



TRƯỜNG ĐẠI HỌC FPT
MÔN DBI202

Báo cáo Assignment

Nhóm thực hiện : nhóm 1

- Nguyễn Đình Huy - HE171288
- Lưu Bá Quý Dương - HE172166
- Trần Trọng Vũ - HE170469
- Mai Quốc Tuấn - HE176869
- Nguyễn Văn Trường - HE172166

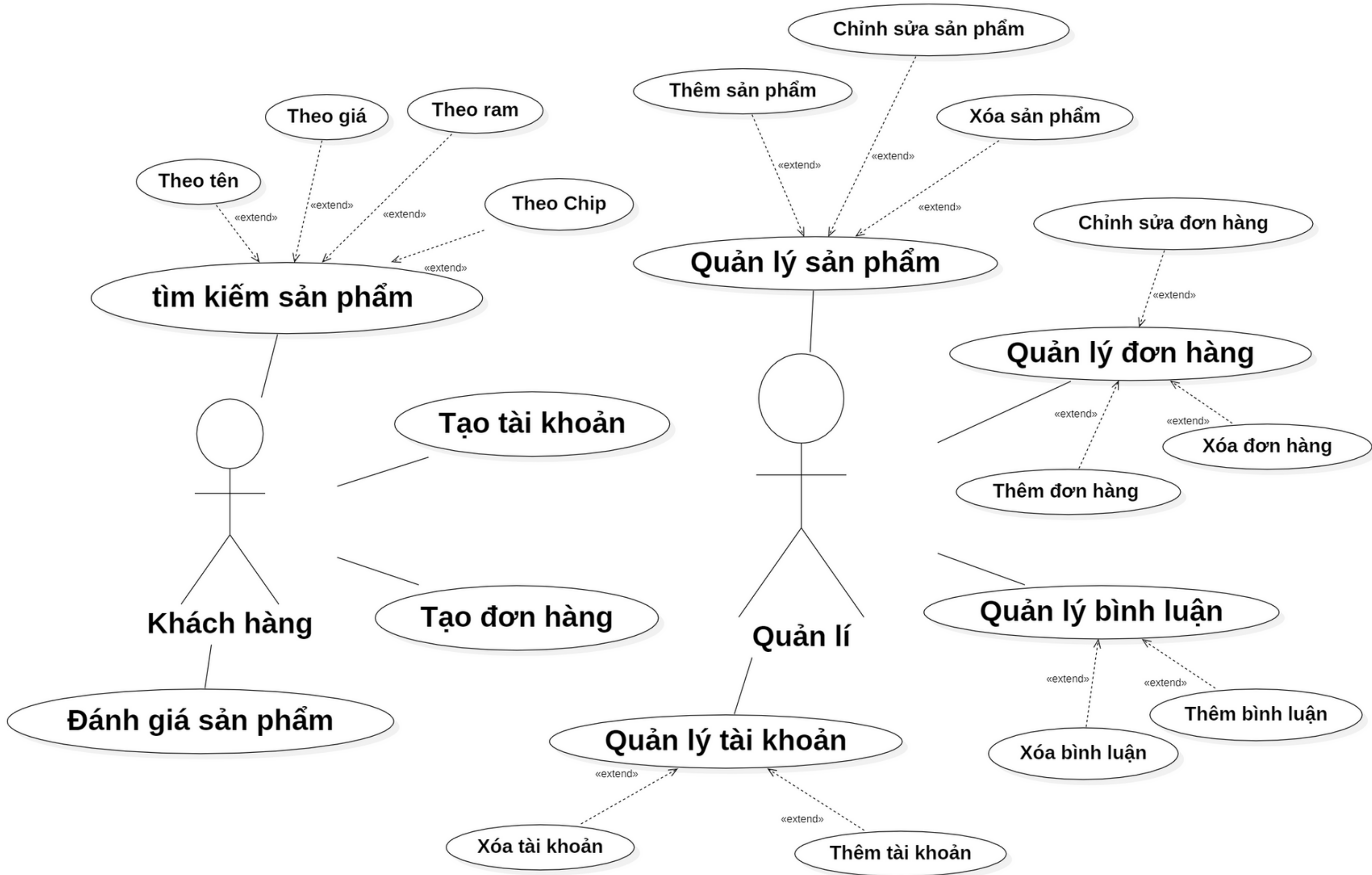
Giảng viên hướng dẫn : Nguyễn Thúy Hương

