

# DBMS PROJECT REPORT FILE

# ECOMMERCE MANAGEMENT SYSTEM

**Submitted By-**

Shalu Rawat(22/10/JC009) , Ayush Sharma(22/10/JC/042),

Saurabh Singh Chauhan(22/10/JC/056), Harshita Yadav(22/10/JC/048)

# Submitted To –

**Prof. T.V. Vijay Kumar**

## Problem Statement:

The problem at hand is the need for a well-designed and efficiently managed database system for an e- commerce platform. The increasing popularity and growth of the e-commerce industry have led to a significant rise in the volume and complexity of data generated by online transactions, product catalogs, user profiles, and other related activities. Without a robust and scalable database system, the platform faces challenges in effectively organizing, storing, and retrieving data, which can result in data inconsistencies, inefficiencies in order management, compromised user experience, and hindered business growth. Thus, there is a critical need for an optimized database solution that can handle the demands of an e-commerce platform, ensuring seamless transactions, data integrity, and efficient management of information, ultimately driving the success and competitiveness of the business.

## Introduction:

The goal of the project is to design and implement a retail based relational database that showcases how various entities are related to retail and commerce. It also tries to cover all the relative scenarios that one can come across in the field of e-commerce such as creating invoice, placing order, method for payment etc. In order to implement, MySQL is used as the relationaldatabase management system (RDBMS).

The database system will involve the mapping of entities, relationships, and attributes from an Entity-Relationship (ER) model to a relational model. This mapping process will translate the conceptual representation of the data into concrete tables, columns, and relationships in the relational database. Proper normalization techniques will be applied to eliminate redundancy, ensure data consistency, and optimize query performance.

## Requirements gathering & Design Approach:

In order to understand how an e-commerce giant like amazon works, detailed research was required. The first phase involved identifying the entities in an e-commerce business and howdo they interact with each other. So, after a brief understanding, below were some main processes that made up e-commerce:

* A seller selling a product
* A buyer placing order
* A buyer can have Shopping cart and Wishlist
* A buyer can provide reviews for a product
* A product can have offers
* A purchased product is a part of an order
* An order is placed after successful payment
* An invoice is generated for an order
* A product’s quantity is updated after each successfully placed order

## List of entities and their attributes –

1. User (User Id, Usertype,Date Created)
   1. Buyer (BuyerID, BuyerFirstName, BuyerLastName, Membership, Phone, Email)
   2. Seller (SellerID, SellerFirstName, SellerLastName, Company Name, Logo, Phone, Email)
2. Shopping Cart (Shopping Card ID, Quantity)
3. Wishlist (Wishlist ID)
4. Reviews (Review ID, customer review, Ratings)
5. Address (Address ID, Address \_Type, Address line1, Country, State, City, Postal Code)
6. Product (Product ID, Product Name, Category, Item Picture, Product Description, Unit Price, Quantity, Units in Stock, Units in Order)
7. Offer (Offer ID, Offer Amount)
8. Order (Order ID, Order Date, Price, Item Quantity, Transaction Status, Payment Date, Required Date, Tax)
9. Invoice (Invoice ID, Invoice type, Invoice amount)
10. Shipper (Shipper ID, Shipper Company Name, Contact name, Phone)
11. Payment (Payment ID, Payment Type)
    1. Payment Card (Card ID, Card Number, CardExpMonth, CardExpYear)
    2. Payment\_GiftCard ( GiftCardID, GiftCardNumber, GiftCardAmount, GiftCardExpMonth, GiftCardExpYear )

## Mapping Cardinalities:

To understand the design better, ER diagram provides the conceptual schematic representationof the database system. But before making the ER diagram, one also needs to understand the relationships and cardinality between various entities:

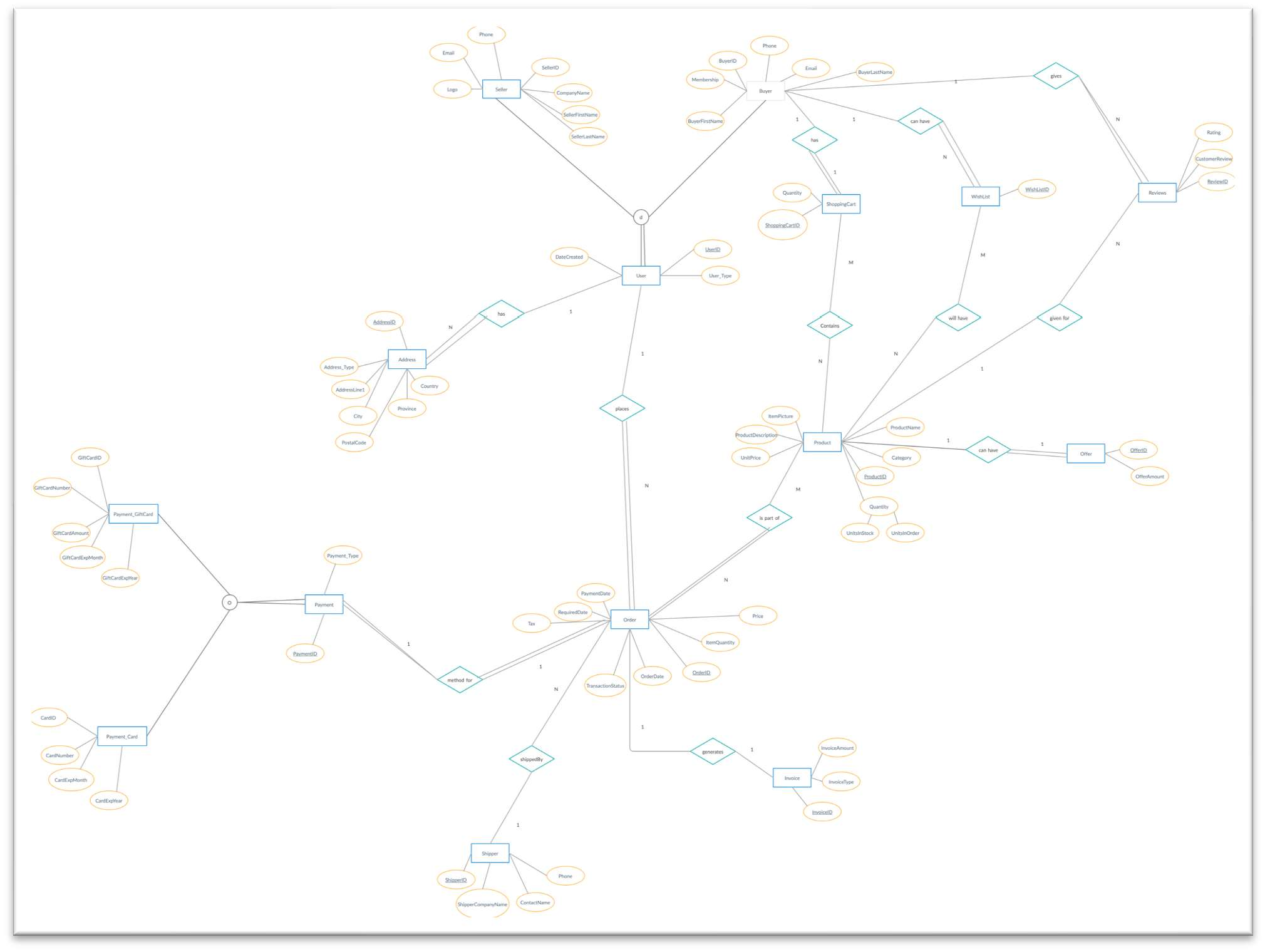
* + - **User-Address (1-N):** A user may have multiple address and one address must belong toone user.
    - **Buyer-Shopping Cart (1-1):** A buyer may have one shopping cart and one shopping cartmust belong to one buyer.
    - **Buyer-Wish List (1-N):** A buyer may have multiple Wish lists and one Wish list mustbelong to one buyer.
    - **Buyer-Reviews (1-N):** A buyer may give multiple reviews for different products and areview must belong to one buyer.
    - **Shopping Cart-Product (M-N):** A shopping cart may contain multiple products and aproduct may be in different shopping carts.
    - **Wish List-Product (M-N):** A wish list may contain multiple products and a product maybe in different wish lists.
    - **Reviews-Product (N-1):** A review may belong to one product and one product may havemultiple reviews.
    - **Product-Offers (1-1):** A product may have one offer and an offer must belong to oneproduct.
    - **User-Order (1-N):** A user may place multiple orders and an order must belong

