Database System for E-Commerce

Mahmoud Hussein Shee

C110156202

University of the People

[CS 2203-01 - AY2026-T1](#)

[Stella Chibuike-Ezike \(Instructor\)](#)

Date: Sep 12, 2025

(a) Entities, Attributes, and Relationships

For an e-commerce company, four primary entities are essential to effectively manage customers, products, orders, and inventory. Each entity includes specific attributes that uniquely describe it and support business operations:

- **Customer:** *CustomerID* (PK), *FirstName*, *LastName*, *Email*, *Address*, *Phone*.
Customers serve as the source of all orders, and each order is directly associated with one customer.
- **Product:** *ProductID* (PK), *Name*, *Description*, *Price*, *Category*. Products represent the core of the business as they are sold, tracked, and inventoried.
- **Order:** *OrderID* (PK), *OrderDate*, *TotalAmount*, *Status*, *CustomerID* (FK → Customer). Each order records the products purchased by a specific customer.
- **Inventory:** *InventoryID* (PK), *ProductID* (FK → Product), *QuantityAvailable*, *WarehouseLocation*. Inventory ensures that product stock levels are monitored for availability.

The relationships among these entities are critical to business operations. A **Customer can place multiple Orders (1:N)**. Each **Order may contain many Products**, and conversely, a product can appear in multiple orders, creating an M:N relationship. This is resolved using an associative entity, **OrderDetails**, with attributes: *OrderID* (FK), *ProductID* (FK), Quantity, and Subtotal. Furthermore, a **Product can be stored in multiple Inventory records** across warehouses (1:N).

This structure ensures accurate representation of business rules and supports scalability by preventing data redundancy while maintaining integrity (Coronel & Morris, 2019; Vidhya et al., 2016).

(b) Translation into Tables with Keys

The conceptual model is converted into normalized relational tables with appropriate primary and foreign keys.

Table	Attributes	Primary Key	Foreign Keys
Customer	CustomerID, FirstName, LastName, Email, Address, Phone	CustomerID	—
Product	ProductID, Name, Description, Price, Category	ProductID	—

Order	OrderID, OrderDate, TotalAmount, Status, CustomerID	OrderID	CustomerID → Customer(CustomerID)
Inventory	InventoryID, ProductID, QuantityAvailable, WarehouseLocation	InventoryID	ProductID → Product(ProductID)
OrderDetails	OrderDetailID, OrderID, ProductID, Quantity, Subtotal	OrderDetailID	OrderID → Order(OrderID); ProductID → Product(ProductID)

This tabular design is in **Third Normal Form (3NF)** to eliminate redundancy and maintain referential integrity. For example, customer information is stored only once in the Customer table, while foreign keys in the Order and OrderDetails tables establish necessary links across entities (Elmasri & Navathe, 2016).