Cách thiết lập một lớp học ảo



Use case

Phone Shop



Use case

Đối với người dùng

- Xem thông tin điện thoại: Xem thông tin điện thoại (bao gồm tên, giá, ram, chip, năm sản xuất).
- Tìm kiếm sản phẩm: theo tên, giá, chip, dung lượng
- Tạo đơn hàng: tạo tạo đơn hàng
- Bình luận: Đánh giá về sản phẩm đã mua.
- Tạo tài khoản: tạo tài khoản để mua hàng

Đối với quản lý

- Quản lý sản phẩm: Quản lý thông tin điện thoại ((bao gồm tên, giá, ram, chip, năm sản xuất).
- Quản lý danh mục sản phẩm: thêm mục sản phẩm theo hãng.
- Quản lý tài khoản người dùng: Xóa hoặc thêm tài khoản.
- Bình luận phản hồi: xóa hoặc bình luận của người mua.

Use case

Requirements

1. Người dùng có thể tìm kiếm sản phẩm theo các tiêu chí sau:

- Tên sản phẩm
- Giá sản phẩm
- Dung lượng
- Chip

2. Người dùng tạo tài khoản cần cung cấp các thông tin sau:

- Tên
- Số điện thoại
- Địa chỉ
- Email

- Admin quản lý các tài khoản của người dùng có thể xóa tài khoản người dùng.

3. Người dùng tạo đơn hàng mua sản phẩm

- Tên sản phẩm
- Số lượng
- Ngày order
- Giá sản phẩm

- Admin có thể quản lý các đơn hàng.

Use case

4. Người dùng chọn sản phẩm muốn mua. Quản lý sẽ kiểm tra và xác nhận các thông tin sau:

- Tên
- Số điện thoại
- Địa chỉ
- Email
- Tên sản phẩm
- Số lượng
- Ngày order
- Giá sản phẩm

5. Admin có thể quản lý sản phẩm, thêm sản phẩm mới cần cung cấp các thông tin sau:

- Tên nước sản xuất
- Tên hãng sản xuất
- Tên sản phẩm
- Giá nhập
- Dung lượng
- Chip
- Năm sản xuất

