

## **1. Survey**

### **1.1 Reality**

Nowadays, information technology is developing more and more so that traditional commerce is gradually replaced by e-commerce. E-commerce is referred to as a form of buying and selling, communication in which buyers and sellers do not know each other and do not need to meet face to face, but all agreements and exchanges take place on the Internet through a special payment system such as bank credit card, e-wallet app such as airpay, momo,...

During the epidemic, people limit their leave of their homes and avoid contact in crowded places and it is extremely reasonable to have a website used to advertise and buy and sell. Moreover, it can also manage the number of buying and selling and bring better efficiency in management.

### **1.2 Purpose**

Based on the knowledge accumulated from Database Design, project will analyze and design an online shopping website in order to create a sale management system including the following functions:

- + ) Manage data of buyers
- + ) Manage data of sellers
- + ) Manage data of products
- + ) Manage system of product distribution

### **1.3 Range**

The project is implemented based on the scope of the Database Management Systems and the objectives outlined above

## **2. Describe**

### **2.1.Job Description**

Each customer can create one or more different orders, each order can have one or more different order details, Each product may appear on one or more detailed orders. A Vendor may also appear on 1 or more detailed orders. A category consists of one or more different products, Each product will be featured on one or more different ads, Each vendor will provide one or more different advertisements. A customer has a certain number of coins corresponding to the customer code, the customer can receive coins when sending feedback about the product and can use the coin to reduce the amount on the bill. Each customer can create one or more different feedbacks and each product can have one or more feedbacks. A vendor can create one or more different discount codes and each discount code can be applied to one or more different orders. In addition, each vendor can choose 1 or more different product distribution

## 2.2 Describing the entities

### 2.2.1. ENTITY: CUSTOMER

ENTITY: CUSTOMER		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
CUS_CODE : PK	CUSTOMER ID	VARCHAR(10)
FULL_NAME	CUSTOMER'S FULL NAME	NVARCHAR(255)
CUS_PHONE	CUSTOMER'S PHONE	VARCHAR(20)
CUS_ADDRESS	CUSTOMER'S ADDRESS	NVARCHAR(255)
DOB	CUSTOMER'S DATE OF BIRTH	SMALLDATETIME

### 2.2.2. ENTITY: VENDOR

ENTITY: VENDOR		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
V_CODE : PK	VENDOR CODE	VARCHAR(10)
V_NAME	VENDOR'S NAME	NVARCHAR(255)
V_ADDRESS	VENDOR'S ADDRESS	NVARCHAR(255)
COUNTRY	COUNTRY	NVARCHAR(255)
V_PHONE	VENDOR'S PHONE NUMBER	VARCHAR(20)
V_EMAIL	VENDOR'S EMAIL	NVARCHAR(255)

### 2.2.3.ENTITY: CATEGORIES

ENTITY: CATEGORIES		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
CATEGORIES_CODE : PK	CATEGORIES' CODE	VARCHAR(10)
CAT_DESCRIPT	CATEGORIES' DESCRIPTION	NVARCHAR(255)

## 2.2.4.ENTITY: PRODUCT

ENTITY: PRODUCT		
ATtribution NAME	EXPLAINING	DATA TYPE
P_CODE: PK	PRODUCT CODE	VARCHAR(10)
CATEGORY_CODE: FK	PRODUCT BELONGS TO WHICH CATEGORIES	VARCHAR(10)
P_DESCRIPT	PRODUCT DESCRIPTION	NVARCHAR(255)
P_QOH	QUANTITIES OF THE PRODUCT	INT
P_PRICE	PRICE OF THE PRODUCT	MONEY
P_RATING	RATING OF THE PRODUCT	FLOAT
STT	STATUS OF THE PRODUCT	NVARCHAR(255)