to oneuser.

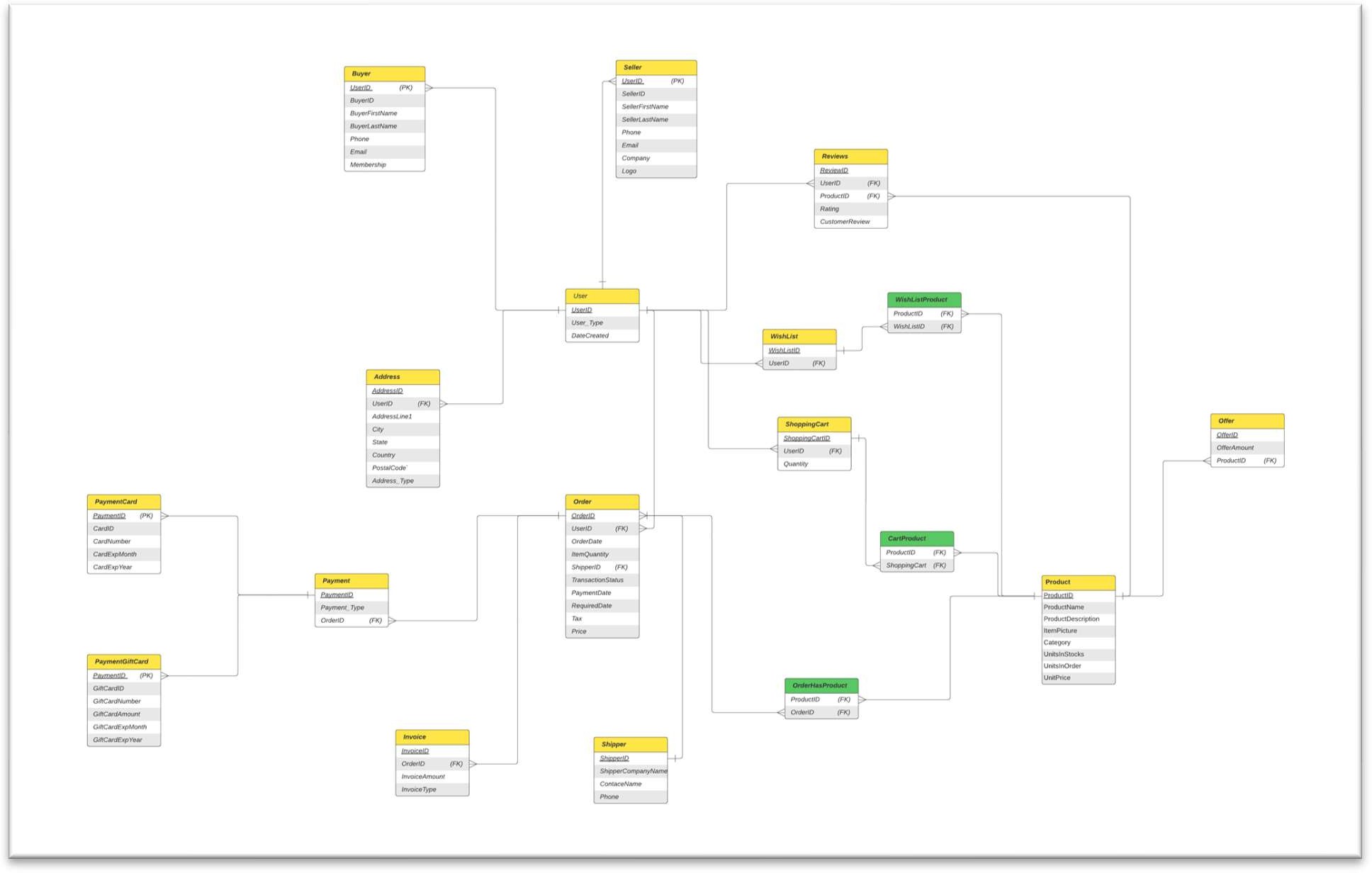
* + - **Product-Order (M-N):** A product may be a part of multiple orders and an order must contain one or multiple products.
    - **Order-Payment (1-1):** An order should have a payment method and a payment method must belong to an order.
    - **Order-Shipper (N-1):** An order may have one shipper and a shipper may have multipleorders to ship.
    - **Order-Invoice (1-1):** An order may have one invoice and an invoice may belong to oneorder.

**Also,** a user has a ***disjoint*** relationship with buyer and seller as a user must only be a buyer or aseller. In addition to above, a payment has an ***overlapping*** relationship with payment gift card and payment card as any of the two or both must be used for payment.

# ER DAIGRAM:



**Relational Model:**



**Normalization:**

**Relation 1:** User (User Id, Usertype, , Date Created)

**Step 1:** : Identify Functional Dependencies:

User Id → Usertype, Date Created

**Step 2:** Create First Normal Form (1NF):

In the given schema, the "User" table is already in 1NF since it contains only atomic values, and there are no repeating groups.

**Step 3:** Create Second Normal Form (2NF):

In 2NF, we need to eliminate partial dependencies by ensuring that each non-key attribute is fully dependent on the entire primary key. Since the "User" table has a single candidate key (User Id), there are no partial dependencies. Therefore, the table already satisfies 2NF.