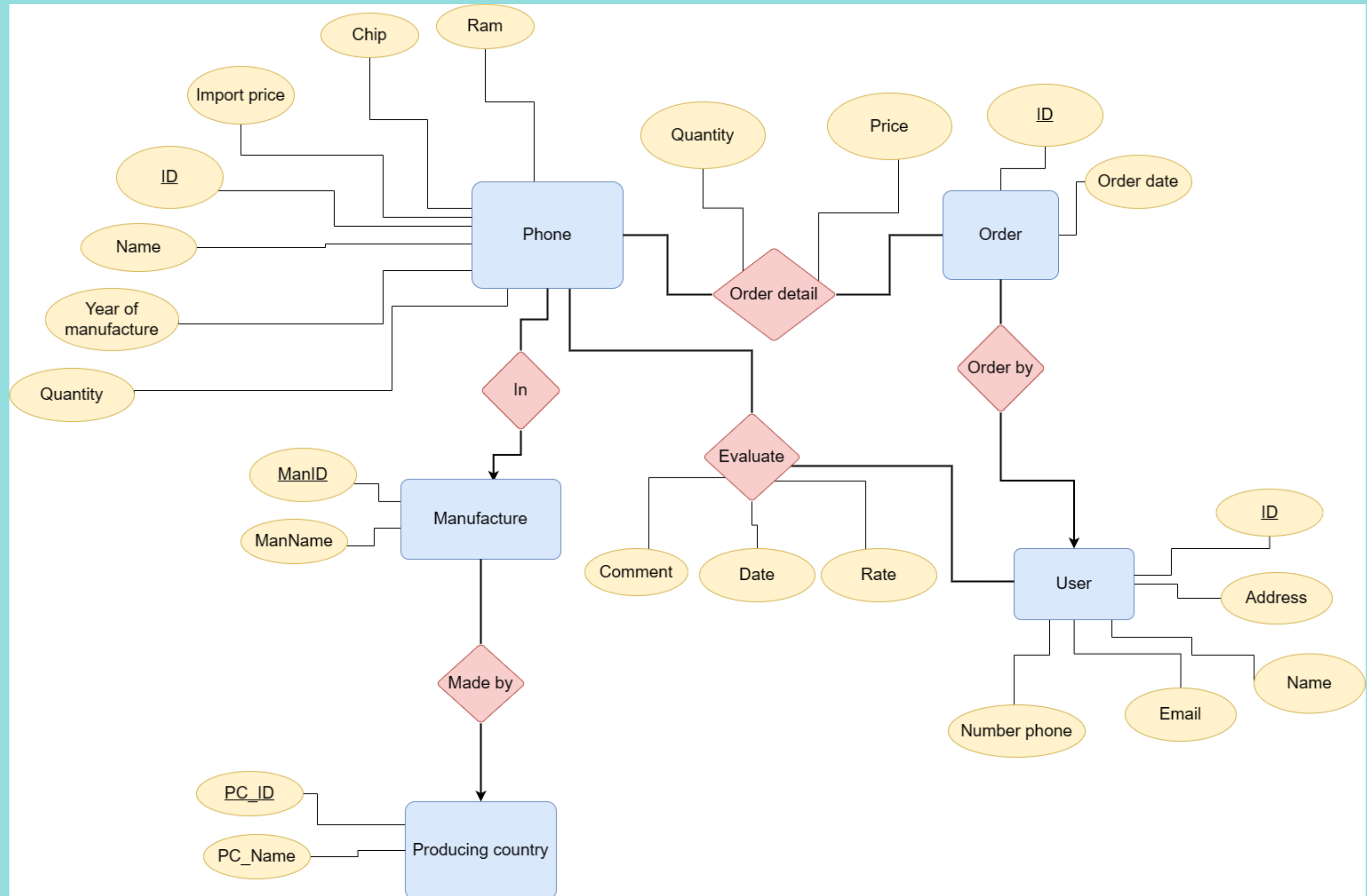
Use case

6. Người dùng đánh giá sản phẩm

- Comment
- Rate
- Ngày đánh giá

Quản lý có thể xóa và quản lý các comments.

ERD



Convert ERD to Tables:

- ProducingCountry(PcID, PcName)

- Manufacture(Man_ID, ManName, PcID)

- PcID references to ProducingCountry(PcID)

- Phone (PhoneID, PhoneName, ManID, Quantity, Import price, Ram, Chip, Year)

ManID references to Manufacture(ManID)

- User(UserID, Name, Phone, Email, Address)

- Order(OrderID, Orderdate, UserID)

UserID references to User(UserID)

- OrderDetails(OrderID, PhoneID, Quantity, Price)

OderID references to Order(OrderID)