## 2.2.5. ENTITY: ORDERS

ENTITY: ORDERS		
ATtribution NAME	EXPLAINING	DATA TYPE
ORDER_CODE : PK	ORDER CODE	VARCHAR(10)
CUS_CODE :FK	CUSTOMER CODE	VARCHAR(10)
ORD_DATE	DATE OF THE ORDER	SMALLDATETIME
DUE_DATE	DUE DATE OF THE ORDER	SMALLDATETIME
SHIP_COST	THE COST OF SHIPPING	MONEY
DELIVERY_TYPE	TYPE OF THE DELIVERY	NVARCHAR(255)
PAYMENT_TYPE	TYPE OF PAYMENT	NVARCHAR(255)
CUS_NOTE	CUSTOMER NOTE	NVARCHAR(255)
STT_ORDER	STATUS OF THE ORDER	NVARCHAR(255)
COIN_USED	COIN CUSTOMER USED IN THE ORDER	INT

### 2.2.6. ENTITY: ORDER\_DETAILS

ENTITY:ORDER_DETAILS		
ATtribution NAME	EXPLAINING	DATA TYPE
ORDER_DETAIL_NUM : PK	CODE OF THE ORDERDETAIL	VARCHAR(10)
ORDER_CODE: PK	CODE OF THE ORDER	VARCHAR(10)
P_CODE: FK	CODE OF THE PRODUCT	VARCHAR(10)
V_CODE: FK	CODE OF THE VENDOR	VARCHAR(10)
UNITS	UNITS OF PRODUCT IN A ORDER DETAILS	INT
PRICES	PRICES OF THE PRODUCT IN A ORDER DETAILS	MONEY

### 2.2.7. ENTITY: ADS

ENTITY: ADS		
ATtribution NAME	EXPLAINING	DATA TYPE
ADS_CODE : PK	CODE OF THE ADVERTISING	VARCHAR(10)
P_CODE : FK	CODE OF THE PRODUCT	VARCHAR(10)
V_CODE : FK	CODE OF THE VENDOR	VARCHAR(10)
ADS_FULLNAME	NAME OF THE ADVERTISING	NVARCHAR(255)
ADS_TYPE	TYPE OF ADVERTISING	NVARCHAR(255)

### 2.2.8. ENTITY: FEEDBACKS

ENTITY: FEEDBACKS		
ATtribution NAME	EXPLAINING	DATA TYPE
FB_ID : PK	ID OF THE FEEDBACKS PROVIDED BY CUSTOMER	INT
CUS_CODE : FK	CODE OF CUSTOMER	VARCHAR(10)
P_CODE : FK	CODE OF PRODUCT	VARCHAR(10)
RATING	CUSTOMER'S RATING	INT
COMMENT	CUSTOMER'S COMMENT	NVARCHAR(255)

### 2.2.9. ENTITY: EMPLOYEES

ENTITY: EMPLOYEES		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
EMP_CODE : PK	CODE OF EMPLOYEE	VARCHAR(10)
EMP_NAME	NAME OF EMPLOYEE	NVARCHAR(255)
EMP_PHONE	PHONE NUMBER OF EMPLOYEE	VARCHAR(20)
EMP_DESCRIPT	DESCRIPTION OF EMPLOYEE	NVARCHAR(255)

### 2.2.10. ENTITY: CUS\_SP

ENTITY: CUS_SP		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
SP_CODE : PK	CODE OF SUPPORT	VARCHAR(10)
CUS_CODE : FK	CODE OF CUSTOMER	VARCHAR(10)
EMP_CODE : FK	CODE OF EMPLOYEE	VARCHAR(10)
SP_DESCRIPT	DESCRIPTION OF SUPPORT	NVARCHAR(255)
SP_DATE	DATE THAT CUSTOMER ASK FOR SUPPORTING	SMALLDATETIME

### 2.2.11. ENTITY: SHIP

ENTITY: SHIP		
ATTRIBUTION NAME	EXPLAINING	DATA TYPE
SHIP_CODE : PK	CODE OF SHIP	VARCHAR(10)
V_CODE : FK	CODE OF VENDOR	VARCHAR(10)
DELIVERY_NAME	NAME OF SHIP'S DELIVERY	NVARCHAR(255)
CUS_NOTE	CUSTOMER NOTE ABOUT SHIP	NVARCHAR(255)
V_NOTE	VENDOR NOTE ABOUT SHIP	NVARCHAR(255)