**Step 4**: Create Third Normal Form (3NF):

In 3NF, we need to remove transitive dependencies. To identify such dependencies, we can examine the functional dependencies and determine if any non-key attribute depends on another non-key attribute.

In the "User" table, there are no non-key attributes that depend on other non-key attributes. Therefore, the table already satisfies 3NF.

**Step 5:** Create Boyce-Codd Normal Form (BCNF):

To achieve BCNF, we need to ensure that every determinant (left-hand side of a functional dependency) is a super key. In the "User" table, the determinant is User Id, which is already the primary key and thus a super key. Hence, the table satisfies BCNF.

### Normalized User Table (BCNF):

User (User Id, Usertype,Date Created)

The resulting table is already in BCNF, meaning it is free from all partial dependencies, transitive dependencies, and other anomalies.

**Relation 2:** Buyer (BuyerID, BuyerFirstName, BuyerLastName, Membership, Phone, Email)

**Step 1:** : Identify Functional Dependencies:

BuyerID → BuyerFirstName, BuyerLastName, Membership, Phone, Email

**Step 2:** Create First Normal Form (1NF):

No atomic values, hence “Buyer” table is already in 1NF.

**Step 3:** Create Second Normal Form (2NF):

No partial dependency, hence “Buyer” table is already in 2NF.

**Step 4**: Create Third Normal Form (3NF):

No transitive dependency**,** hence “Buyer” table is already in 3NF.

**Step 5:** Create Boyce-Codd Normal Form (BCNF):

BuyerID (left-hand side of a functional dependency) is a super key. Hence it satisfies BCNF.

### Normalized User Table (BCNF):

Buyer (BuyerID, BuyerFirstName, BuyerLastName, Membership, Phone, Email)

**Relation 3:** Seller (SellerID, SellerFirstName, SellerLastName, Company Name, Logo, Phone, Email)

**Step 1**: Identify Functional Dependencies:

SellerID → SellerFirstName, SellerLastName, Company Name, Logo, Phone, Email

**Step 2:** Create First Normal Form (1NF):

The attributes in the "Seller" table are already atomic, so it is already in 1NF.

**Step 3:** Create Second Normal Form (2NF):

There are no partial dependencies in the "Seller" table since all non-key attributes depend on the entire primary key. Hence, it is already in 2NF.

**Step 4:** Create Third Normal Form (3NF):

There are no transitive dependencies in the "Seller" table since all non-key attributes are directly dependent on the primary key. Therefore, it is already in 3NF.

**Step 5:** Create Boyce-Codd Normal Form (BCNF):

SellerID (left-hand side of the functional dependency) is the super key. Hence, it satisfies BCNF.

### Normalized Seller Table (BCNF):

Seller (SellerID, SellerFirstName, SellerLastName, Company Name, Logo, Phone, Email)

Repeat the same process for the remaining tables:

**Relation 4:** Shopping Cart (Shopping Cart ID, Quantity)

* Already in BCNF: Shopping Cart (Shopping Cart ID, Quantity)

**Relation 5:** Wishlist (Wishlist ID)

* Already in BCNF: Wishlist (Wishlist ID)

**Relation 6:** Reviews (Review ID, Customer Review, Ratings)

* Already in BCNF: Reviews (Review ID, Customer Review, Ratings)

**Relation 7:** Address (Address ID, Address Type, Address Line1, Country, State, City, Postal Code)

* Already in BCNF: Address (Address ID, Address Type, Address Line1, Country, State, City, Postal Code)

**Relation 8:** Product (Product ID, Product Name, Category, Item Picture, Product Description, Unit Price, Quantity, Units in Stock, Units in Order)

* Already in BCNF: Product (Product ID, Product Name, Category, Item Picture, Product Description, Unit Price, Quantity, Units in Stock, Units in Order)

**Relation 9:** Oﬀer (Oﬀer ID, Oﬀer Amount)

* Already in BCNF: Oﬀer (Oﬀer ID, Oﬀer Amount)

**Relation 10:** Order (Order ID, Order Date, Price, Item Quantity, Transaction Status, Payment Date, Required Date, Tax)

* Already in BCNF: Order (Order ID, Order Date, Price, Item Quantity, Transaction Status, Payment Date, Required Date, Tax)

**Relation 11:** Invoice (Invoice ID, Invoice Type, Invoice Amount)

* Already in BCNF: Invoice (Invoice ID, Invoice Type, Invoice Amount)

**Relation 12:** Shipper (Shipper ID, Shipper Company Name, Contact Name, Phone)

* Already in BCNF: Shipper (Shipper ID, Shipper Company Name, Contact Name, Phone)

**Relation 13:** Payment (Payment ID, Payment Type)

* Already in BCNF: Payment (Payment ID, Payment Type)

**Relation 14:** Payment Card (Card ID, Card Number, Card Exp Month, Card Exp Year)

* Already in BCNF: Payment Card (Card ID, Card Number, Card Exp Month, Card Exp Year)

**Relation 15:** Payment GiftCard (GiftCard ID, GiftCard Number, GiftCard Amount, GiftCard Exp Month, GiftCard Exp Year)

* Already in BCNF: Payment GiftCard (GiftCard ID, GiftCard Number, GiftCard Amount, GiftCard Exp Month, GiftCard Exp Year)

All the remaining tables were already in BCNF, so no further normalization is required.

**SQL Script:**

Create database ecommerce; Use ecommerce;



Table 1. User CREATE TABLE User1(

UserID VARCHAR(8) NOT NULL, USER\_TYPE VARCHAR(1) NOT NULL,

DateCreated DATE NOT NULL, PRIMARY KEY (UserID));

INSERT INTO User1 (UserID, USER\_TYPE, DateCreated) VALUES ('U1234567', 'A', '2023-01-15'),

('U2345678', 'B', '2022-11-30'),

('U3456789', 'C', '2023-02-28');

Table 2. Buyer

CREATE TABLE Buyer(

BuyerID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL,

Membership VARCHAR(1) NOT NULL, BuyerFirstName VARCHAR(50) NOT NULL, BuyerLastName VARCHAR(50) NULL,

PhoneNumber VARCHAR(15) DEFAULT 'xxx-xxx-xxxx' NOT NULL, Email VARCHAR(50) NOT NULL,

PRIMARY KEY (UserID)

);

INSERT INTO Buyer (BuyerID, UserID, Membership, BuyerFirstName, BuyerLastName,

PhoneNumber, Email) VALUES

('B0000001', 'U1234567', 'G', 'John', 'Doe', '123-456-7890', 'john.doe@example.com'),

('B0000002', 'U2345678', 'S', 'Jane', 'Smith', '555-123-4567', 'jane.smith@example.com'),