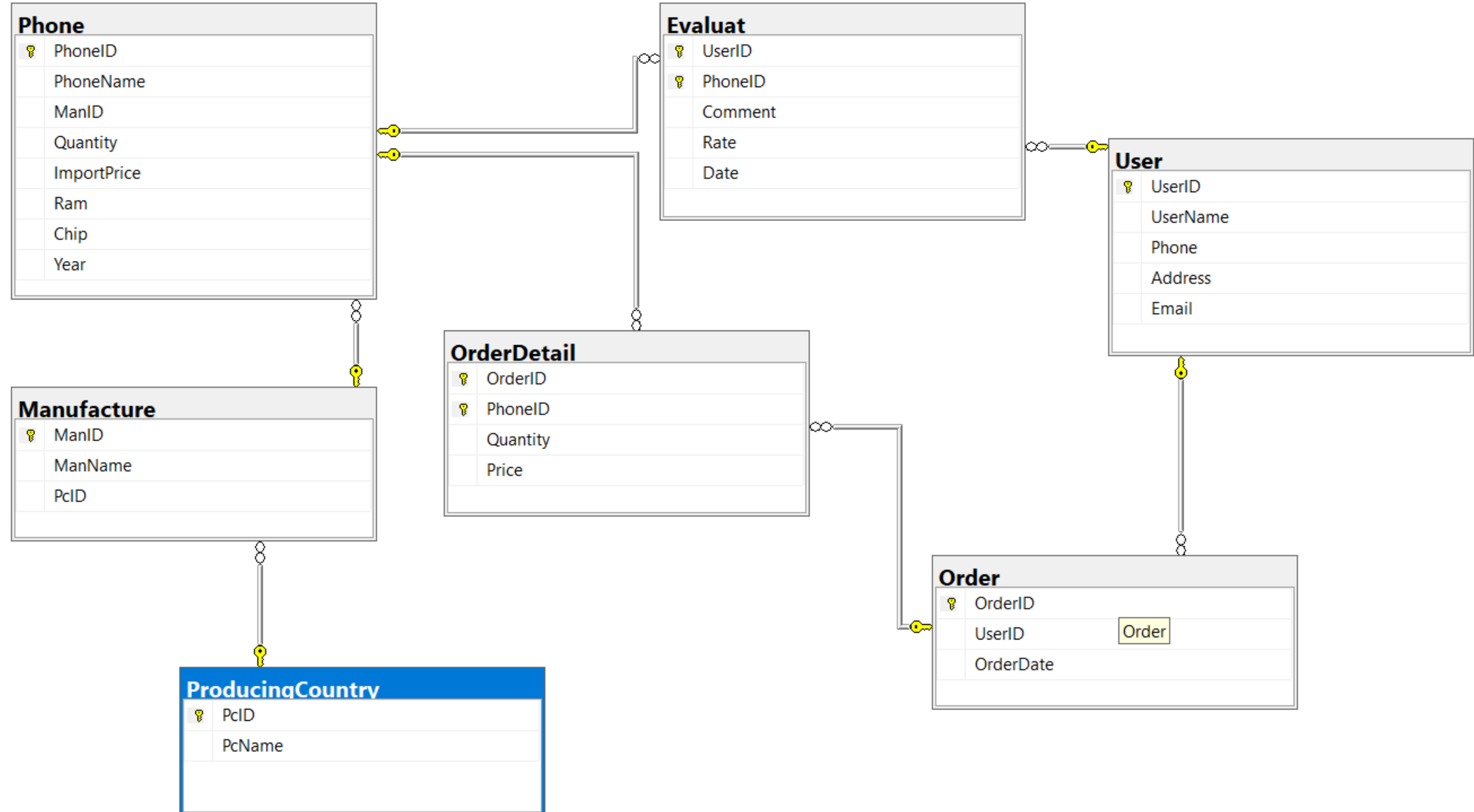
PhoneID references to Phone(PhoneID)

- Evaluate (UserID, PhoneID, Rate, Comment, Date)

PhoneID references to Phone(PhoneID)

UserID references to User(UserID)

Diagram database



SQL DATABASE

Tạo Database Phone_shop

```
USE [master]
```

```
GO
```

```
/*  
Drop database if it exists  
***/
```

```
IF EXISTS (SELECT name FROM master.dbo.sysdatabases WHERE name = N'Phone_Shop')
```

```
BEGIN
```

```
    ALTER DATABASE [Phone_Shop] SET OFFLINE WITH ROLLBACK IMMEDIATE;
```

```
    ALTER DATABASE [Phone_Shop] SET ONLINE;
```

```
    DROP DATABASE [Phone_Shop];
```

```
END
```

```
GO
```

```
CREATE DATABASE Phone_Shop
```

```
GO
```

