

# **BÀI TẬP LỚN**

# **CƠ SỞ DỮ LIỆU**

## **NHÓM 15**

**CƠ SỞ DỮ LIỆU QUẢN LÝ SÀN THƯƠNG MẠI ĐIỆN TỬ**

# **NHÓM 15**

---

**HOÀNG SƠN (NT)**  
**PHẠM MINH TÚ**  
**NGUYỄN CÔNG TRÌNH**  
**NGUYỄN CÔNG VINH**

1 MÔ TẢ HỆ THỐNG

2 ER MODEL

3 MH QUAN HỆ VÀ PHỤ  
THUỘC HÀM

4 CONSTRAINTS

5 TRIGGERS

6 PROCEDURES

7 TRANSACTION

8 QUERIES

9 TỔNG KẾT

# 1. MÔ TẢ HỆ THỐNG

## CÁC THÀNH PHẦN:

1. users: Quản lý thông tin người dùng
2. shops: Lưu thông tin cửa hàng và chủ sở hữu
3. products: Quản lý sản phẩm
4. productLines: Nhóm sản phẩm theo danh mục
5. payments: Theo dõi thanh toán
6. paymentMethods: Lưu thông tin phương thức thanh toán
7. vouchers: Quản lý mã giảm giá

## **CÁC THÀNH PHẦN:**

- 8. voucherStorage: Quản lý mã giảm giá của người dùng**
- 9. orders: Quản lý đơn hàng**
- 10. orderDetails: Chi tiết sản phẩm trong đơn hàng**
- 11. itemsInCart: Quản lý giỏ hàng người dùng**
- 12. userPenalty: Ghi nhận vi phạm của người dùng**
- 13. shopPenalty: Ghi nhận vi phạm của cửa hàng**
- 14. shippingUnits: Quản lý thông tin đơn vị vận chuyển**

## **QUY TRÌNH HOẠT ĐỘNG:**

### **1. Quản lý sản phẩm**

- **Mỗi sản phẩm có mã duy nhất.**
- **Cửa hàng cung cấp đầy đủ thông tin sản phẩm vào hệ thống**

### **2. Người mua chọn sản phẩm**

- **Người mua thêm sản phẩm vào giỏ hàng hoặc mua trực tiếp mà không cần giỏ hàng**

## **QUY TRÌNH HOẠT ĐỘNG:**

### **3. Quản lý giỏ hàng:**

- Giỏ hàng lưu thông tin sản phẩm được chọn
- Khi đặt mua, sản phẩm từ giỏ hàng sẽ chuyển vào danh sách đặt hàng và bị xóa khỏi giỏ hàng

### **4. Tạo và xác nhận đơn hàng:**

- Khi người mua xác nhận, hệ thống tạo đơn hàng và danh sách chi tiết sản phẩm, số lượng, trạng thái

## **QUY TRÌNH HOẠT ĐỘNG:**

### **5. Sử dụng voucher:**

- **Người mua áp dụng mã giảm giá nếu đáp ứng điều kiện.**
- **Voucher có thể áp dụng cho:**
  - **Một cửa hàng hoặc loại sản phẩm cụ thể.**
  - **Mọi sản phẩm từ tất cả cửa hàng.**

### **6. Thanh toán:**

- **Người mua lựa chọn thanh toán ngay khi đặt hàng hoặc khi nhận hàng**



## **QUY TRÌNH HOẠT ĐỘNG:**

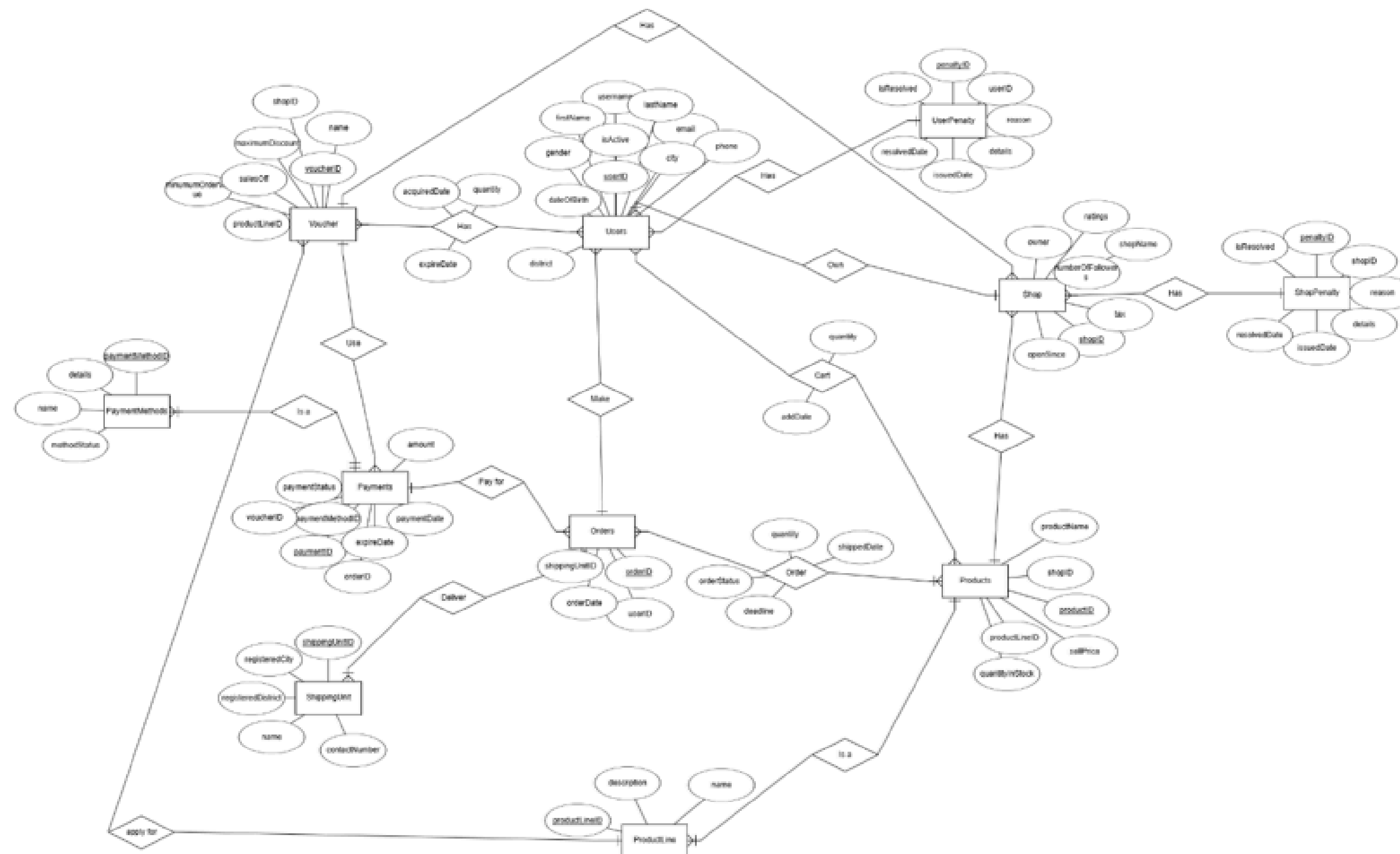
### **7. Vận chuyển:**

- Sau khi đơn hàng được xác nhận, hệ thống bàn giao sản phẩm cho đơn vị vận chuyển
- Đơn vị vận chuyển chọn shipper phù hợp dựa trên khu vực khách hàng