('B0000003',

'U3456789',

'G',

'Michael',

'Johnson',

'999-888-7777',

'michael.johnson@example.com');



Table 3. Seller CREATE TABLE Seller (

SellerID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL,

CompanyName VARCHAR(50) NOT NULL, SellerFirstName VARCHAR(50) NOT NULL, SellerLastName VARCHAR(50) NOT NULL,



PhoneNumber VARCHAR(15) DEFAULT 'xxx-xxx-xxxx' NOT NULL,

Email VARCHAR(50) NOT NULL,

INSERT INTO Seller (SellerID, UserID,

PhoneNumber, Email, Logo) VALUES

CompanyName, SellerFirstName,

SellerLastName,

('S0000001',

'U1234567',

'ABC

Company', 'John',

'Doe',

'123-456-7890',

'john.doe@example.com', NULL),

('S0000002',

'U2345678',

'XYZ

Inc.',

'Jane',

'Smith',

'555-123-4567',

'jane.smith@example.com', NULL),

('S0000003', 'U3456789', '123 Corporation', 'Michael', 'Johnson', '999-888-7777',

'michael.johnson@example.com', NULL);

Table 4. Address

CREATE TABLE Address (

AddressID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL,

Address\_Type VARCHAR(20) NOT NULL, AddressLine1 VARCHAR(50) NULL,

City VARCHAR(50) NULL, Province VARCHAR(50) NULL, Country VARCHAR(50) NULL,

PostalCode VARCHAR(10) NULL,

PRIMARY KEY (AddressID));

INSERT INTO Address (AddressID, UserID, Address\_Type, AddressLine1, City, Province,

Country, PostalCode) VALUES

('A0000001', 'U1234567', 'Billing', '123 Main Street', 'New York', 'New York', 'United States', '10001'),

('A0000002', 'U2345678', 'Shipping', '456 Elm Avenue', 'Los Angeles', 'California', 'United States', '90001'),



('A0000003', 'U3456789', 'Billing', '789 Oak Drive', 'Chicago', 'Illinois', 'United States', '60601');



Table 5. Product CREATE TABLE Product (

ProductID VARCHAR(8) NOT NULL,

Department VARCHAR(50) NOT NULL, ProductName VARCHAR(50) NOT NULL, UnitPrice DECIMAL NOT NULL,

ProductDescription VARCHAR(255) NULL, UnitsInStock INT NULL,

UnitsInOrder INT NULL, ItemPicture BLOB NULL, PRIMARY KEY (ProductID));

INSERT INTO Product (ProductID, Department, ProductName, UnitPrice, ProductDescription,

UnitsInStock, UnitsInOrder, ItemPicture) VALUES

('P0000001', 'Electronics', 'Smartphone', 599.99, 'A high-end smartphone with advanced features.', 100, 10, NULL),

('P0000002', 'Clothing', 'T-Shirt', 19.99, 'A comfortable and stylish t-shirt for everyday wear.', 50, 5, NULL),

('P0000003', 'Home Appliances', 'Coﬀee Maker', 39.99, 'A programmable coﬀee maker for brewing your favorite coﬀee.', 30, 8, NULL);

Table 6. Shopping Cart

CREATE TABLE Shopping\_Cart ( ShoppingCartID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL,

Quantity VARCHAR(8),

PRIMARY KEY (ShoppingCartID));



INSERT INTO Shopping\_Cart (ShoppingCartID, UserID, Quantity) VALUES ('SC0000001', 'U1234567', '2'),

('SC0000002', 'U2345678', '3'),

('SC0000003', 'U3456789', '1');



Table 7. Wishlist CREATE TABLE Wishlist (

WishlistID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL, PRIMARY KEY (WishlistID));

INSERT INTO Wishlist (WishlistID, UserID) VALUES ('W0000001', 'U1234567'),

('W0000002', 'U2345678'),

('W0000003', 'U3456789');

Table 8. Reviews

CREATE TABLE Reviews (

ReviewID VARCHAR(8) NOT NULL, ProductID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL,

CustomerReview VARCHAR(100) NULL, Rating VARCHAR(1) NOT NULL,

PRIMARY KEY (ReviewID));

INSERT INTO Reviews (ReviewID, ProductID, UserID, CustomerReview, Rating) VALUES

('R0000001', 'P0000001', 'U1234567', 'Great smartphone!', '5'),

('R0000002', 'P0000002', 'U2345678', 'Nice t-shirt.', '4'),



('R0000003', 'P0000003', 'U3456789', 'Good coﬀee maker.', '4');



Table 9. Oﬀer CREATE TABLE Oﬀer (

OﬀerID VARCHAR(8) NOT NULL, ProductID VARCHAR(8) NULL,

OﬀerAmount VARCHAR(8) NULL, PRIMARY KEY (OﬀerID));

INSERT INTO Oﬀer (OﬀerID, ProductID, OﬀerAmount) VALUES ('O0000001', 'P0000001', '500'),

('O0000002', 'P0000002', '15'),

('O0000003', 'P0000003', '35');

Table 10. Orders CREATE TABLE Orders (

OrderID VARCHAR(8) NOT NULL, UserID VARCHAR(8) NOT NULL, ShipperID VARCHAR(8) NOT NULL,

OrderDate DATE NOT NULL, RequiredDate DATE NOT NULL, Tax DECIMAL NOT NULL,

TransactionStatus VARCHAR(50) NOT NULL, PaymentDate DATE NOT NULL, ItemQuantity INT NULL,

Price DECIMAL NOT NULL, PRIMARY KEY (OrderID));



INSERT INTO Orders (OrderID, UserID, ShipperID, OrderDate, RequiredDate, Tax,

TransactionStatus, PaymentDate, ItemQuantity, Price) VALUES

('ORD00001', 'U1234567', 'SHP00001', '2023-06-01', '2023-06-15', 10.0, 'Pending', '2023-06-

05', 5, 99.99),

('ORD00002', 'U2345678', 'SHP00002', '2023-06-02', '2023-06-16', 7.5, 'Completed', '2023-06-

06', 3, 49.99),

('ORD00003', 'U3456789', 'SHP00001', '2023-06-03', '2023-06-17', 5.0, 'Shipped', '2023-06-

07', 2, 29.99);



Table 11. Order\_Product CREATE TABLE Order\_Product ( OrderID VARCHAR(8) NOT NULL,

ProductID VARCHAR(8) NOT NULL,

PRIMARY KEY (OrderID,ProductID));

INSERT INTO Order\_Product (OrderID, ProductID) VALUES ('ORD00001', 'P0000001'),

('ORD00001', 'P0000002'),

('ORD00002', 'P0000003');