SQL DATABASE

Tạo Table

```
USE Phone_Shop
GO

/*****
Drop tables if exists
*****/

DECLARE @sql nvarchar(MAX)
SET @sql = N''

SELECT @sql = @sql + N'ALTER TABLE ' + QUOTENAME(KCU1.TABLE_SCHEMA)
+ N'.' + QUOTENAME(KCU1.TABLE_NAME)
+ N' DROP CONSTRAINT ' -- + QUOTENAME(rc.CONSTRAINT_SCHEMA) + N'.' -- not in MS-SQL
+ QUOTENAME(rc.CONSTRAINT_NAME) + N'; ' + char(13) + char(10)
FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS AS RC

INNER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE AS KCU1
ON KCU1.CONSTRAINT_CATALOG = RC.CONSTRAINT_CATALOG
AND KCU1.CONSTRAINT_SCHEMA = RC.CONSTRAINT_SCHEMA
AND KCU1.CONSTRAINT_NAME = RC.CONSTRAINT_NAME

EXECUTE(@sql)
GO

DECLARE @sql2 nvarchar(max)=''

SELECT @sql2 += ' Drop table ' + QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) + '; '
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'

Exec Sp_executesql @sql2
GO
```

SQL DATABASE

Tạo Table

```
create table [ProducingCountry]
(
    [PcID] nvarchar(10) NOT NULL ,
    [PcName] nvarchar(20) NOT NULL,
    PRIMARY KEY (PcID)
)
go

create table [Manufacture]
(
    [ManID] nvarchar(20) NOT NULL ,
    [ManName] nvarchar(20) NOT NULL,
    [PcID] nvarchar(10) NOT NULL,
    PRIMARY KEY (ManID),
    foreign key([PcID]) references [dbo].[ProducingCountry]
)
go

create table [Phone]
(
    [PhoneID] nvarchar(20) NOT NULL primary key ,
    [PhoneName] nvarchar(20) NOT NULL,
    [ManID] nvarchar(20) NOT NULL ,
    [Quantity] int CHECK ([Quantity]>0),
    [ImportPrice] int CHECK ([ImportPrice]>0),
    [Ram] nvarchar(30) NOT NULL,
    [Chip] nvarchar(50) NOT NULL,
    [Year] decimal(4,0) NOT NULL
    foreign key([ManID]) references [dbo].[Manufacture]
)
go
```

```
create table [User]
([UserID] nvarchar(10) NOT NULL primary key,
 [UserName] nvarchar(30) NOT NULL,
 [Phone] nvarchar(20) NULL,
 [Address] nvarchar(250) NULL,
 [Email] nvarchar(50) NULL )
go

create table [dbo].[Order]
([OrderID] int NOT NULL PRIMARY KEY (OrderID) ,
 [UserID] nvarchar(10) FOREIGN KEY REFERENCES [User](UserID),
 [OrderDate] datetime CHECK([OrderDate]<getdate()) )
go

create table [dbo].[OrderDetail]
([OrderID] int NOT NULL FOREIGN KEY REFERENCES [Order](OrderID),
 [PhoneID] nvarchar(20) NOT NULL FOREIGN KEY REFERENCES [Phone](PhoneID),
 PRIMARY KEY (OrderID, [PhoneID]),
 [Quantity] int CHECK ([Quantity]>0),
 [Price] int )
go

create table [dbo].[Evaluat]
([UserID] nvarchar(10) NOT NULL references [dbo].[User](UserID),
 [PhoneID] nvarchar(20) NOT NULL references [dbo].[Phone](PhoneID),
 [Comment] nvarchar(50) ,
 [Rate] int ,
 [Date] date,
 check([Rate] >= 1 AND [Rate] <= 5),
 CONSTRAINT ID_Evaluat PRIMARY KEY (UserID, PhoneID) )
go
```


● Function

- Description: Tạo một hàm có tên là “TotalOfPrice1” sẽ tính được tổng tiền phải trả cho mỗi hóa đơn.
- Có thể truyền vào tham số OrderID
- Các cột trong bảng:
 - OrderID : ID of hóa đơn
 - Total Price Of Order : Tổng giá bán trên mỗi hóa đơn.

```
create function fnTotalOfPrice1
(@OrderID int)
returns int
as
begin
    declare @TotalPrice int
    select @TotalPrice = (select SUM([Total Price]) as [Total Price of Order]
                        from (SELECT od.OrderID, p.PhoneName, od.Quantity,
                                (od.Price * od.Quantity) AS [Total Price],
                                (od.Price - p.ImportPrice) * od.Quantity AS [Total Profit]
                                FROM OrderDetail od JOIN Phone p ON od.PhoneID = p.PhoneID)
                        as d where d.OrderID = @OrderID)

    return @TotalPrice
end
```