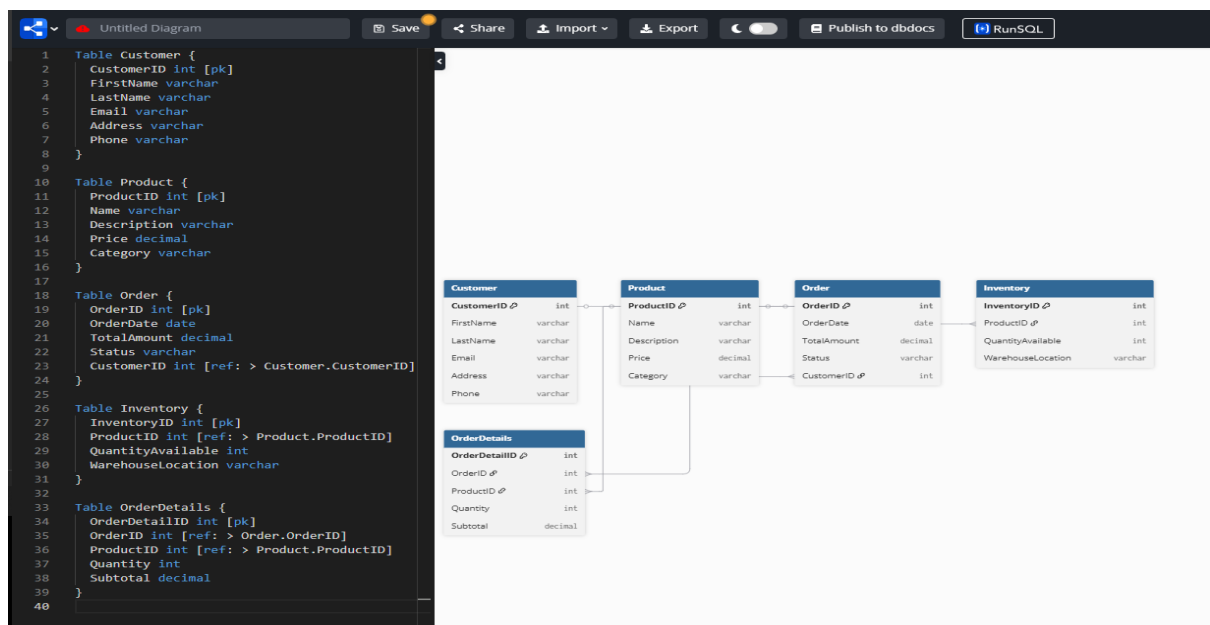
(c) Entity-Relationship (ER) Diagram

The Entity-Relationship (ER) diagram below illustrates the structure of the e-commerce company's database system. It defines the five entities—Customer, Product, Order, Inventory, and OrderDetails—and shows how they are connected through primary and foreign keys.

- Customer is linked to Order in a 1:N relationship, where one customer can place multiple orders.

- Order and Product are connected through the junction table OrderDetails to resolve the M:N relationship, ensuring that each order can contain multiple products and each product can appear in multiple orders.
- Product is related to Inventory in a 1:N relationship, meaning a single product can be stored in multiple inventory records (e.g., across different warehouses).
- Keys are clearly defined, with primary keys underlined (e.g., CustomerID, ProductID, OrderID, InventoryID) and foreign keys marked (e.g., CustomerID in Order, ProductID in Inventory, and both OrderID and ProductID in OrderDetails).
- Cardinality and participation constraints are represented, showing mandatory participation for customers in orders and optional participation where applicable (e.g., inventory entries depend on the presence of products).

Figure 1: ER diagram of the e-commerce database showing entities, attributes, relationships, and constraints.



(d) Conceptual vs. Physical Design

The conceptual design model shows entities, attributes, and relationships from a high-level, business-focused view. For example, identification of Customers, Orders, with the M:N relationship between Orders and Products forms part of the conceptual model. Thus stakeholders can validate that the system aligns with business needs. They do so just because this abstraction prevents them from considering the technical implementation (Nalimov, 2021).

However, the physical design model details implementation for the database on a specific DBMS. It is inclusive of defining data types like “INT”, “VARCHAR”, and “DECIMAL”, and it sets down primary and also foreign key constraints. Optimization of storage is also a part of it, and so are indexing strategies. For faster query performance the physical design introduces the OrderDetails table, specifies foreign key relationships, and applies indexing (Groves, 2022), while the conceptual model states that “Orders contain Products” in this e-commerce case.

These models work to complement each other: the clarity and the accuracy that is required for requirements gathering comes from the conceptual design, while technical efficiency in addition to reliable implementation comes from the physical design (Coronel & Morris, 2019).

Conclusion

A strong database system for e-commerce relies on the accuracy with which we identify all entities and their relationships, the translation of each entity into normalized tables with primary and foreign keys, and the representation of all entities in an ER diagram.

Understanding the difference between conceptual and physical designs is equally important

because the former guarantees alignment with business goals, the latter ensures technical feasibility performance. For supporting scalability and data integrity, the company can optimize order processing, customer information handling, and inventory management.

References

Coronel, C., & Morris, S. (2019). *Database systems: design, implementation, and management* (13th ed.). Cengage Learning.

Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems* (7th ed.). Pearson.

Groves, M. (2022, October 7). What is data modeling? Conceptual, physical, logical.

Couchbase. [The Differences Between Conceptual, Logical, & Physical Data Models](https://couchbase.com/learn/whitepapers/the-differences-between-conceptual-logical-and-physical-data-models)
(couchbase.com)

Nalimov, C. (2021, September 22). What is a conceptual data model? With examples! Gleek.
<https://www.gleek.io/blog/conceptual-data-model.html>

Vidhya, V., Jeyaram, G., & Ishwarya, K. (2016). *Database management systems*. Alpha Science International.