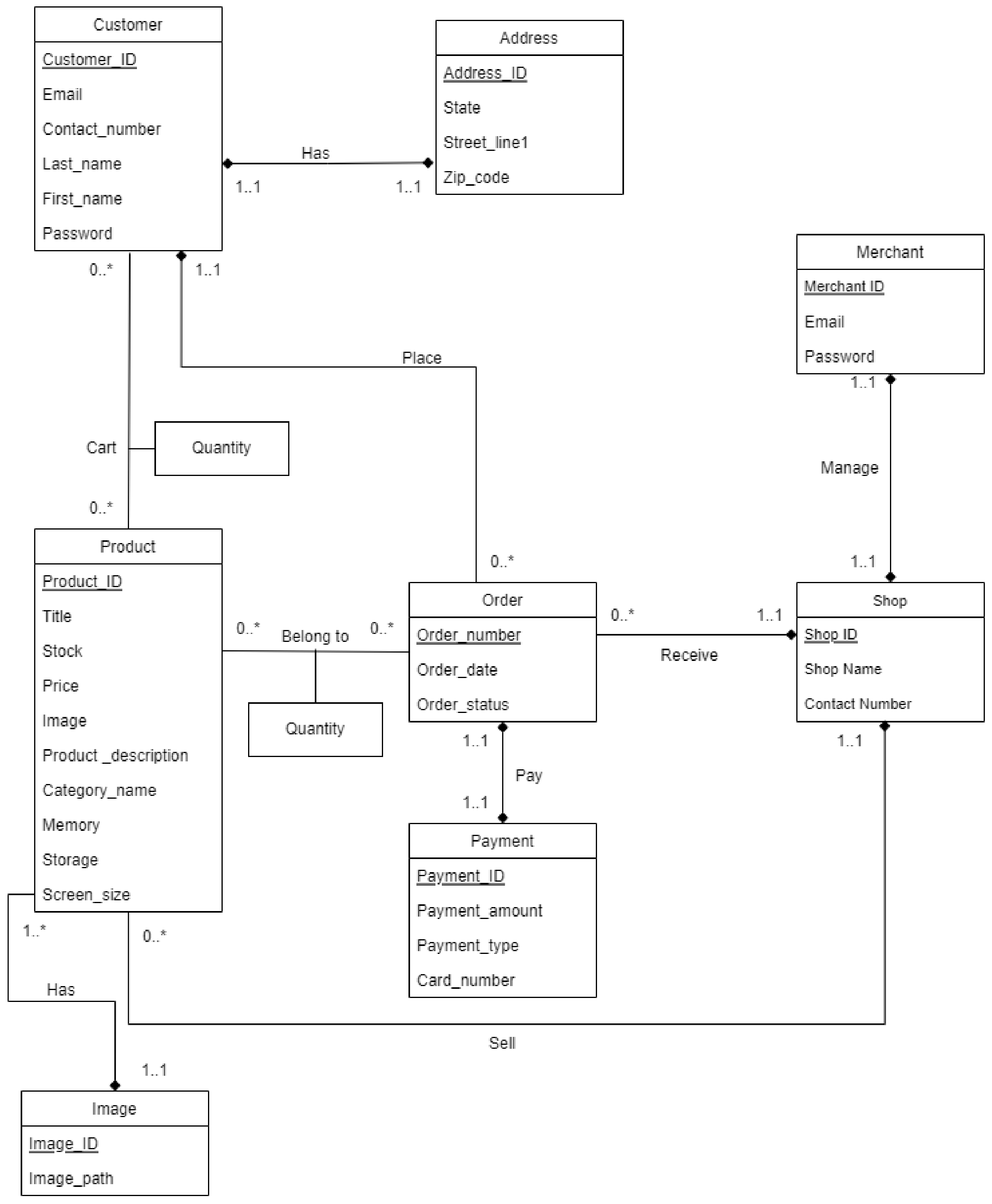
CS 411 Stage2

UML



In this UML, the entity “Product” does not represent one product, but the abstraction of products which have many instances belonging to the same product title in the real world. The category name (like phone, computer, and mp4) represents the category that a product title belongs to.

Relationship Description

1.Product-Order

One kind of product belongs to many orders. One order can have many kinds of products.

2.Product-Shop

One kind of product is only sold by one shop. One shop can sell many kinds of products.

3. Customer-address

The customer could only have one address, and one address belongs to exactly one customer.

4. Customer-Product

The customer could have many products in the cart, and one product could be put in many customers’ carts. There would be a relationship table Cart to store the quantity of products in the cart.

5.Order-payment

An order can be paid exactly once, and a payment can only pay for exactly one order.

6.Order-Customer

An order can be owned by exactly one customer, but a customer can have more than one order at a time.

7. Order-Shop

An order is received by exactly one shop, while a shop receives multiple orders

8. Merchant-Shop

A merchant manages exactly one shop, and a shop is managed by exactly one merchant.

9. Product - Image

A product could only have one image (due to the storage limitation), an image could belong to the products of the same series, because they have the same appearance.