Table 12. Cart\_Product CREATE TABLE Cart\_Product (

ShoppingCartID VARCHAR(8) NOT NULL, ProductID VARCHAR(8) NOT NULL,

PRIMARY KEY (ShoppingCartID,ProductID));

INSERT INTO Cart\_Product (ShoppingCartID, ProductID) VALUES ('SC0000001', 'P0000001'),

('SC0000001', 'P0000002'),

('SC0000002', 'P0000003');



Table 13. WishList\_Product CREATE TABLE WishList\_Product ( WishlistID VARCHAR(8) NOT NULL, ProductID VARCHAR(8) NOT NULL,

PRIMARY KEY (WishlistID,ProductID));

INSERT INTO WishList\_Product (WishlistID, ProductID) VALUES ('W0000001', 'P0000001'),

('W0000001', 'P0000002'),

('W0000002', 'P0000003');



Table 14 . Invoice CREATE TABLE Invoice (

OrderID VARCHAR(8) NOT NULL, InvoiceID VARCHAR(8) NOT NULL,

Invoice\_Type VARCHAR(4) NOT NULL, InvoiceAmount DECIMAL NOT NULL, PRIMARY KEY (InvoiceID));

INSERT INTO Invoice (OrderID, InvoiceID, Invoice\_Type, InvoiceAmount) VALUES ('ORD00001', 'INV00001', 'Type1', 100.50),

('ORD00002', 'INV00002', 'Type2', 75.25),

('ORD00003', 'INV00003', 'Type1', 50.75);

Table 15. Shipper CREATE TABLE Shipper (

ShipperID VARCHAR(8) NOT NULL,



ShipperCompanyName VARCHAR(50) NULL,

ContactName VARCHAR(50) NULL,

Phone VARCHAR(15) NULL, PRIMARY KEY (ShipperID));

INSERT INTO Shipper (ShipperID, ShipperCompanyName, ContactName, Phone) VALUES

('SHP00001', 'Company A', 'John Doe', '123-456-7890'),

('SHP00002', 'Company B', 'Jane Smith', '987-654-3210'),

('SHP00003', 'Company C', 'Mike Johnson', '555-555-5555');



Table 16. Payment CREATE TABLE Payment (

PaymentID VARCHAR(8) NOT NULL, OrderID VARCHAR(8) NOT NULL,

Payment\_Type VARCHAR(1) NOT NULL, PRIMARY KEY (PaymentID));

INSERT INTO Payment (PaymentID, OrderID, Payment\_Type) VALUES ('PAY00001', 'ORD00001', 'A'),

('PAY00002', 'ORD00002', 'B'),

('PAY00003', 'ORD00003', 'C');

Table 17. Payment\_Card CREATE TABLE Payment\_Card ( CardID VARCHAR(8) NOT NULL,

PaymentID VARCHAR(8) NOT NULL, CardNumber VARCHAR(20) NOT NULL,

CardExpMonth INT NOT NULL, CardExpYear INT NOT NULL,



PRIMARY KEY (PaymentID));

INSERT INTO Payment\_Card (CardID, PaymentID, CardNumber, CardExpMonth, CardExpYear)

VALUES

('CARD1', 'PAY00001', '1234567890123456', 12, 2025),

('CARD2', 'PAY00002', '9876543210987654', 6, 2024),

('CARD3', 'PAY00003', '5678901234567890', 9, 2023);



Table 18. Payment\_GiftCard CREATE TABLE Payment\_Giftcard ( GiftcardID VARCHAR(8) NOT NULL, PaymentID VARCHAR(8) NOT NULL,

GiftCardAmount VARCHAR(8) NOT NULL, GiftCardNumber VARCHAR(16) NOT NULL, GiftcardExpMonth VARCHAR(2) NOT NULL, GiftcardExpYear VARCHAR(4) NOT NULL, PRIMARY KEY (PaymentID));

INSERT INTO Payment\_Giftcard (GiftcardID, PaymentID, GiftCardAmount, GiftCardNumber,

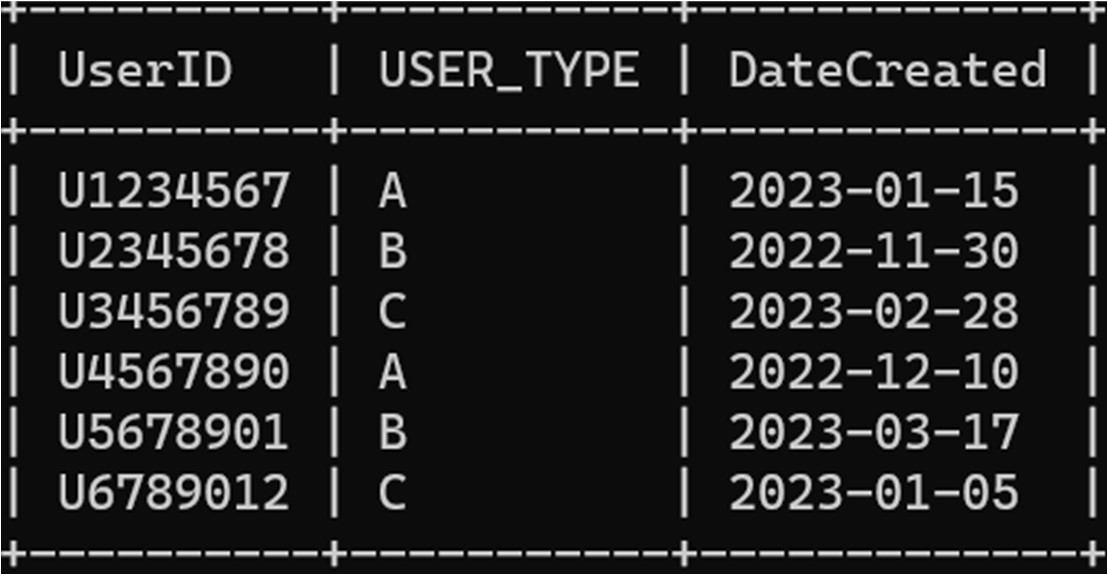
GiftcardExpMonth, GiftcardExpYear) VALUES