• Function

Demo

```
select distinct OrderID, dbo.fnTotalOfPrice1(OrderID) as [Total Price of Order] from OrderDetail
```

	OrderID	Total Price of Order
1	1	3320
2	2	600
3	3	600
4	4	870
5	5	800
6	6	870
7	7	750
8	8	870
9	9	850
10	10	800
11	11	750
12	12	850
13	13	800
14	14	750
15	15	1600
16	16	1550
17	17	1450
18	18	1650
19	19	1350
20	20	2900

● Procedure

- Description: Tạo ra một thủ tục có tên là “Total Price Profit” in ra bảng chứa tên điện thoại, số lượng bán, giá nhập, giá bán và lợi nhuận thu được trên số lượng điện thoại đã bán của từng sản phẩm.
- Các cột trong bảng:
 - PhoneName : Tên điện thoại đã bán.
 - Quantity : Số lượng điện thoại đã bán.
 - Total Import Price : Tổng giá tiền nhập điện thoại trên số lượng điện thoại đã bán của từng sản phẩm.
 - Total Price : Tổng giá bán điện thoại trên số lượng điện thoại đã bán của từng sản phẩm.
 - Total Profit : Tổng lợi nhuận thu được trên số lượng điện thoại đã bán của từng sản phẩm
- Cột cuối cùng trong bảng sẽ hiện thị tổng, tổng giá nhập trên tất cả sản phẩm, giá bán trên tất cả sản phẩm , tổng lợi nhuận

• Procedure

```
create procedure Total_Price_Profit
as
begin
    with a as
    (
        SELECT ROW_NUMBER() OVER (ORDER BY p.PhoneName) as [ROW],
        p.PhoneName, t.Quantity, (t.ImportPrice * t.Quantity) as [TotalImportPrice], (t.Price * t.Quantity) as [TotalPrice],
        (t.Price - p.ImportPrice) * t.Quantity as [TotalProfit]
        from [Phone] as p
        join(
            select p1.PhoneID, p1.PhoneName, p1.ImportPrice,
            sum(p1.Quantity) as [Quantity of Order],
            sum(o.Quantity) as [Quantity], o.Price
            from OrderDetail as o
            left join Phone as p1 on o.PhoneID = p1.PhoneID
            group by p1.PhoneID, p1.PhoneName, p1.ImportPrice, o.Price
            ) as t on p.PhoneID = t.PhoneID
        )
        select * from a
        union
        SELECT (select count(*) + 1 from a), 'Total ', SUM(Quantity), Sum([TotalImportPrice]), SUM([TotalPrice]), SUM([TotalProfit]) from a
    end
```

• Function

Demo

```
exec Total_Price_Profit
```

	ROW	PhoneName	Quantity	TotalImportPrice	TotalPrice	TotalProfit
1	1	Huawei 14	2	2800	2900	100
2	2	iPhone 12 Pro	3	2250	2400	150
3	3	iPhone 12 Pro Max	4	3280	3480	200
4	4	iPhone 13 Pro	1	750	800	50
5	5	iPhone 13 Pro	1	700	750	50
6	6	iPhone 13 Pro Max	1	800	850	50
7	7	iPhone 14 Pro	1	1400	1450	50
8	8	iPhone 14 Pro Max	1	1500	1550	50
9	9	Oppo 12	2	1400	1500	100
10	10	SamSung S21	3	2250	2400	150
11	11	SamSung S21	2	1100	1200	100
12	12	SamSung S23	1	1300	1350	50
13	13	Xiaomi 11	2	1600	1700	100
14	14	Xiaomi 13	1	1600	1650	50
15	15	Total	25	22730	23980	1250