### **8. Xử lý vi phạm:**

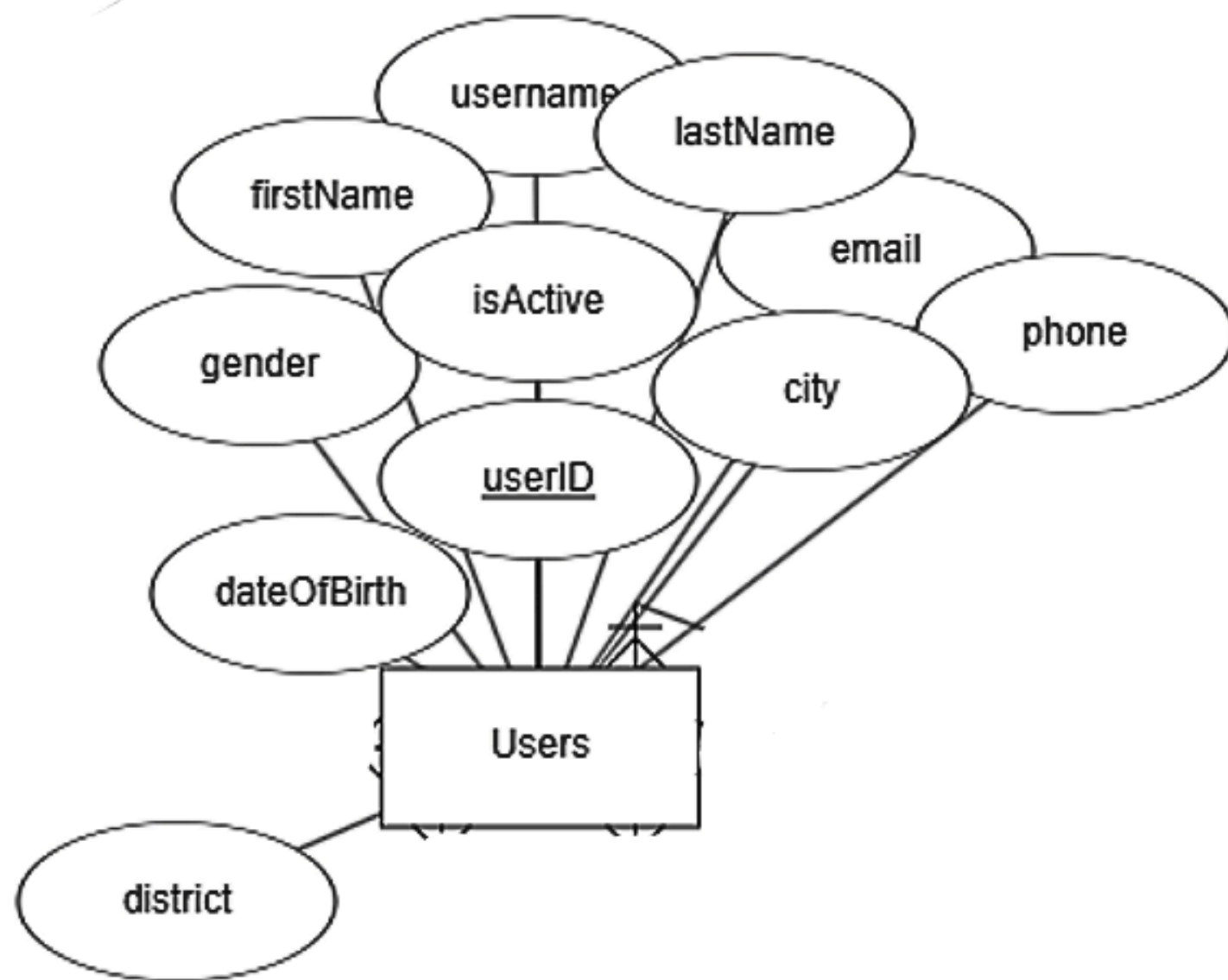
- Hệ thống ghi nhận và quản lý các vi phạm của người dùng hoặc cửa hàng nếu xảy ra

