Introduction:

E-commerce, also known as electronic commerce or internet commerce, refers to the buying and selling of goods or services using the internet, and the transfer of money and data to execute these transactions.

E-Commerce is an online store where customers can browse categories and select products of interest. The selected items may be collected in a shopping cart. the items in the shopping cart will be presented as an order if customer submit it. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e-mail notification is sent to the customer as soon as the order is placed.

Types of E-commerce Models:

There are four main types of ecommerce models that can describe almost every transaction that takes place between consumers and businesses.

1. Business to Consumer (B2C):

When a business sells a good or service to an individual consumer

2. Business to Business (B2B):

When a business sells a good or service to another business (e.g. A business sells software-as-a-service for other businesses to use)

3. Consumer to Consumer (C2C):

When a consumer sells a good or service to another consumer (e.g. You sell your old furniture on eBay to another consumer).

4. Consumer to Business (C2B):

When a consumer sells their own products or services to a business or organization

Software system scope:

In our scope we won’t be able to cover all aspects of the system but we will cover them later. So now, we will discuss our scope showing which aspects including in our system and which is not.

Our system focused on **Business to Consumer (B2C) model** which declared before.

Aspects covered in our System Scope:

1- Browse categories to choose certain product.

2- Ordering the selected products.

3- Give a feedback to any product.

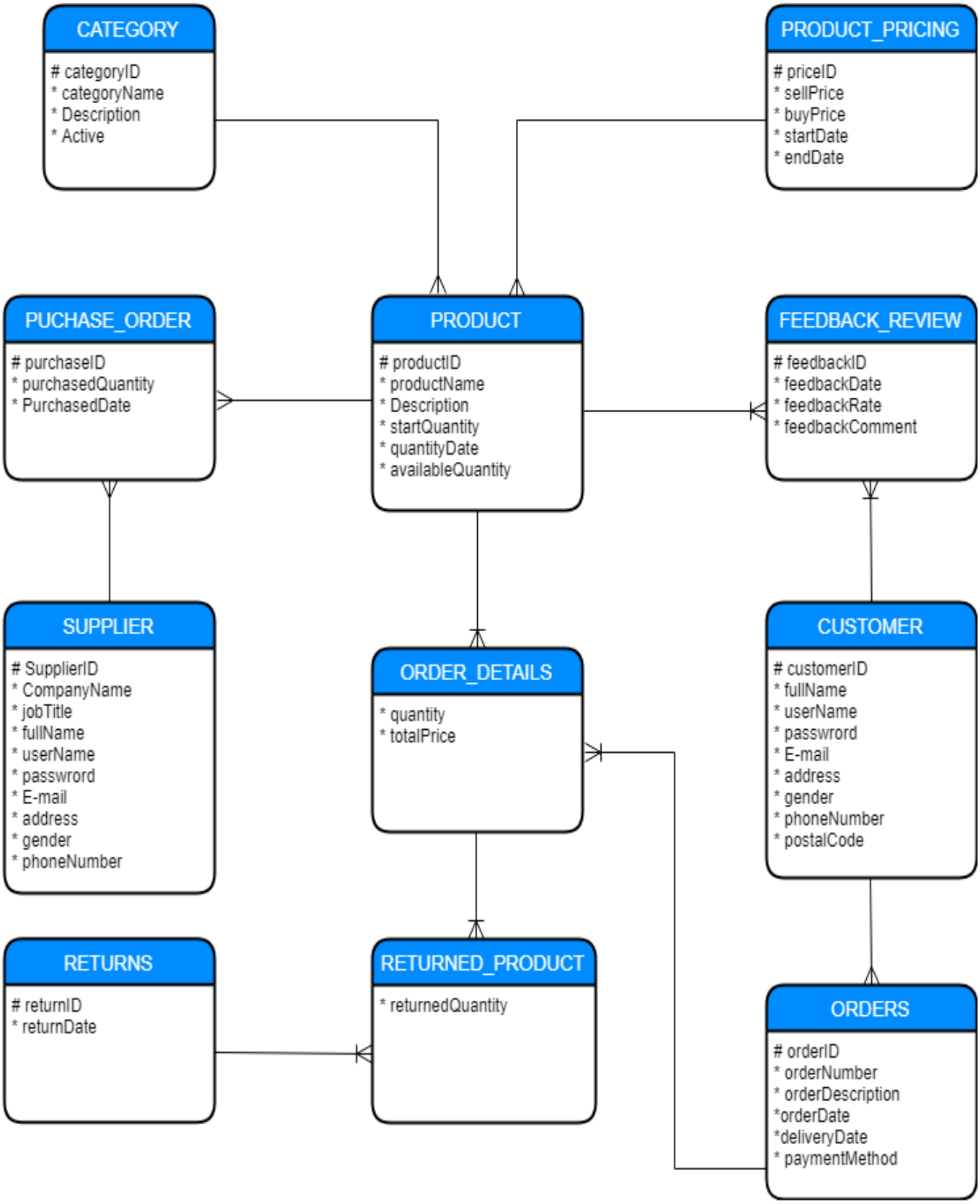
4- Return any product we don’t need after ordering it at the specified time.

