

Assignment1:

Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form

Ans:Business Scenario

Let's consider a simplified business scenario of an e-commerce platform. The platform involves customers placing orders for products. Each order can contain multiple products, and each product can be part of multiple orders. The company also wants to keep track of shipments for each order.

Based on this scenario, we can identify the following entities and their attributes:

Customer:

CustomerID (Primary Key)

Name

Email

Address

Phone

Product:

ProductID (Primary Key)

ProductName

Description

Price

StockQuantity

Order:

OrderID (Primary Key)

OrderDate

CustomerID (Foreign Key from Customer)

OrderDetail (to handle the many-to-many relationship between Order and Product):

OrderDetailID (Primary Key)

OrderID (Foreign Key from Order)

ProductID (Foreign Key from Product)

Quantity

UnitPrice

Shipment:

ShipmentID (Primary Key)

ShipmentDate

OrderID (Foreign Key from Order)

ShipmentStatus

Relationships and Cardinality

Customer-Order: One customer can place many orders. Each order is placed by one customer (1-to-Many).

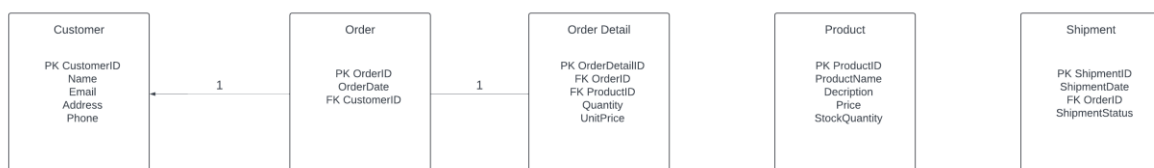
Order-OrderDetail: One order can have multiple order details, and each order detail belongs to one order (1-to-Many).

Product-OrderDetail: One product can appear in multiple order details, and each order detail contains one product (Many-to-1).

Order-Shipment: One order can have multiple shipments, but each shipment is related to one order (1-to-Many).

ER Diagram

Here's how the ER diagram looks:



Normalization

To ensure the schema is normalized up to the third normal form (3NF):

First Normal Form (1NF):

Ensure that the table columns contain atomic values.

Ensure that each column contains values of a single type.

Second Normal Form (2NF):

Ensure that the schema is in 1NF.

Remove partial dependencies; i.e., non-key attributes must depend on the whole primary key.

Third Normal Form (3NF):

Ensure that the schema is in 2NF.

Remove transitive dependencies; i.e., non-key attributes must depend only on the primary key.

Analysis

The Customer table has atomic columns and a unique CustomerID.

The Product table has atomic columns and a unique ProductID.

The Order table has atomic columns and a unique OrderID which refers to CustomerID.

The OrderDetail table handles the many-to-many relationship between Order and Product using OrderID and ProductID. Each OrderDetail also has atomic columns and a unique OrderDetailID.

The Shipment table has atomic columns and a unique ShipmentID which refers to OrderID.

This design follows proper normalization rules and is normalized up to 3NF.