# 2. ER MODEL

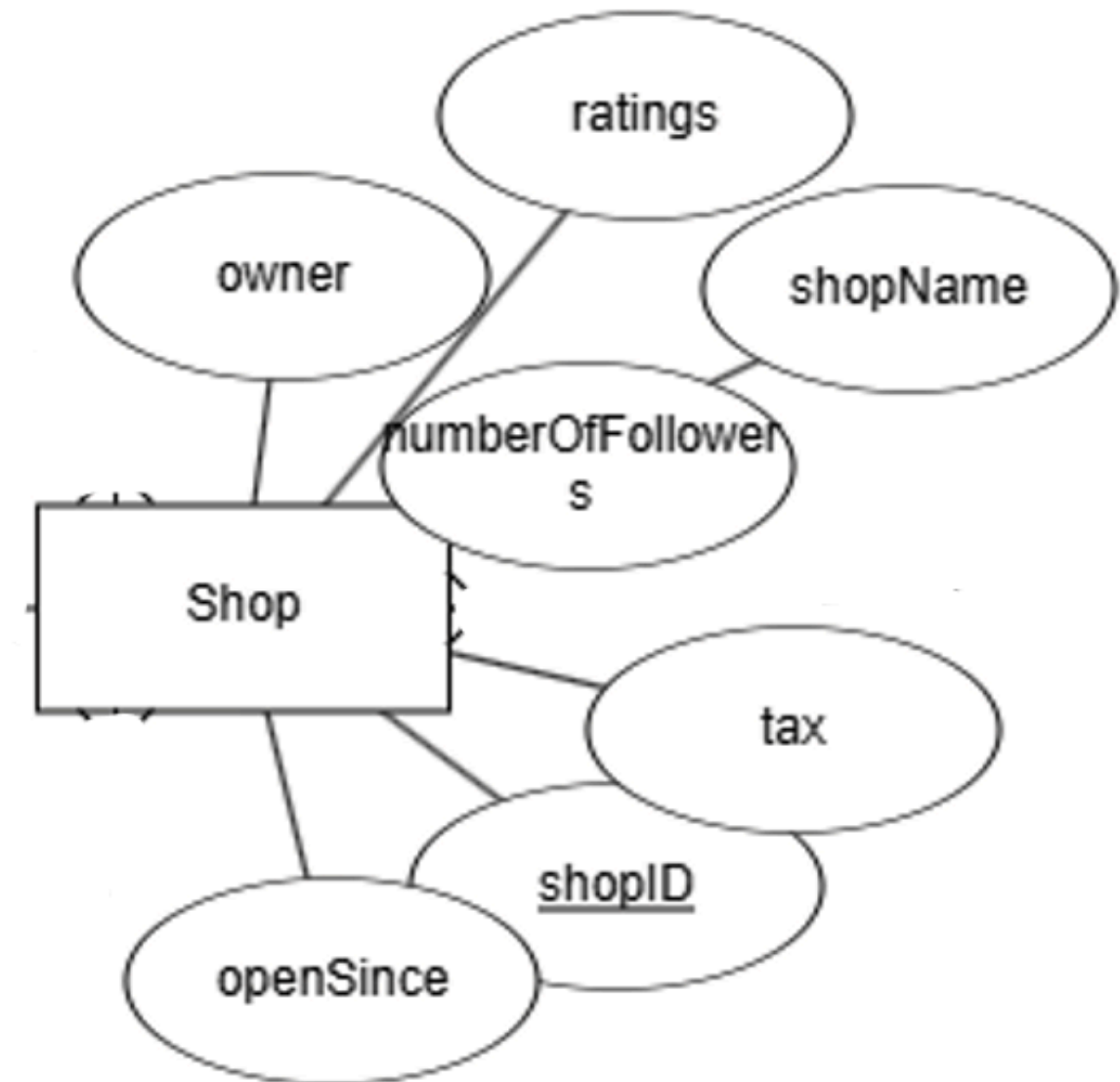


# CÁC THỰC THỂ:

## 1. Users



## 2. Shop

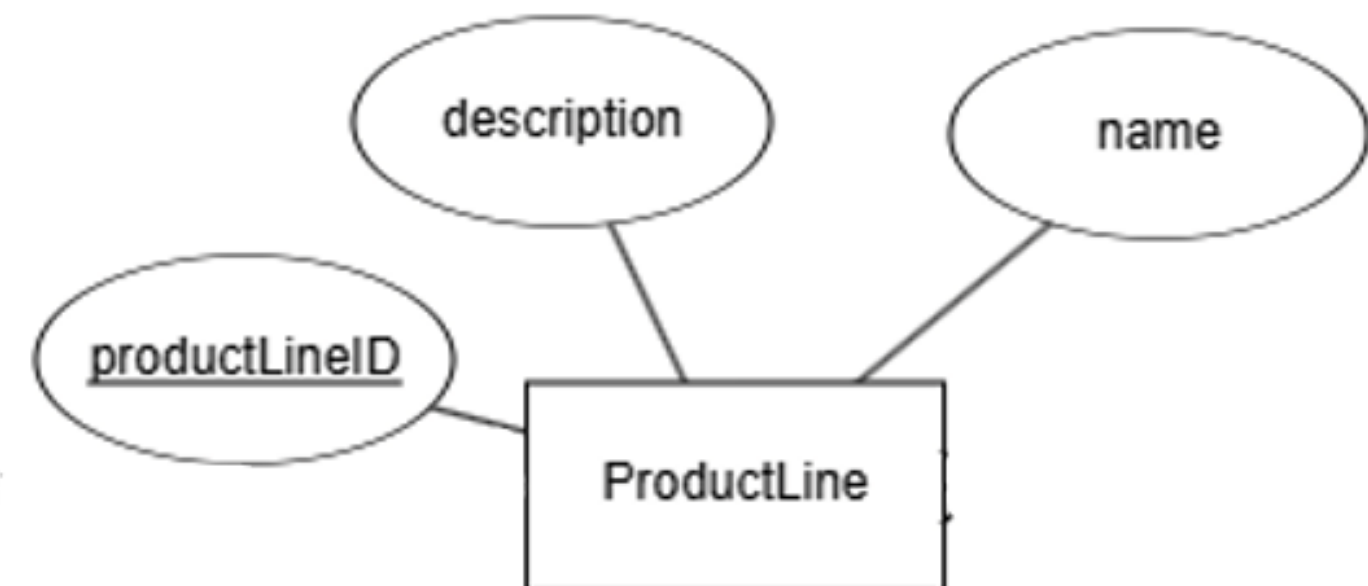


# CÁC THỰC THỂ:

## 3. Products

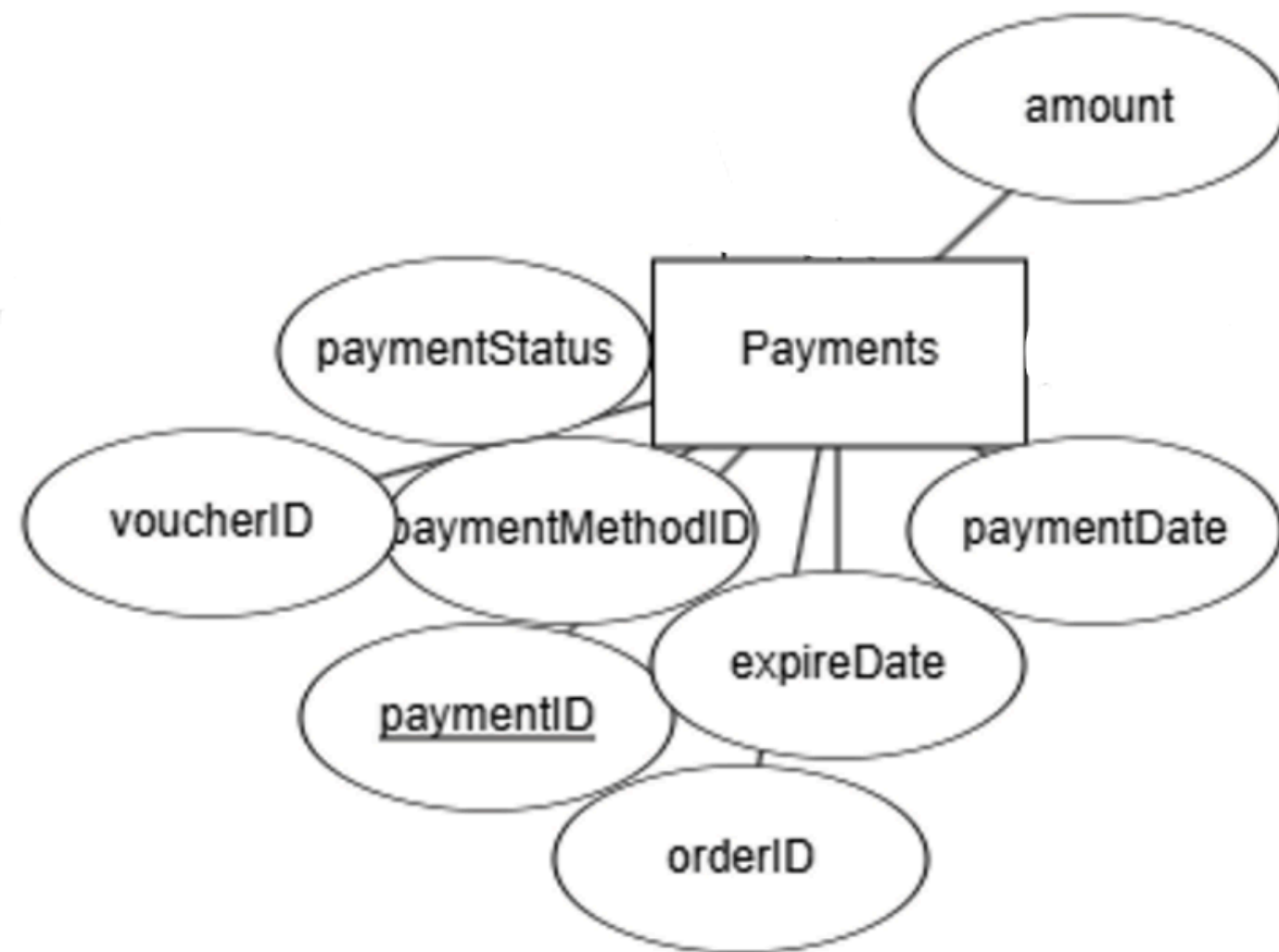


## 4. ProductLine

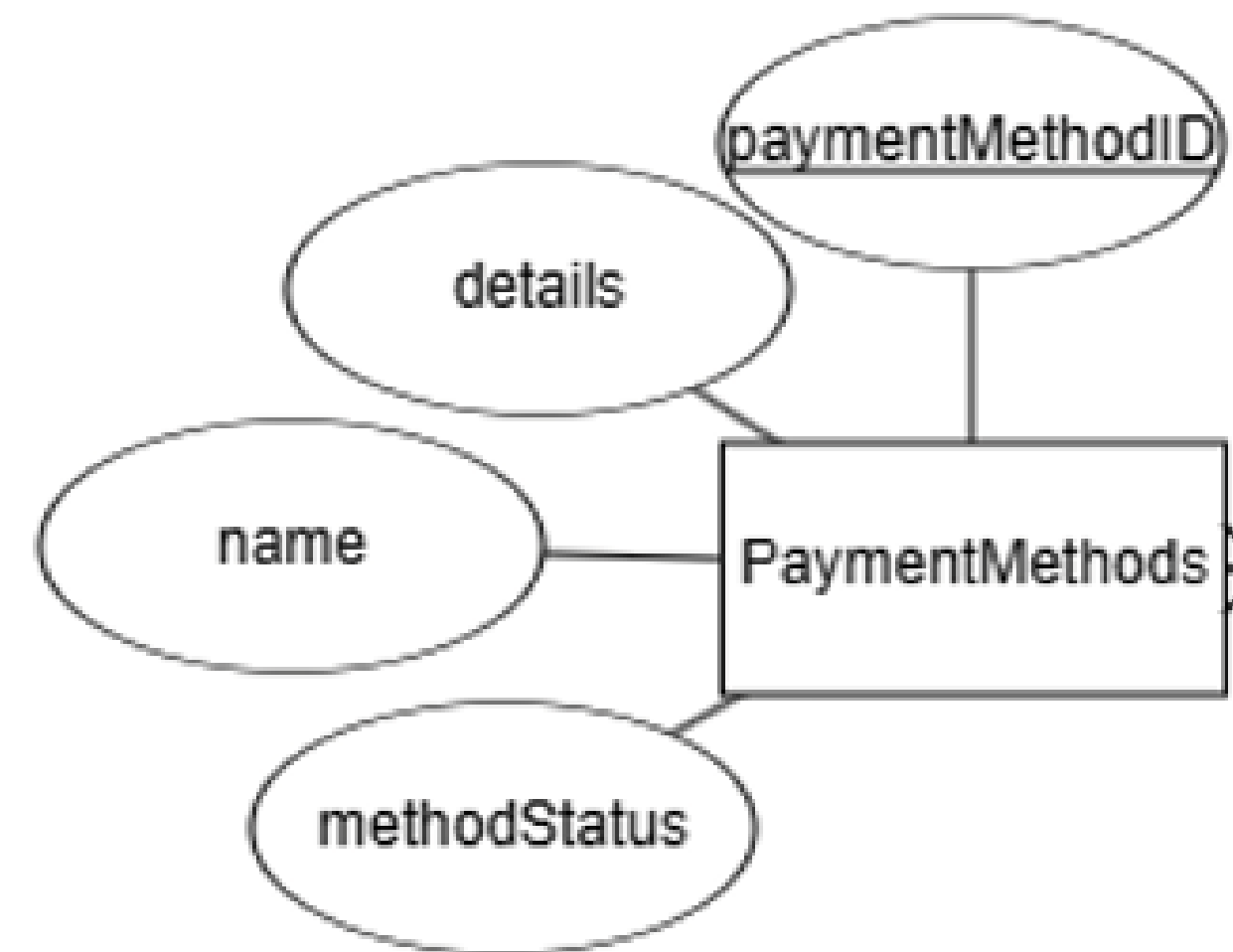


# CÁC THỰC THỂ:

## 5. Payments

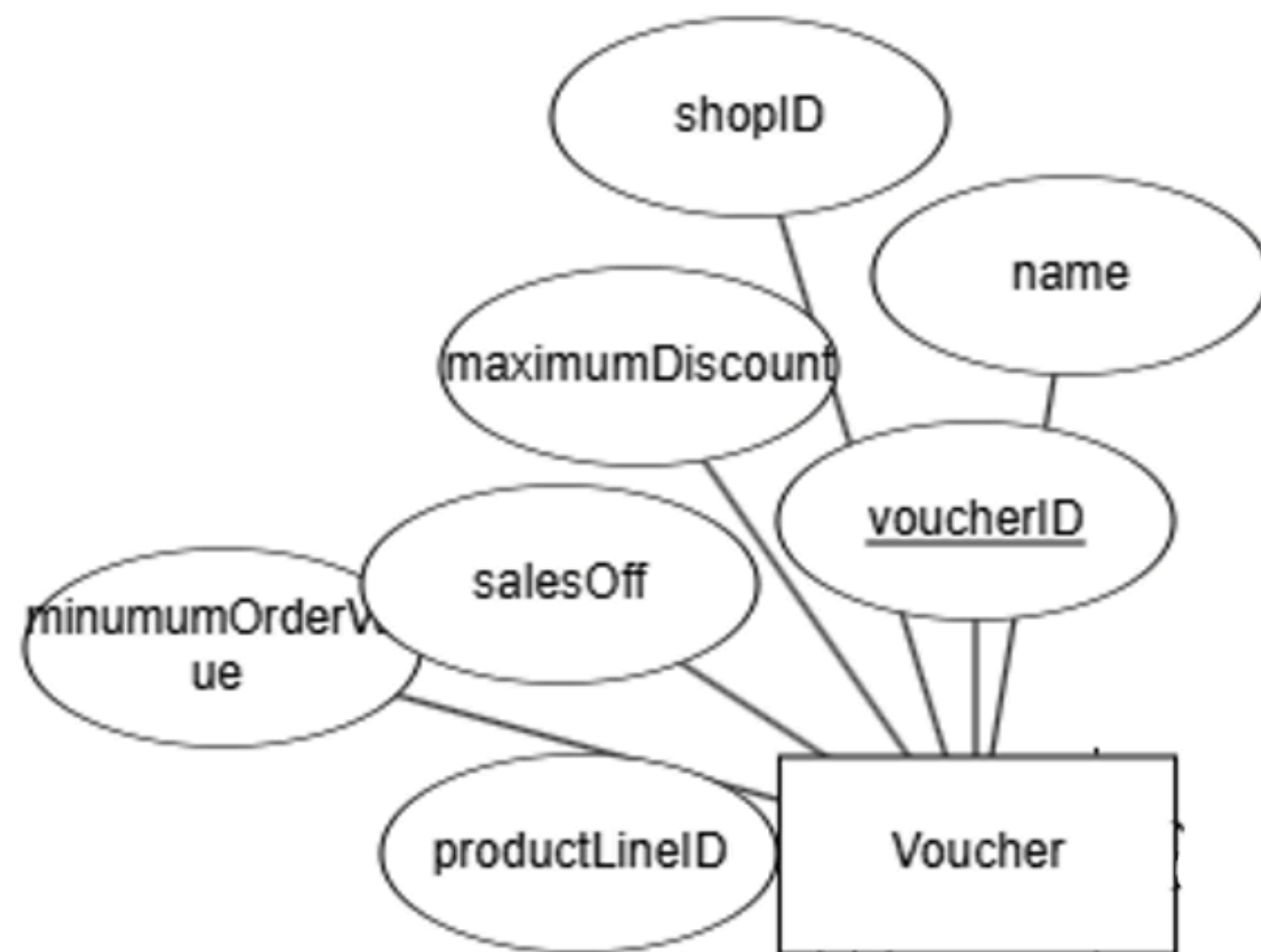


## 6. PaymentMethods

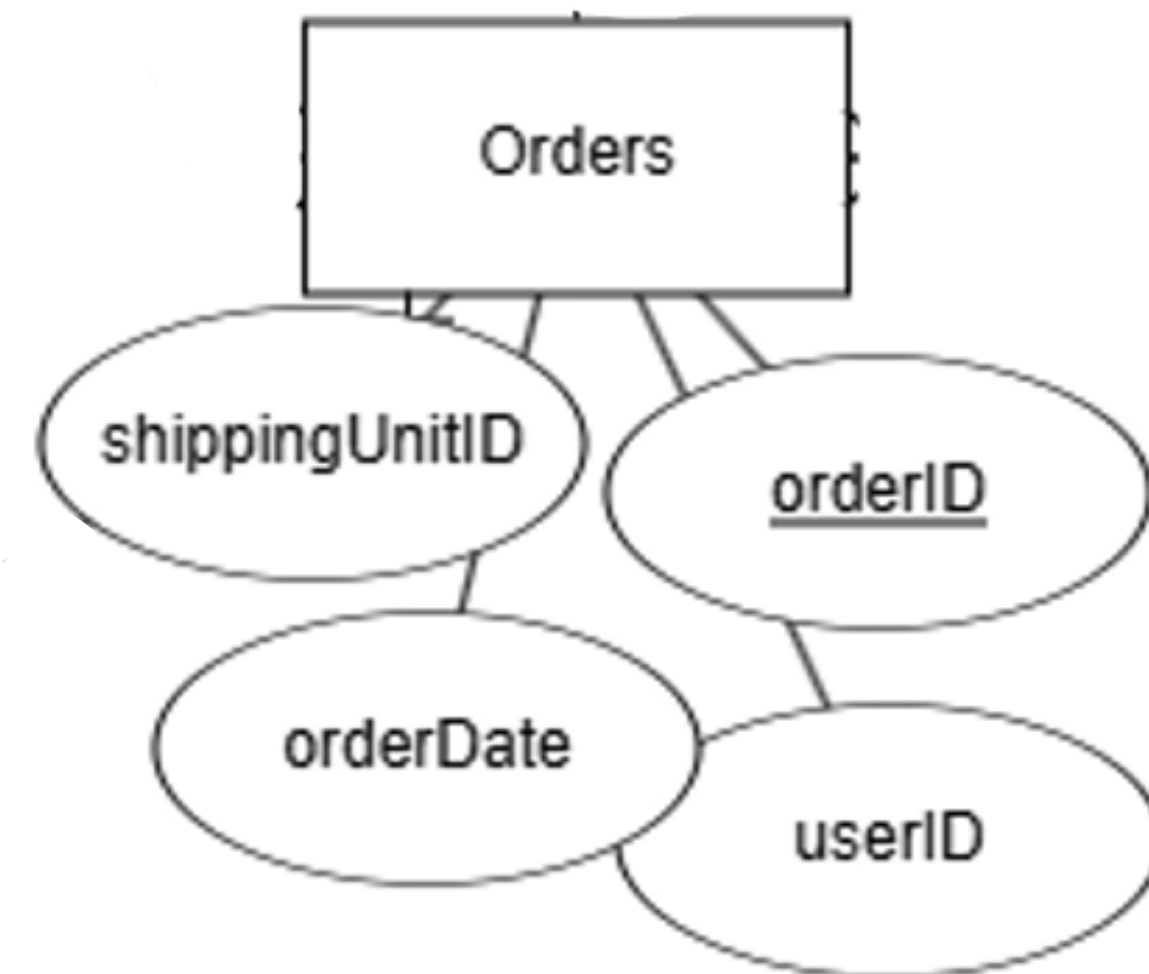


# CÁC THỰC THỂ:

## 7. Voucher

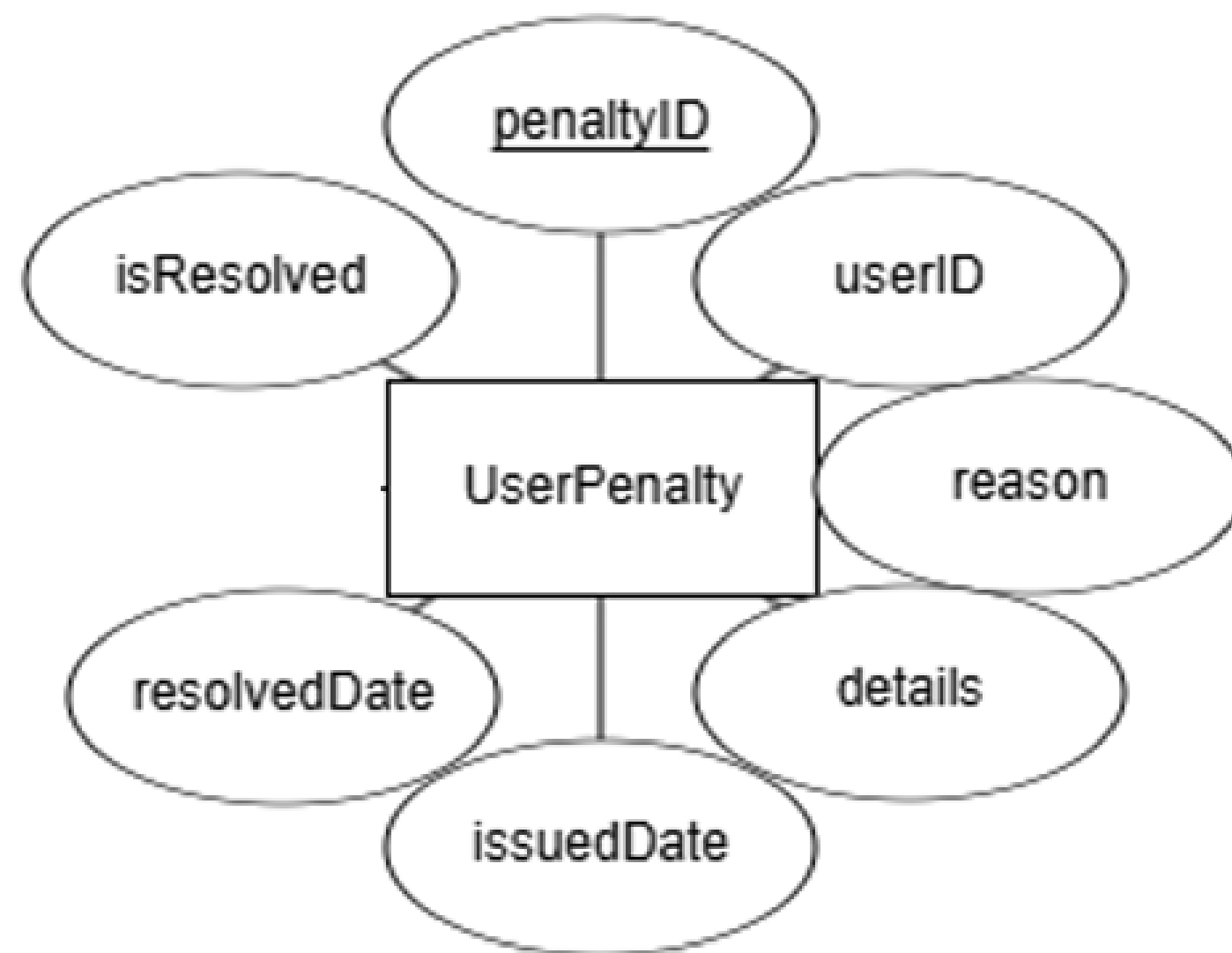


## 8. Order

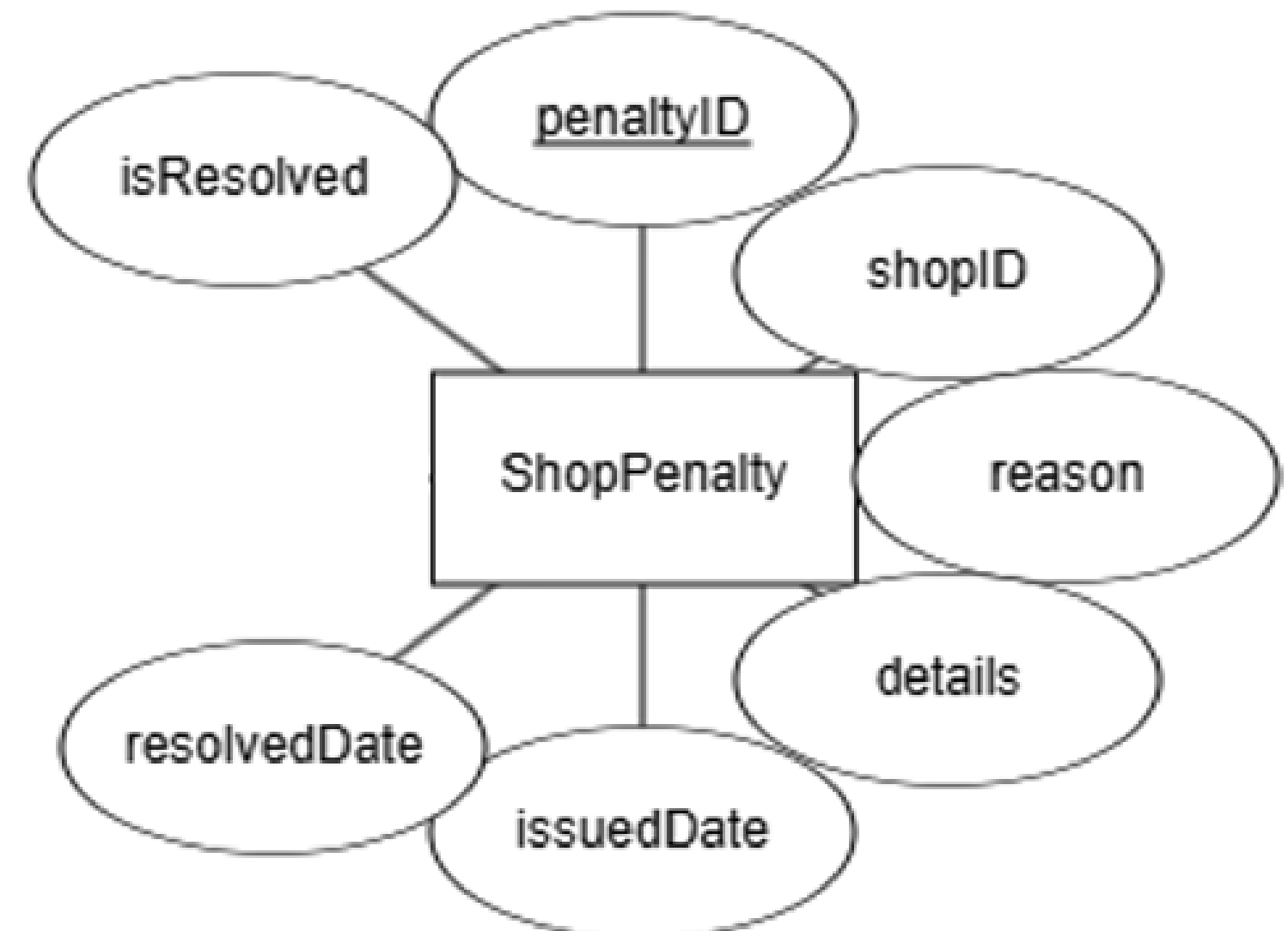


# CÁC THỰC THỂ:

## 9. UserPenalty



## 10. ShopPenalty



# CÁC THỰC THỂ:

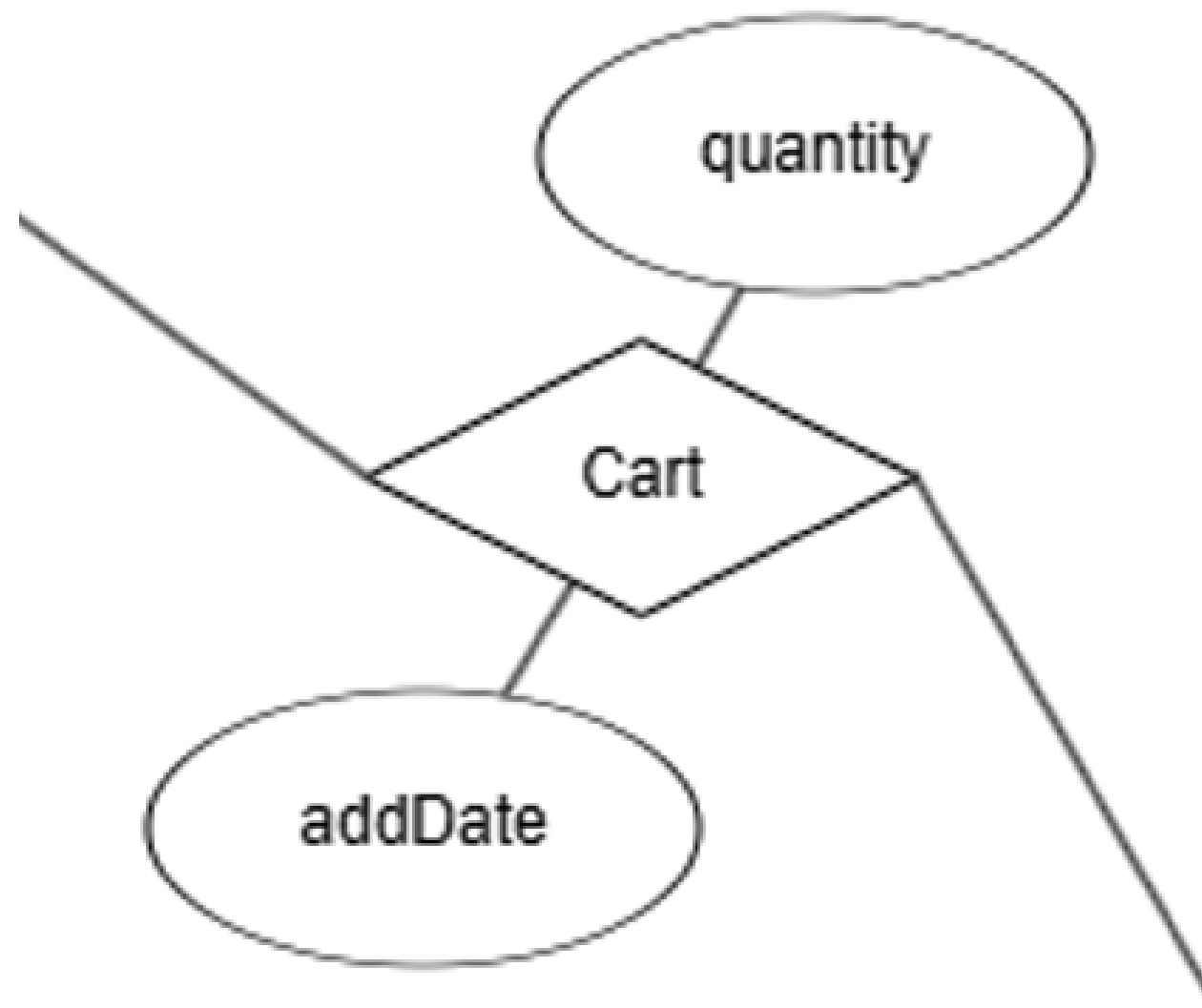
## 11. ShippingUnits



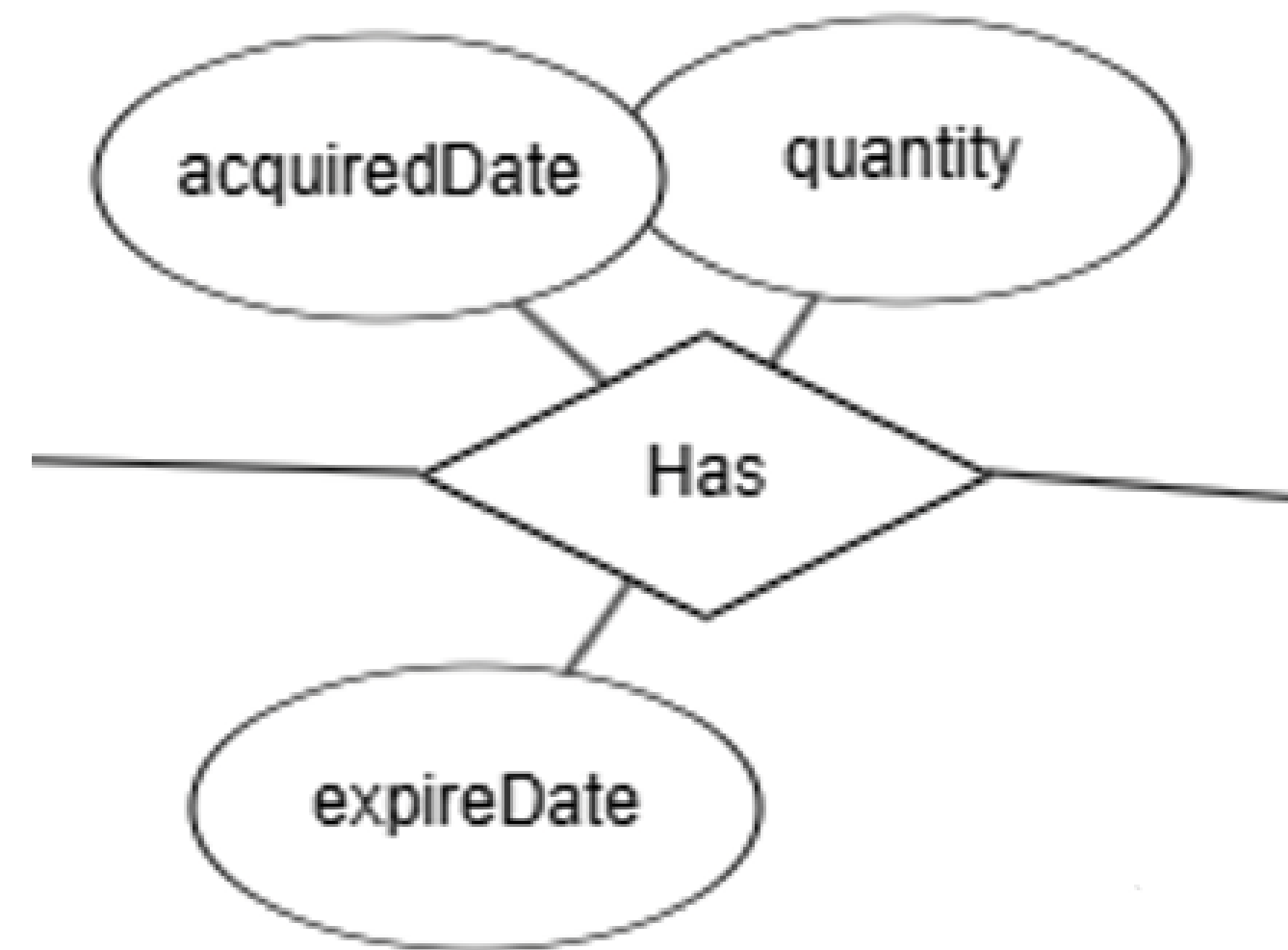


## CÁC QUAN HỆ:

### 1. Cart (Users-Product)

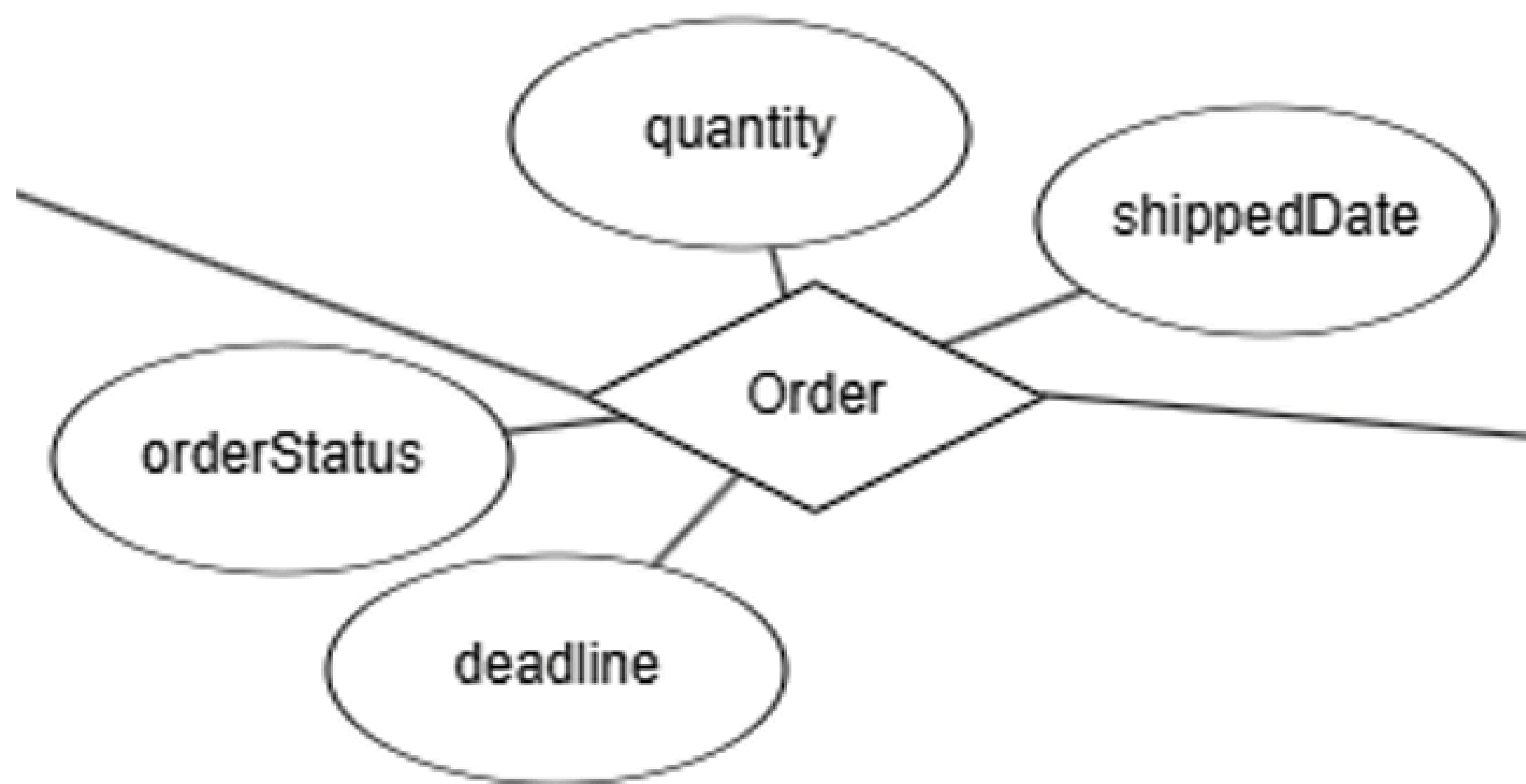


### 2. Has (Users-Voucher)



## CÁC QUAN HỆ:

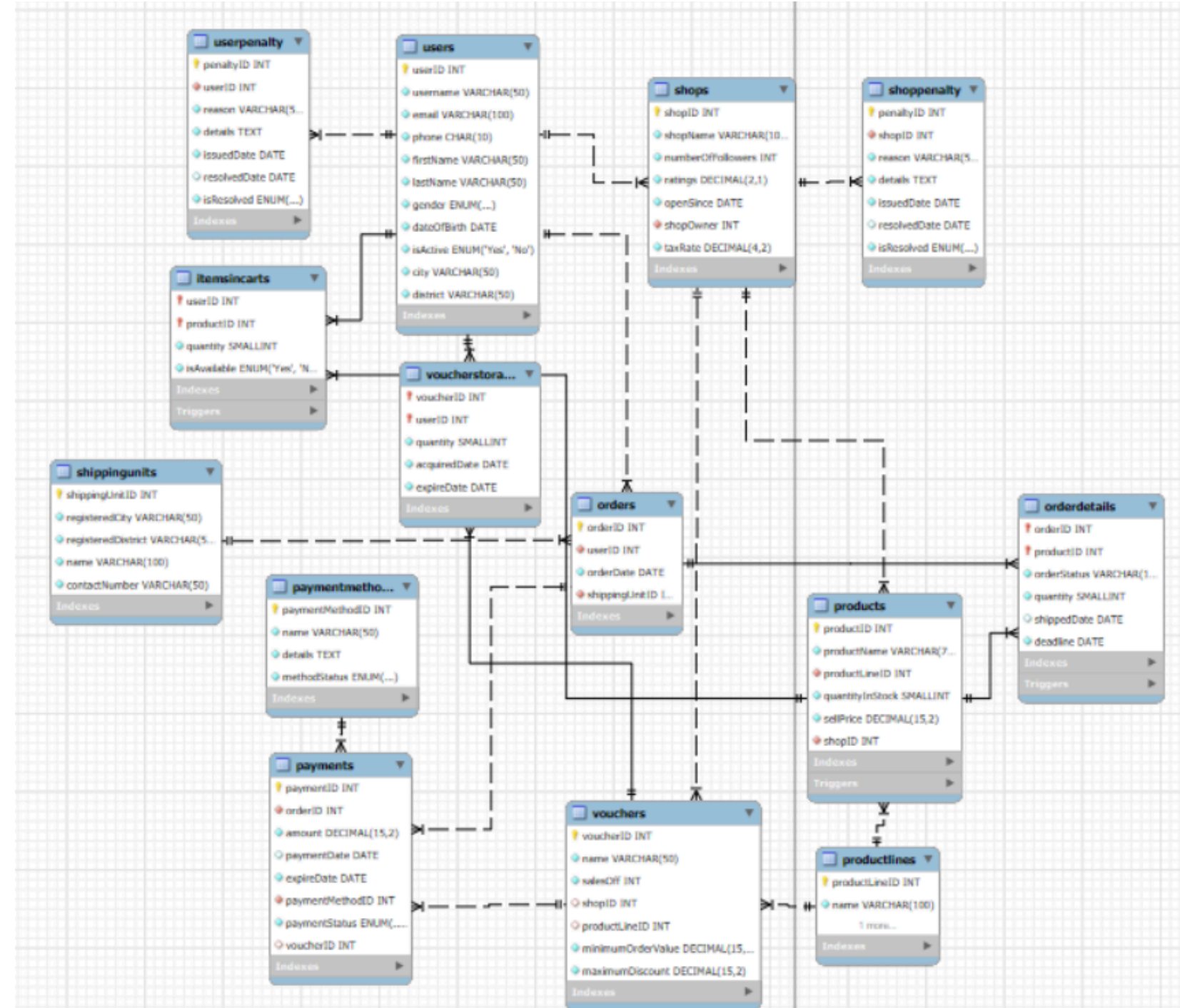