Aspects not covered in our System Scope:

1- Tracking orders.

2- Shipment Companies.

ERD:


CATEGORY
categoryID
* categoryName
* Description
* Active
PRODUCT_PRICING
priceID
* sellPrice
* buyPrice
* startDate
* endDate
PUCHASE_ORDER
purchaseID
* purchasedQuantity
* PurchasedDate
PRODUCT
productID
* productName
* Description
* startQuantity
* quantityDate
* availableQuantity
FEEDBACK_REVIEW
feedbackID
* feedbackDate
* feedbackRate
* feedbackComment
SUPPLIER
SupplierID
* CompanyName
* jobTitle
* fullName
* userName
* passwrord
* E-mail
* address
* gender
* phoneNumber
ORDER_DETAILS
* quantity
* totalPrice
CUSTOMER
customerID
* fullName
* userName
* passwrord
* E-mail
* address
* gender
* phoneNumber
* postalCode
RETURNS
returnID
* returnDate
RETURNED_PRODUCT
* returnedQuantity
ORDERS
orderID
* orderNumber
* orderDescription
*orderDate
*deliveryDate
* paymentMethod

SQL Commands:

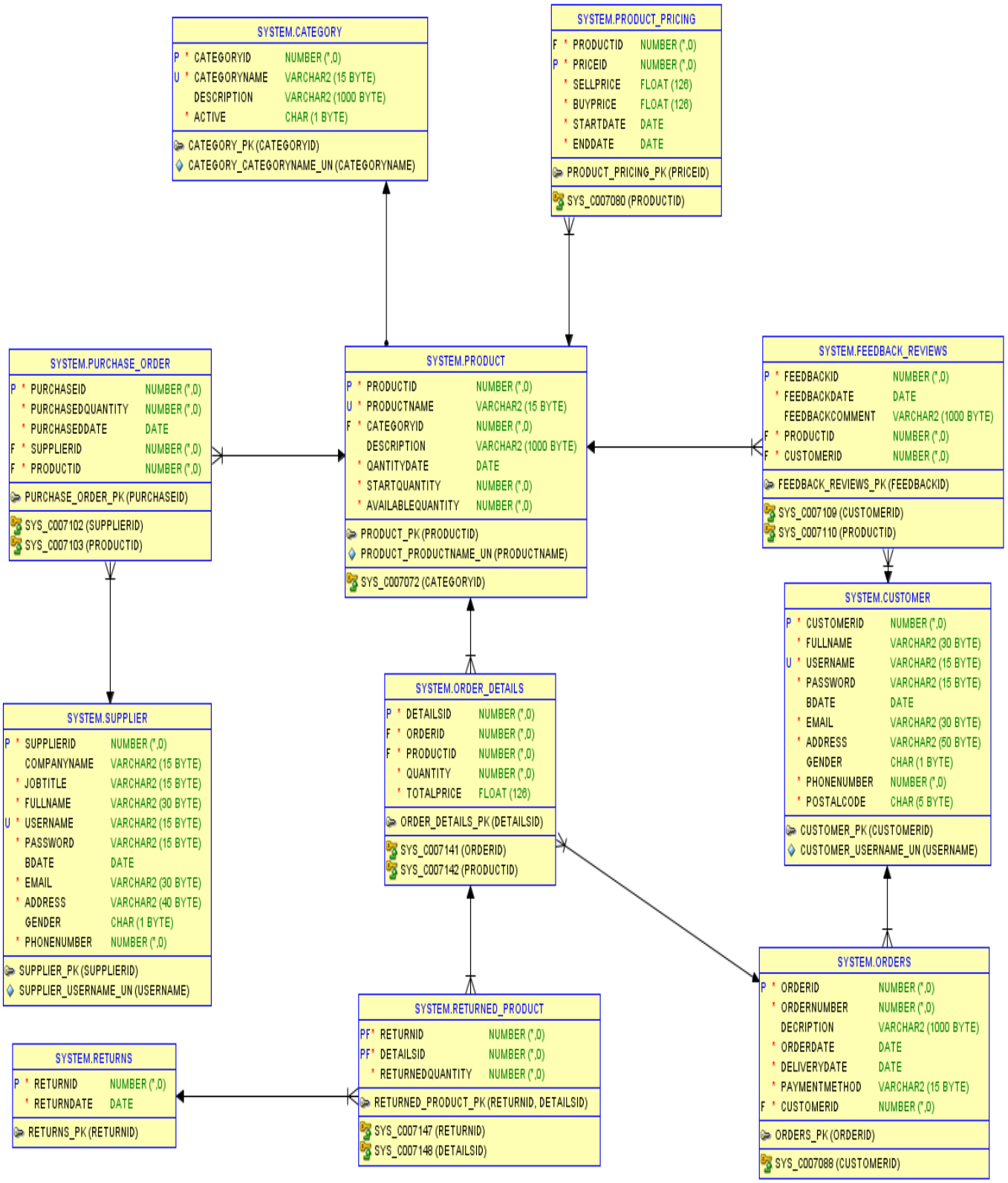
```sql
CREATE TABLE CUSTOMER
(
    CustomerID INT NOT NULL,
    FullName VARCHAR(30) NOT NULL,
    UserName VARCHAR(15) NOT NULL,
    Password VARCHAR (15) NOT NULL,
    Bdate DATE,
    Email VARCHAR(30) NOT NULL,
    Address VARCHAR(50) NOT NULL,
    Gender CHAR,
    PhoneNumber INT NOT NULL,
    PostalCode CHAR(5) NOT NULL,
    PRIMARY KEY (CustomerID),
    UNIQUE (UserName)
);
CREATE TABLE SUPPLIER
(
    SupplierID  INT           NOT NULL,
    CompanyName VARCHAR(15),
    JobTitle    VARCHAR(15)   NOT NULL,
    FullName    VARCHAR(30)   NOT NULL,
    UserName    VARCHAR(15)   NOT NULL,
    Password    VARCHAR(15)   NOT NULL,
    Bdate DATE,
    Email       VARCHAR(30)   NOT NULL,
    Address     VARCHAR(40)   NOT NULL,
    Gender      CHAR,
    PhoneNumber INT       NOT NULL,
    PRIMARY KEY(SupplierID),
    UNIQUE (UserName)
);
```

```
33 CREATE TABLE CATEGORY
34 (
35     CategoryID       INT              NOT NULL,
36     CategoryName     VARCHAR(15)      NOT NULL,
37     Description      VARCHAR2(1000),
38     Active           CHAR              NOT NULL,
39     PRIMARY KEY (CategoryID),
40     UNIQUE(CategoryName)
41 );
42 CREATE TABLE PRODUCT
43 (
44     ProductID           INT          NOT NULL ,
45     ProductName         VARCHAR(15) NOT NULL,
46     CategoryID          INT           NOT NULL,
47     Description         VARCHAR2(1000) ,