• Trigger

- Mục đích chính: Trình kích hoạt sẽ kiểm tra tổng số lượng của một sản phẩm nhất định trong kho nếu nó bằng hoặc lớn hơn số lượng của Khách hàng đối với sản phẩm đó.
- Mô tả: Khi chèn vào bảng [OrderDetails] sẽ tự động thiết lập
 - giá bán bằng đơn giá (giá nhập trong bảng Sản phẩm) + 50
 - Tổng số tiền phải trả cho đơn hàng đó = giá bán * số lượng
 - Tổng tiền lợi nhuận của đơn hàng đó = (giá bán - giá nhập) * số lượng order
- Sau khi chèn vào bảng [OrderDetails] số lượng mà sản phẩm được order ở trong bảng [Phone] sẽ bị giảm đi , số lượng trong bảng = số lượng ban đầu của bảng - số lượng order
- Tạo 2 hàng ở bảng OrderDetail
 - Tổng số tiền phải trả cho đơn hàng đó
 - Tổng tiền lợi nhuận của đơn hàng đó

```
--Create tạo thêm 2 cột trong OrderDetail  
  
alter table [dbo].[OrderDetail]  
ADD Total_of_Price int,  
    Total_of_Profit int;
```

• Trigger

• Code Trigger

```
create trigger UpdateOrderDetail on OrderDetail AFTER INSERT
as
begin
    TRANSACTION;
    --Khai báo biến
    DECLARE @PhoneID nvarchar(20);
    DECLARE @Quantity int;
    DECLARE @OrderID int;
    DECLARE @ImportPrice int;
    DECLARE @Price int;
    DECLARE @Available int;
    --Lấy giá trị từ bảng inserted
    SELECT @PhoneID = Inserted.PhoneID,
           @Quantity = Inserted.Quantity,
           @OrderID = Inserted.OrderID FROM Inserted;
    --Lấy đơn giá từ bảng Phone theo PhoneID
    SELECT @ImportPrice = ImportPrice FROM [Phone] WHERE PhoneID = @PhoneID ;
    --Thiết lập giá bán bằng đơn giá + 50
    SET @Price = @ImportPrice + 50;
    --Lấy số lượng sản phẩm từ bảng Phone theo PhoneID
    SELECT @Available = Quantity FROM [Phone] WHERE PhoneID = @PhoneID ;
    --Kiểm tra điều kiện
    IF @Quantity > @Available
        BEGIN
```

```
        --Kiểm tra điều kiện
        IF @Quantity > @Available
            BEGIN
                --In thông báo lỗi
                PRINT N'Hết hàng';
                --Hủy bỏ giao dịch
                ROLLBACK TRANSACTION;
            END
        ELSE
            BEGIN
                -- Cập nhập số lượng trong bảng Phone
                UPDATE Phone
                SET Quantity = Phone.Quantity - (select inserted.Quantity from inserted
                                                where inserted.PhoneID = Phone.PhoneID )
                from Phone join inserted on Phone.PhoneID = inserted.PhoneID
                --Cập nhật giá bán OrderDetail
                UPDATE OrderDetail
                SET Price = @Price WHERE OrderID = @OrderID AND PhoneID = @PhoneID;
                --Cập nhật tổng số tiền và lợi nhuận của đơn hàng trong bảng Order
                UPDATE [OrderDetail]
                SET [Total_of_Price] = (@Price * @Quantity),
                    [Total_of_Profit] = (@Price - @ImportPrice)*@Quantity
                WHERE OrderID = @OrderID AND PhoneID = @PhoneID;
                PRINT N'Dơn hàng đã đặt'
                --Xác nhận giao dịch
                COMMIT TRANSACTION;
            END
        END
```