### 3. Order (Orders-Products)



## **CÁC QUAN HỆ:**

- 4. Use (Payments-Voucher)**
- 5. is a (Payments-PaymentMethod)**
- 6. apply for (Voucher-ProductLine)**
- 7. is a (Products-ProductLine)**
- 8. has (Shop-ShopPenalty)**
- ...**

# 3. MH QUAN HỆ, PHỤ THUỘC HÀM



- **users(userID, username, email, phone, firstName, lastName, gender, dateOfBirth, isActive, city, district)**
- **F = {userID -> (username, email, phone, firstName, lastName, gender, dateOfBirth, isActive, city, district),  
username -> (userID, email, phone, firstName, lastName, gender, dateOfBirth, isActive, city, district)}**

- **shops**(shopID, numberOfFollowers, ratings, openSince, shopOwner, taxRate)

**F = {shopID -> (numberOfFollowers, ratings, openSince, shopOwner, taxRate)}**

---

- **products**(productID, productName, productLineID, quantityInStock, sellPrice, shopID)

**F = {productID -> (productName, productLineID, quantityInStock, sellPrice, shopID)}**

- **productLines**(productLineID, name, description)

**F = {productLineID -> (name, description)}**

---

- **payments**( payemntID, orderID, amount, paymentDate, expireDate, paymentMethodID, paymentStatus, voucherID)

**F = {paymentID -> (orderID, orderID, amount, paymentDate, expireDate, paymentMethodID, paymentStatus, voucherID)}**

- **paymentMethods( paymentMethodID, name, details, isActive)**

**F = {paymentMethodID -> ( name, details, isActive)}**

---

- **vouchers( voucherID, name, salesOff, shopID, productLineID,  
minimumOrderValue, maximumDiscount)**

**F = {voucherID -> (name, salesOff, shopID, productLineID,  
minimumOrderValue, maximumDiscount)}**



- **voucherStorage( voucherID, userID, quantity, acquiredDate, expireDate)**

**$F = \{(\text{voucherID}, \text{userID}) \rightarrow (\text{quantity}, \text{acquiredDate}, \text{expireDate})\}$**

---

- **orders(orderID, userID, orderDate, shippingUnitID)**