48     QantityDate         DATE         NOT NULL,
49     StartQuantity       INT          NOT NULL,
50     AvailableQuantity   INT          NOT NULL,
51     PRIMARY KEY (ProductID),
52     UNIQUE(ProductName) ,
53     FOREIGN KEY (CategoryID) REFERENCES CATEGORY (CategoryID)
54 );
56 CREATE TABLE PRODUCT_PRICING
57 (   ProductID           INT       NOT NULL ,
58     PriceID             INT       NOT NULL,
59     SellPrice           FLOAT     NOT NULL,
60     BuyPrice            FLOAT     NOT NULL,
61     StartDate           DATE      NOT NULL,
62     EndDate             DATE      NOT NULL ,
63     PRIMARY KEY (PriceID),
64     FOREIGN KEY (ProductID) REFERENCES PRODUCT (ProductID)
65 );
```

```
66 CREATE TABLE ORDERS
67 (
68     OrderID              INT          NOT NULL,
69     OrderNumber          INT          NOT NULL,
70     Decription           VARCHAR2(1000),
71     OrderDate            DATE         NOT NULL ,
72     DeliveryDate         DATE         NOT NULL,
73     PaymentMethod        VARCHAR(15)  NOT NULL,
74     CustomerID           INT          NOT NULL,
75     PRIMARY KEY (OrderID),
76     FOREIGN KEY (CustomerID) REFERENCES CUSTOMER (CustomerID)
77 );
78 CREATE TABLE ORDER_DETAILS
79
80 (
81     DetailsID            INT          NOT NULL,
82     OrderID              INT          NOT NULL,
83     ProductID            INT          NOT NULL ,
84     Quantity             INT             NOT NULL ,
85     TotalPrice           FLOAT        NOT NULL,
86     PRIMARY KEY(DetailsID),
87     FOREIGN KEY (OrderID)REFERENCES ORDERS(OrderID),
88     FOREIGN KEY (ProductID)REFERENCES PRODUCT(ProductID)
89 );
90 CREATE TABLE PURCHASE_ORDER
91 (
92     PurchaseID           INT          NOT NULL ,
93     PurchasedQuantity    INT          NOT NULL,
94     PurchasedDate        DATE         NOT NULL ,
95     SupplierID           INT          NOT NULL,
96     ProductID            INT          NOT NULL ,
97     PRIMARY KEY (PurchaseID),
98     FOREIGN KEY (SupplierID) REFERENCES SUPPLIER (SupplierID),
99     FOREIGN KEY (ProductID) REFERENCES PRODUCT (ProductID)
100 );
```

```
101 CREATE TABLE FEEDBACK_REVIEWS
102 (
103     FeedbackID            INT          NOT NULL,
104     FeedbackDate          DATE         NOT NULL,
105     FeedbackComment       VARCHAR2(1000),