Relational Schema

Table1 Customer

```
Table - Customer (
        Customer_ID: INT [PK],
        Email: VARCHAR(255),
        Contact_number: VARCHAR(10),
        Last_name: VARCHAR (20),
        First_name: VARCHAR (20),
        Password: VARCHAR (20),
        Address_ID: INT [FK to Address.Address_ID])
```

The customer entity has a primary key customerID as its identification, basic information like contact number, Last name, and first name.

The customer entity also has an email and password for the login function. The login page different login portal for merchants and customers, so when a user logs in, the front-end would send requests with role information to the back-end to identify the user.

Customers' address information is stored in the Address table, in order to support the frequent read-only operation in future use.

Table 2 Address

```
Table - Address (
        Address_ID: INT [PK],
        State: VARCHAR(15),
        Street_line: VARCHAR(255),
        Zip_code: INT,
        Customer_ID: INT [FK to Cutomer.Customer_ID])
```

The address entity stores detailed address information for order delivery. The customer could update the address by Editing personal information functionality on the web application. And the customer_ID in the Address table would be set as Cascade to make sure if the customer accidentally is deleted, the address information would be deleted too.

Table 3 Merchant

```
Table - Merchant (
        Merchant_ID: INT [PK],
        Email: VARCHAR(20),
        Password: VARCHAR(20),
        Shop_ID: INT [FK to Shop.Shop_ID])
```

The Merchant table records every manufacturer who provides our inventory's detailed information. The table includes three attributes: Merchant ID, email, and password. The Merchant ID and password are belonged to Merchant’s account information, as every manufacturer can log in to check their products’ selling conditions, and if products are out of stock, warning notice will be sent to their account. We expect every merchant has exactly one account on our website. The Merchant table is uniquely identified by Merchant ID.

Table 4 Shop

Table - Shop (
    Shop_ID: INT [PK],
    Shop Name: VARCHAR(20),
    Contact Number: VARCHAR(20),
    Merchant_ID: INT [FK to Merchant.Merchant_ID])

The Shop table is used to record every shop’s information. Since we expect our website is in self-operated business mode, that our platform plays the role of agency to sell products for each brand, that we obtain inventory from each manufacturer (which is our Merchant), and we sell the products on our own. Hence, we assume that we have exactly one shop on our website for each Merchant, which has three attributes: Shop ID, Shop Name, and Contact Number. Each shop sells many Products, and receives many Orders. The Shop table is uniquely identified by Shop ID.

Table 5 Product

Table – Product (
  Product_ID: INT [PK],
  Title: VARCHAR(20),
  Stock: INT,
  Price: INT,
  Product _description: VARCHAR(255),
  Category_name: VARCHAR(20),
  Shop_ID: INT [FK to Shop.Shop_ID],
  Image_ID: INT [FK to Image.Image_ID])

The product table records the details of a kind of product such as its ID, title, stock, price, and so on. The PK Product_ID represents each unique kind of product. The FK SHOP_ID represents which shop sells the product. The FK Image_ID represents the image of the product.

Table 6 Cart_Product

Table - Cart_Product(
    CustomerID: INT [FK to Customer.Customer_ID],
    Product_ID: INT [FK to Product.Product_ID],
    Quantity: INT)

The Cart_Product table includes the information of carts. One cart may have different kinds of products, so in this table, one cart will have several tuples to record the quantity of each kind of product in this cart. The FK Customer_ID tells which customer the cart belongs to.

Table 7 Order

```
Table - Order (
        Order_number: INT [PK],
        Customer_ID: INT [FK to table.Customer],
        Payment_ID: INT [FK to table.Payment],
        Shop_ID: INT [FK to table.Shop],
        Product_ID: INT [FK to table.Product],
        Order_status: VARCHAR(20),
        Order_date: VARCHAR(20))
```

Order table is used to record the information of each order. It is uniquely identified by Order_number. Other attributes are Order_status and Order_date. Customer_ID, Payment_ID, Shop_ID, and Product_ID are all foreign keys for this table. And they would be set as SET NULL to ensure the security and integrity of order information. In this table, we can look up the order creation time and status information, like merchant shipped and order completed.

Table 8 payment

```
Table - Payment (
        Payment_ID: INT [PK],
        Payment_amount: real,
        Payment_type: VARCHAR(20),
        Card_number: INT)
```

Payment table is used to describe payment information. It is uniquely identified by Payment_ID. Other attributes are Payment_amount, Payment_type and Card_number. We will record the payment method for each payment, like credit or debit cards, and also the amount of the payment.

Table 9 Product-order

```
Table - Product_order(
        Order_number: INT [FK to Order.Order_number],
        Product_ID: INT [FK to Product.Product_ID],
        Quantity: INT
)
```

This product - order is a relationship table for Product and Order table, store the information about product quantity in order. An order could have more than one tuple in this Product-order table, reference to different products.

Table 10 Shop - order

```
Table - Shop_order(
        Shop_ID: INT [FK to Shop.Shop_ID],
        Order_number: INT [FK to Order.Order_number]
        )
```

This Shop_order table is a relationship table for Shop and Order table, store the information about orders been received by shop. A shop could have more than one tuple in this Shop_order table.

Table 11 Image

```
Table - Image(
	Image_ID: INT [PK],
	Image_path:VARCHAR(255)
)
```

This image table stored the image path for products. One product could only have one image due to the limitation of database storage, while one image could belong to multiple products from the same series because they have the same appearance.