**$F = \{\text{orderID} \rightarrow (\text{userID}, \text{orderDate}, \text{shippingUnitID})\}$**

- **orderDetails( orderID, productID, orderStatus, shippedDate, deadline, quantity)**

**F = {(orderID, productID) -> (quantity, orderStatus, shippedDate, deadline)}**

---

- **itemsInCart(userID, productID, quantity, isAvailable)**

**F = {(userID, productID) -> (quantity, isAvailable)}**

- **userPenalty**( penaltyID, userID, reason, details, issuedDate, resolvedDate, isResolved)

**F = {penaltyID -> (userID, reason, details, issuedDate, resolvedDate, isResolved)}**

---

- **shopPenalty**( penaltyID, shopID, reason, details, issuedDate, resolvedDate, isResolved)

**F = {penaltyID -> (shopID, reason, details, issuedDate, resolvedDate, isResolved)}**

- **shippingUnits( shippingUnitID, registeredCity, registeredDistrict, name, contactNumber)**

**F = {shippingUnitID -> (registeredCity, registeredDistrict, name, contactNumber)}**

---

# 4. CONSTRAINTS

## DEFAULT CONSTRAINTS:

- Bảng **Payments**:
  - paymentStatus: trạng thái mặc định là 'Pending' nếu không được khai báo
- Bảng **UserPenalty** và **ShopPenalty**:
  - isResolved: Giá trị mặc định là chưa xử lý
- Bảng **OrderDetails**:
  - orderStatus: Giá trị mặc định là 'Processing' nếu không được khai báo.

## INDEX CONSTRAINTS:

- Bảng **Products**: Đặt chỉ mục cho productID.
- Bảng **ProductLines**: Đặt chỉ mục cho productLineID.
- Bảng **ShippingUnits**: Đặt chỉ mục cho shippingUnitID.
- Bảng **Vouchers**: Đặt chỉ mục cho voucherID.
- Bảng **Users**: Đặt chỉ mục cho userID

> Tối ưu hóa tìm kiếm

# CHECK CONSTRAINTS:

## - Bảng **Shops**:

- ratings: Giới hạn giá trị từ 0 đến 5, đảm bảo đánh giá trên thang 5 sao.

## - Bảng **Vouchers**:

- **salesOff**: Giới hạn mức giảm giá từ 0% đến 100%, không cho phép vượt ngoài giới hạn hợp lý.