('GIFT1', 'PAY00001', '100.00', '1234567890123456', '12', '2025'),

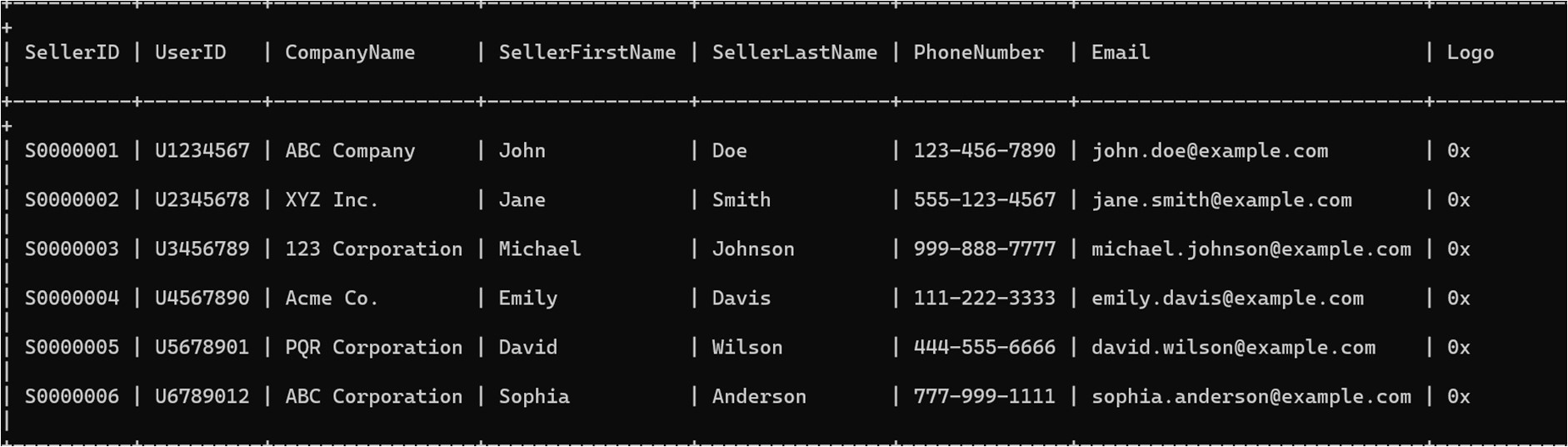
('GIFT2', 'PAY00002', '50.00', '9876543210987654', '06', '2024'),

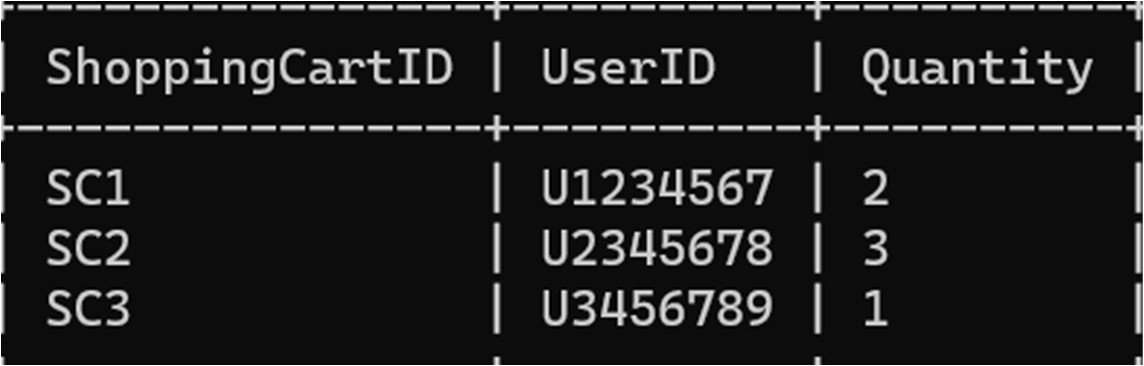
('GIFT3', 'PAY00003', '75.00', '5678901234567890', '09', '2023');

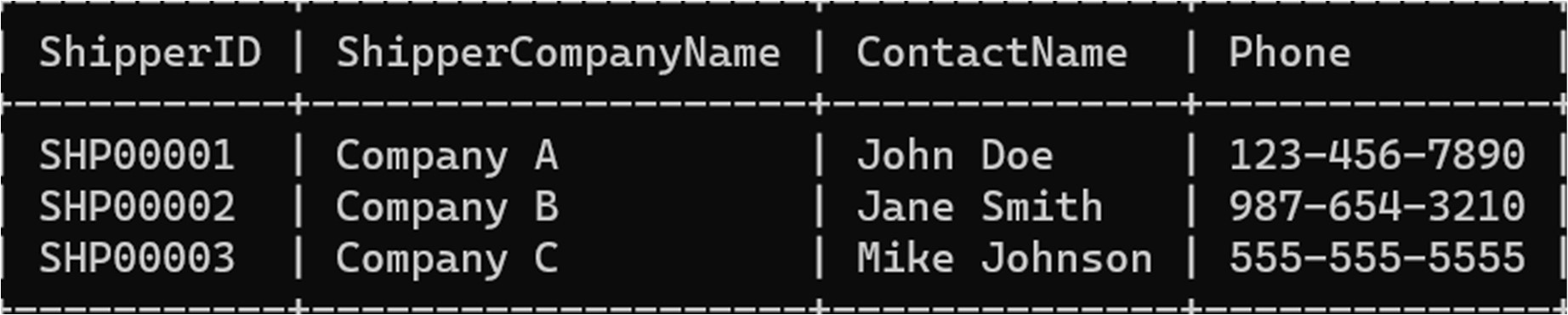
### Database Tables:

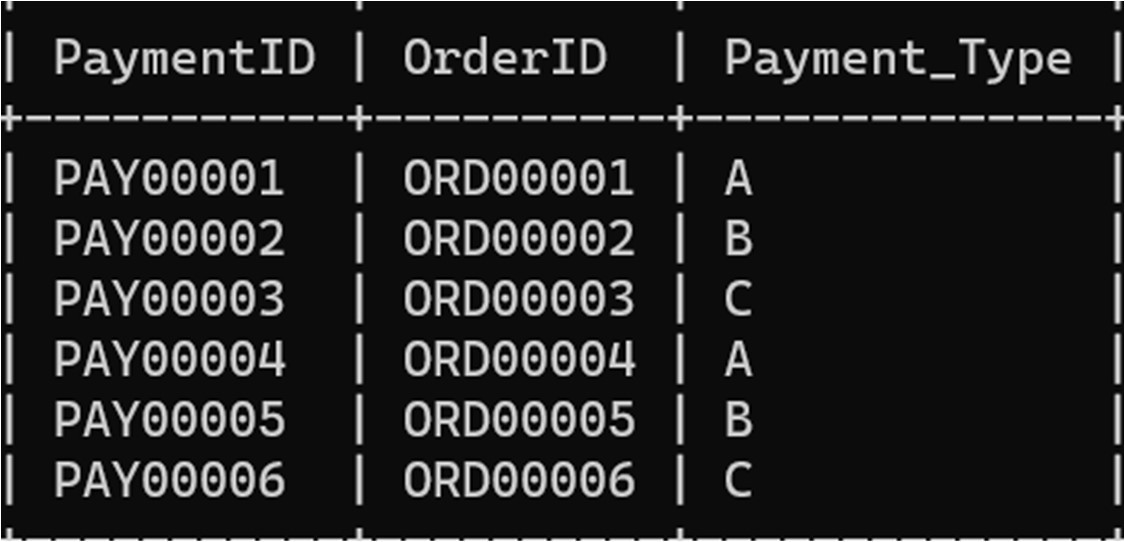


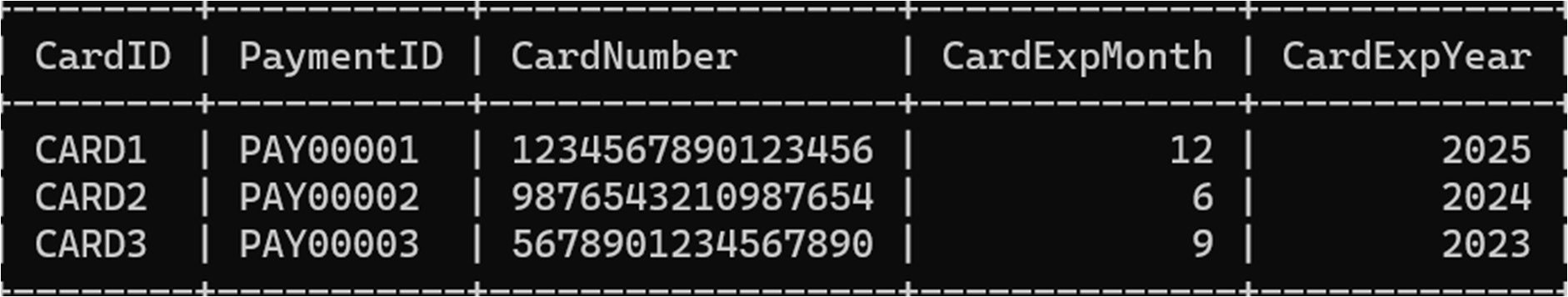


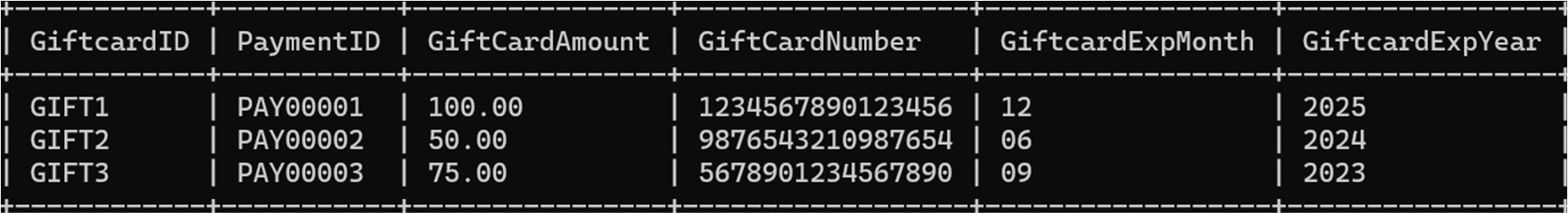


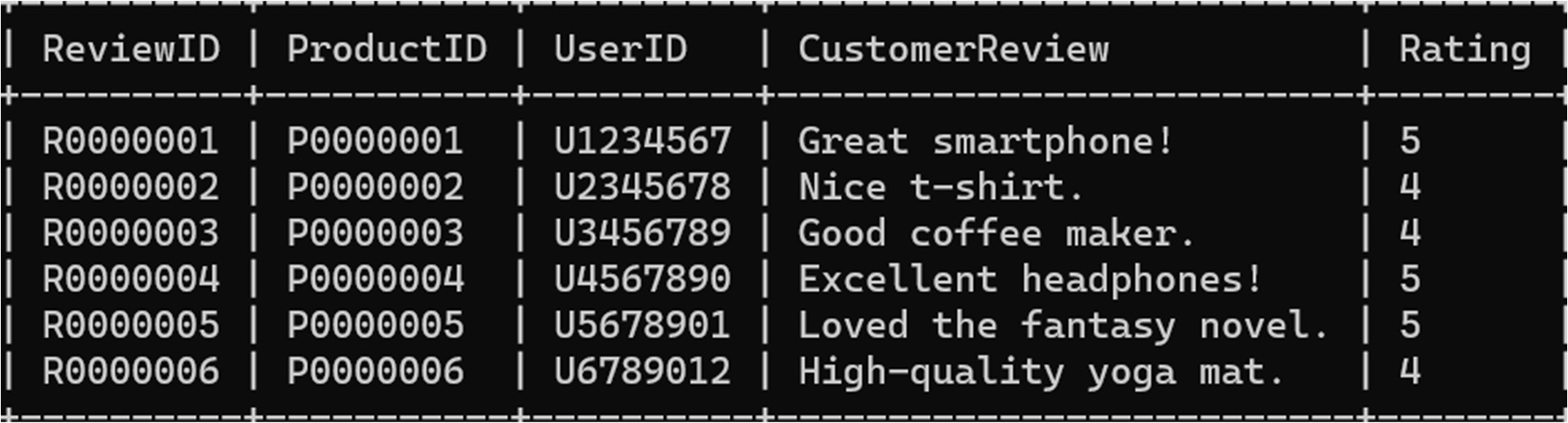


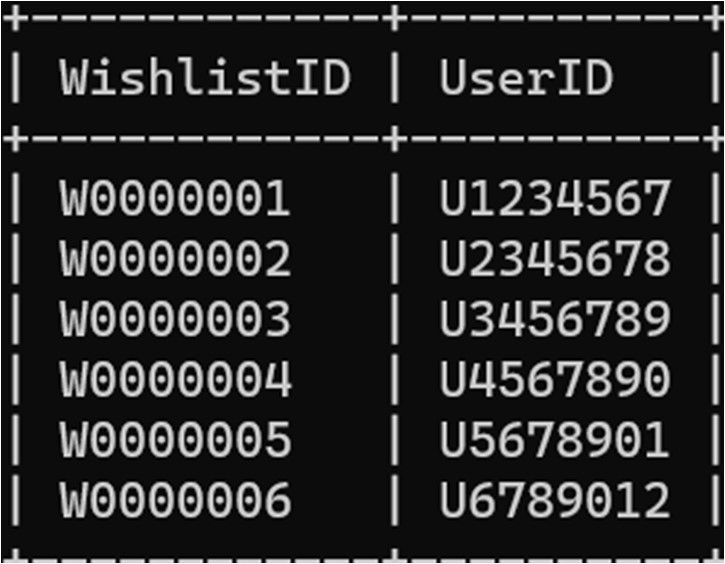


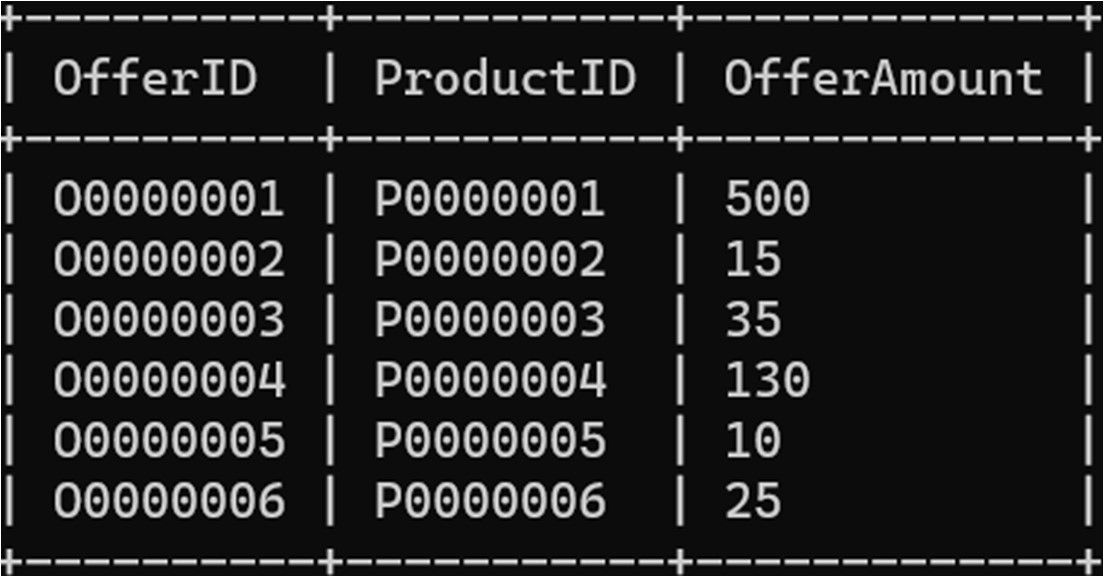


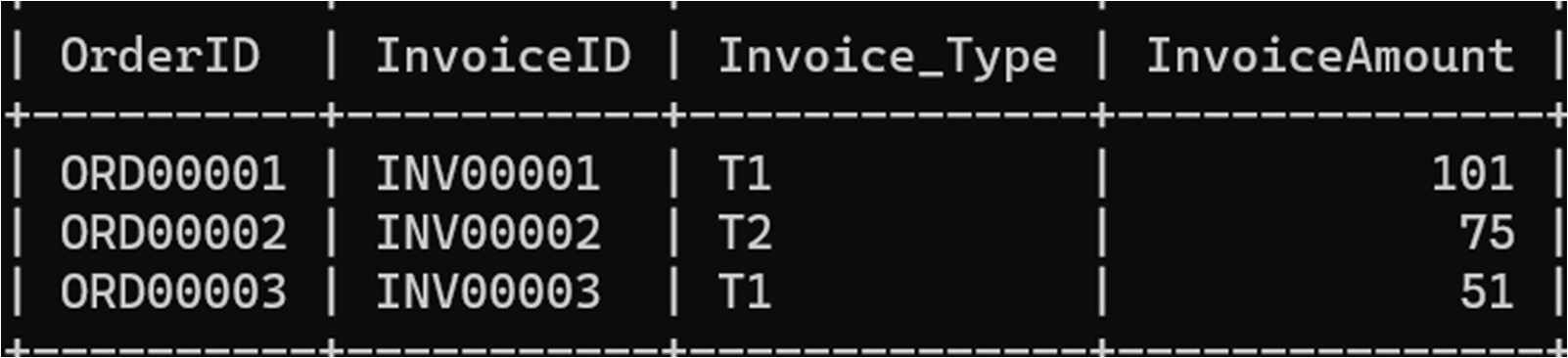


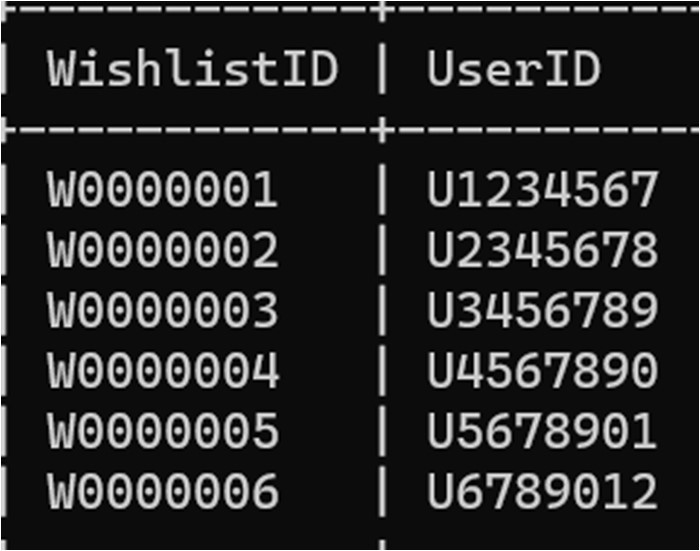


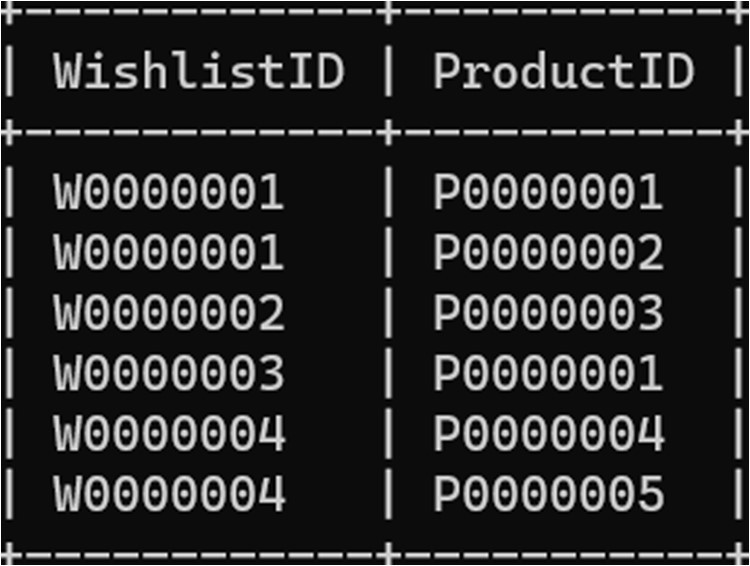












**SYSTEM CONFIGURATION:**

### Software used:

* 1. MySQL -VERSION 8.0.30
  2. Notepad

### System Specifications:

* 1. OS version -64 bit Operating System
  2. Processor – intel core i5
  3. RAM 8GB.

### Conclusion:

The above project presents a relational schema for ecommerce system. It involves creating a database schema with tables for users, buyers, sellers, addresses, products, shopping carts, wishlists, reviews, oﬀers, orders, invoices, shippers, payments, and more. The tables store relevant information related to these entities to support an eCommerce platform.