• Transection

- Mục đích chính: Giao dịch này sẽ kiểm tra số lượng của sản phẩm đó trong bảng Điện thoại theo PhoneID của đơn hàng đó. Nếu đủ số lượng thì hóa đơn đó sẽ được nhập vào bảng 'OrderDetail', nếu không đủ số lượng thì hóa đơn đó không được nhập vào bảng 'OrderDetail' và in ra thông báo hết hàng.

```
begin tran
    declare @orderQuantity int ;
    set @orderQuantity = 10; -- nhập số lượng muốn mua
    declare @PhoneID nvarchar(20);
    set @PhoneID = 'IP18' ; -- nhập mã máy muốn mua
    insert into OrderDetail(OrderID, PhoneID, Quantity) values (21, @PhoneID, @orderQuantity)
    if @@ERROR <> 0
    begin
        print N'Lỗi tạo đơn hàng '
        rollback
    end
    else
    begin
        if (@orderQuantity <= (select Quantity from Phone where PhoneID = @PhoneID))
        begin
            if @@ERROR <> 0
            begin
                print 'Error on update Phone'
                rollback
            end
            else
            print 'Add new order Successfully!'
            end
        end
    end
commit
```

• Transection

DEMO

- Nhập khách hàng mua mới

```
INSERT into [dbo].[User] VALUES (N'HE888888', N'DEMO', N'DEMO', N'DEMO', N'DEMO@gmail.fpt.edu')
INSERT into [dbo].[Order] ([OrderID], [UserID], [OrderDate]) VALUES (21, N'HE888888', CAST(getdate() AS Date))
```

- Lúc đầu

	PhoneID	PhoneName	ManID	Quantity	ImportPrice	Ram	Chip	Year
10	IP16	iPhone 14 P...	IPHONE	9	1100	128GB	APPLE A23	2023
11	IP17	iPhone 14 P...	IPHONE	18	1300	256GB	APPLE A23	2023
12	IP18	iPhone 14 P...	IPHONE	20	1500	512GB	APPLE A23	2023
13	IP2	iPhone 12 P...	IPHONE	8	820	256GB	APPLE A20	2021
14	IP3	iPhone 13	IPHONE	8	820	256GB	APPLE A21	2022
15	IP4	iPhone 13 P...	IPHONE	10	700	128GB	APPLE A21	2022

	OrderID	PhoneID	Quantity	Price	Total_of_Price	Total_of_Profit
16	16	IP18	1	1550	NULL	NULL
17	17	IP15	1	1450	NULL	NULL
18	18	X4	1	1650	NULL	NULL
19	19	SS5	1	1350	NULL	NULL
20	20	H2	2	1450	2900	100

• Transection

DEMO

- Lúc sau : số lượng IP18 giảm còn 10.

	Results	Messages						
	PhoneID	PhoneName	ManID	Quantity	ImportPrice	Ram	Chip	Year
10	IP16	iPhone 14 P...	IPHONE	9	1100	128GB	APPLE A23	2023
11	IP17	iPhone 14 P...	IPHONE	18	1300	256GB	APPLE A23	2023
12	IP18	iPhone 14 P...	IPHONE	10	1500	512GB	APPLE A23	2023
13	IP2	iPhone 12 P...	IPHONE	8	820	256GB	APPLE A20	2021

- Order thêm đơn 21 thành công

	OrderID	PhoneID	Quantity	Price	Total_of_Price	Total_of_Profit
17	17	IP15	1	1450	NULL	NULL
18	18	X4	1	1650	NULL	NULL
19	19	SS5	1	1350	NULL	NULL
20	20	H2	2	1450	2900	100
21	21	IP18	10	1550	15500	500

• Transection

DEMO

- Trường hợp khách hàng order quá số lượng

```
begin tran
declare @orderQuantity int ;
set @orderQuantity = 30; -- nhập số lượng muốn mua
declare @PhoneID nvarchar(20);
set @PhoneID = 'IP17' ; -- nhập mã máy muốn mua
insert into OrderDetail(OrderID, PhoneID, Quantity) values (21, @PhoneID, @orderQuantity)
if @@ERROR <> 0
begin
    print N'Lỗi tạo đơn hàng '
    rollback
end
else
begin
    if (@orderQuantity <= (select Quantity from Phone where PhoneID = @PhoneID))
    begin
        if @@ERROR <> 0
        begin
            print 'Error on update Phone'
            rollback
        end
        else
        print 'Add new order Successfully!'
        end
    end
end
commit
```

Messages

Hết hàng

Msg 3609, Level 16, State 1, Line 8

The transaction ended in the trigger. The batch has been aborted.