# 5. TRIGGERS

## - Trigger: `check_product_stock` (bảng ItemsInCarts)

- Sự kiện kích hoạt: Trước khi thêm dữ liệu mới (INSERT).
- Hành động:
  - Kiểm tra số lượng sản phẩm người dùng thêm vào giỏ
  - Điều kiện:  $\leq$  số lượng hàng trong kho
  - Nếu không thỏa mãn: Báo lỗi, không thực hiện thêm dữ liệu

```
CREATE TRIGGER check_product_stock
BEFORE INSERT ON itemsInCarts
FOR EACH ROW
BEGIN
    -- Declare new variable to store available stock
    DECLARE available_stock INT;

    -- Store the quantity in stock to variable available_stock
    SELECT quantityInStock INTO available_stock
    FROM products
    WHERE productID = NEW.productID;

    -- If no product found (i.e., available_stock is NULL), raise an error
    IF available_stock IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Product not found';
    -- Conditional statement to check the quantity added to cart is valid
    ELSEIF NEW.quantity > available_stock THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Insufficient quantity for product';
    END IF;
END;
```



- Trigger: **check\_items\_in\_carts** (bảng Products)

- Sự kiện kích hoạt: Sau khi cập nhật bảng Products (UPDATE).
- Hành động:
  - Kiểm tra lại số lượng sản phẩm trong giỏ hàng của người dùng.
  - Điều kiện:  $\leq$  số lượng hàng trong kho.
  - Nếu không thỏa mãn: Cập nhật thuộc tính isAvailable trong bảng ItemsInCarts thành 'No'

```
CREATE TRIGGER check_items_in_carts
AFTER UPDATE ON products
FOR EACH ROW
BEGIN
    -- compare the new quantity in stock of products with quant
    IF OLD.quantityInStock <> NEW.quantityInStock THEN
        UPDATE itemsInCarts
        SET isAvailable =
            CASE
                WHEN quantity > NEW.quantityInStock THEN 'No'
                ELSE 'Yes'
            END
        WHERE productID = NEW.productID;
    END IF;
END;
```

- Trigger: **check\_valid\_order** (bảng OrderDetails)

- Sự kiện kích hoạt: Trước khi thêm dữ liệu mới (INSERT).
- Hành động:
  - Kiểm tra số lượng sản phẩm đặt hàng của người dùng.
  - Điều kiện:  $\leq$  số lượng hàng trong kho.
  - Nếu không thỏa mãn: Báo lỗi, không thực hiện thêm dữ liệu.

```
CREATE TRIGGER check_valid_order
BEFORE INSERT ON orderDetails
FOR EACH ROW
BEGIN
    -- create new variable to store quantityInStock
    DECLARE available_stock INT;

    -- store the quantityInStock of the product added to orderDetails to available_stock
    SELECT quantityInStock INTO available_stock
    FROM products
    WHERE productID = NEW.productID;

    -- If no product found, raise an error
    IF available_stock IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Product not found';
    -- Check if the ordered quantity exceeds available stock
    ELSEIF NEW.quantity > available_stock THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid order: Ordered quantity exceeds available stock';
    END IF;
END;
```

- Trigger: update\_quantity\_in\_stock(bảng OrderDetails)

- Sự kiện kích hoạt: Sau khi một order được tạo thành công (INSERT).
- Hành động:
  - Tự động trừ số lượng sản phẩm trong kho tương ứng với số lượng đặt hàng.

```
CREATE TRIGGER update_quantity_in_stock
AFTER INSERT ON orderDetails
FOR EACH ROW
BEGIN
    -- update the quantity in stock of the product
    UPDATE products
    SET quantityInStock = quantityInStock - NEW.quantity
    WHERE productID = NEW.productID;
END;
```

# 6. PROCEDURES

- Procedure **applyVoucher** được thiết kế để xử lý việc áp dụng mã giảm giá (voucher) vào một giao dịch thanh toán cụ thể

```
CREATE PROCEDURE applyVoucher(IN applied_voucherID INT, IN applied_paymentID INT)
BEGIN
    -- Declare variables
    DECLARE applied_userID INT;
    DECLARE salesOffValue INT;
    DECLARE paymentAmount INT;
    DECLARE maximumDiscount INT;
    DECLARE minimumOrderValue INT;

    -- Start the transaction to apply the voucher
    START TRANSACTION;

    -- Check if the voucher is valid
    IF NOT is_valid_voucher(applied_voucherID, applied_paymentID) THEN
        ROLLBACK;
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid voucher, try another voucher!';
    END IF;
```

```
-- Check if the user has the voucher
IF NOT has_voucher(applied_voucherID, applied_paymentID) THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'The user does not have the voucher, try another voucher!';
END IF;

-- Retrieve the userID for the payment
SET applied_userID = payments_user(applied_paymentID);
-- If no userID found, exit the procedure
IF applied_userID IS NULL THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'User not found for this payment';
END IF;
```

# 7. TRANSACTION

- Transaction cấp mã giảm giá cho người dùng mới:
  - Thêm một người dùng mới vào hệ thống.
  - Kiểm tra và thêm mã giảm giá dành cho người dùng mới (nếu chưa tồn tại trong hệ thống).
  - Cấp mã giảm giá cho người dùng vừa tạo và lưu trữ trong bảng voucherStorage.

## - Transaction cấp mã giảm giá cho người dùng mới

```
START TRANSACTION;
-- insert new customer
INSERT INTO users(username, email, phone, firstName, lastName, gender, dateOfBirth, isActive, city, district) VALUES
('hoangsonbandon', 'hs.hoangson15062005@gmail.com', '0389261245', 'Hoang', 'Son', 'Male', '2005-6-15', 'Yes', 'Hanoi', 'Hadong');
-- insert voucher for new user if not exist
INSERT INTO vouchers(name, salesOff, shopID, productLineID, minimumOrderValue, maximumDiscount)
SELECT 'New Comer Voucher', 50, NULL, NULL, 100000.00, 1000000.00
WHERE NOT EXISTS (
    SELECT 1 FROM vouchers WHERE name = 'New Comer Voucher'
);
-- insert user's voucher to voucherStorage
INSERT INTO voucherStorage(voucherID, userID, quantity, acquiredDate, expireDate)
SELECT v.voucherID, u.userID, 10, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 30 DAY)
FROM vouchers v
JOIN users u ON u.username = 'hoangsonbandon'
WHERE v.name = 'New Comer Voucher';
ROLLBACK;
```

# 8. QUERIES

## - Truy vấn phương thức thanh toán của các giao dịch

# Find the payment method of the payments

SELECT

payments.\*,  
paymentMethods.\*

FROM payments

JOIN paymentMethods ON payments.paymentMethodID = paymentMethods.paymentMethodID;

paymentID	orderID	amount	paymentDate	expireDate	paymentMethodID	paymentStatus	voucherID
1	1	2538000.00	2024-05-10	2024-05-27	1	Completed	4
2	2	8000000.00	2024-07-20	2024-07-27	2	Completed	5
3	3	850000.00	2024-08-13	2024-08-20	2	Completed	NULL
4	4	1024800.00	2024-10-25	2024-11-10	3	Completed	6
5	5	500000.00	2024-10-19	2024-11-07	1	Completed	NULL

## - Truy vấn mã giảm giá được sử dụng cho giao dịch

# Find the voucher used for the payments

SELECT

payments.\*,

vouchers.\*

FROM payments

LEFT JOIN vouchers ON vouchers.voucherID = payments.voucherID;

paymentID	orderID	amount	paymentDate	expireDate	paymentMethodID	paymentStatus	voucherID	voucherID	name
1	1	2538000.00	2024-05-10	2024-05-27	1	Completed	4	4	CT500
2	2	8000000.00	2024-07-20	2024-07-27	2	Completed	5	5	CTMAX
3	3	850000.00	2024-08-13	2024-08-20	2	Completed	NULL	NULL	NULL
4	4	1024800.00	2024-10-25	2024-11-10	3	Completed	6	6	Giam 30k
5	5	500000.00	2024-10-19	2024-11-07	1	Completed	NULL	NULL	NULL



## - Truy vấn sản phẩm bán chạy nhất của mỗi cửa hàng

```
# Find the best selling products of each shop based on quantity ordered
SELECT shopDetails.shopName, bestSellingProducts.productName, bestSellingProducts.totalQuantity
FROM (SELECT p.shopID, p.productID, p.productName, SUM(od.quantity) AS totalQuantity
      FROM products p
      JOIN orderDetails od ON p.productID = od.productID
      GROUP BY p.shopID, p.productID, p.productName
    ) AS bestSellingProducts
JOIN shops shopDetails
  ON bestSellingProducts.shopID = shopDetails.shopID
WHERE
  bestSellingProducts.totalQuantity = (
    SELECT MAX(totalQuantity)
    FROM ( SELECT p.shopID, p.productID, SUM(od.quantity) AS totalQuantity
          FROM products p
          JOIN orderDetails od ON p.productID = od.productID
          GROUP BY p.shopID, p.productID
        ) AS subQuery
    WHERE subQuery.shopID = bestSellingProducts.shopID
  );
```

	shopName	productName	totalQuantity
▶	Do Go Cong Trinh	Tu bon canh	2
	Do Go Cong Trinh	Ke tivi	2
	Do Go Cong Trinh	Ghe go	2
	Thu Trang Shop	Bitis BSB005100XNG	1
	Thu Trang Shop	Ao giu nhiet nu co cao	1

## 9. TỔNG KẾT

- Thiết kế và triển khai được cơ sở dữ liệu quản lý sản phẩm thương mại điện tử
- Hỗ trợ hiệu quả các hoạt động như quản lý sản phẩm, người dùng, đơn hàng, giỏ hàng, khuyến mãi, và vận chuyển
- Duy trì tính toàn vẹn và hiệu suất của cơ sở dữ liệu bằng cách sử dụng các ràng buộc và triggers