106     ProductID            INT           NOT NULL ,
107     CustomerID           INT           NOT NULL,
108     PRIMARY KEY (FeedbackID),
109     FOREIGN KEY (CustomerID)REFERENCES CUSTOMER(CustomerID),
110     FOREIGN KEY (ProductID)REFERENCES PRODUCT(ProductID)
111
112 );
113 CREATE TABLE RETURNS
114 (
115     ReturnID         INT NOT NULL,
116     ReturnDate       DATE  NOT NULL,
117     PRIMARY KEY (ReturnID)
118 );
119 CREATE TABLE RETURNED_PRODUCT
120 (   ReturnID            INT         NOT NULL,
121     DetailsID           INT          NOT NULL,
122     ReturnedQuantity    INT         NOT NULL,
123     PRIMARY KEY (ReturnID,DetailsID),
124     FOREIGN KEY (ReturnID) REFERENCES RETURNS (ReturnID),
125     FOREIGN KEY (DetailsID) REFERENCES ORDER_DETAILS (DetailsID)
126
127 );
```

Created ERD for all tables:



ERD Explanation:

Relationships between entities:

1- Category to Product:

one to many. we have several products in category but each product lies in one category.

2- Product to product_pricing:

one to many. price of each product varies by the time.

3- Product to Feedback_Review:

one to many. Every product has many feedbacks but every feedback has a certain product.

4- Customer to Orders:

one to many. Customer may order more than one time but every order has one customer only.

5- Customer to Feedback-Review:

one to many. Customer may give many feedbacks to products but every feedback on this product is gave by this customer only. No one has the same feedback for sure.

6- Product to Order:

many to many. Customer may order many products at the same time, so in this order we have many products. And every product is ordered by many customers. So, we created a new entity which is called "**Order_Details**". It will inform us the quantity of products ordered by the same customer and the total price for these products.

7- Returns to Order_Details:

many to many. Products in the order may be returned or part of them. So, we created a new entity to know which products will be returned which is called "**Returned_Product**". It will inform us the quantity of the products that will be returned.

8- Supplier to Purchase_Order:

one to many. Supplier can supply many products

9- Product to Purchase_Order:

one to many. Products can be