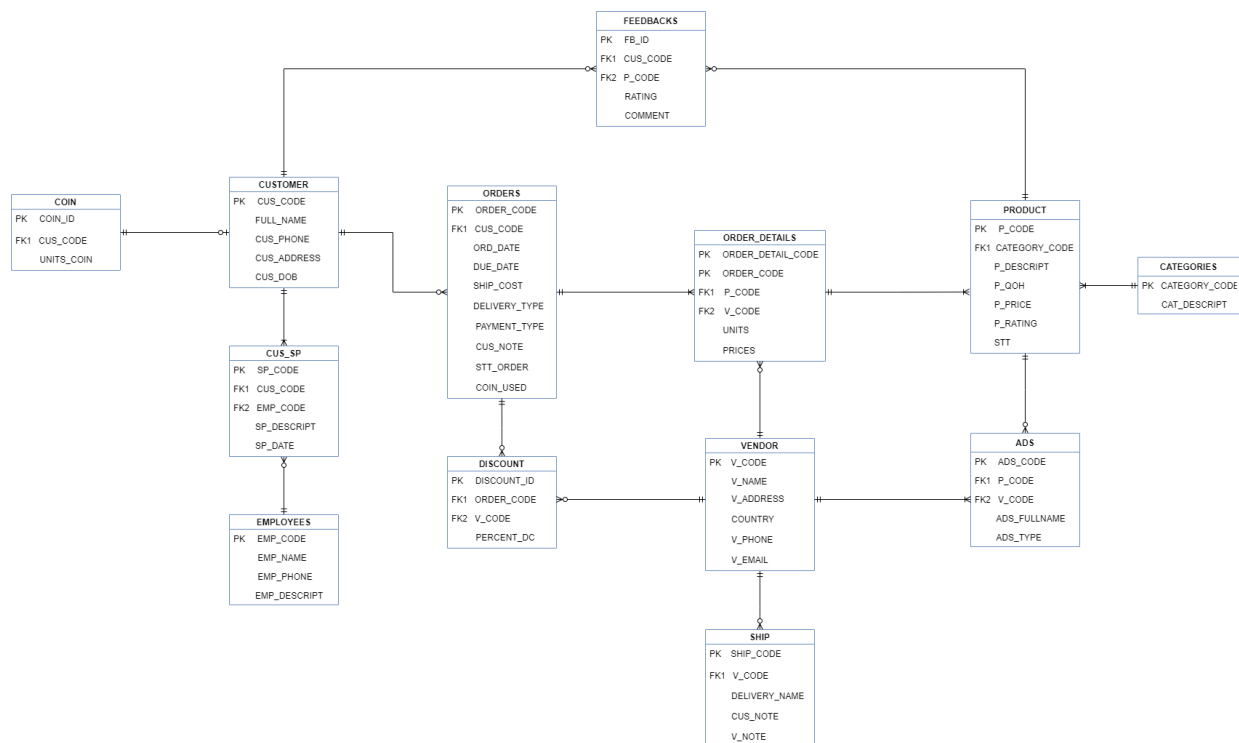
### 2.2.12. ENTITY: COIN

ENTITY: COIN		
ATtribution NAME	EXPLAINING	DATA TYPE
COIN_ID : PK	ID OF COIN	VARCHAR(10)
CUS_CODE : FK	CODE OF CUSTOMER	VARCHAR(10)
UNITS_COIN	NUMBER OF COINS THAT CUSTOMER HOLD	INT

### 2.2.13. ENTITY: DISCOUNT

ENTITY: DISCOUNT		
ATtribution NAME	EXPLAINING	DATA TYPE
DISCOUNT_ID : PK	ID OF THE DISCOUNT	VARCHAR(10)
ORDER_CODE : FK	CODE OF THE ORDER	VARCHAR(10)
V_CODE : FK	CODE OF THE VENDOR	VARCHAR(10)
PERCENT_DC	PERCENT OF DISCOUNT	FLOAT

## 2.3: ERD Model



## 2.4 Convert ERD model to relational model

2.3.1.CUSTOMER(CUS\_CODE, FULL\_NAME, CUS\_PHONE, CUS\_ADDRESS, DOB)

2.3.2.VENDOR(V\_CODE, V\_NAME, V\_ADDRESS, COUNTRY, V\_PHONE, V\_EMAIL)

2.3.3.CATEGORIES(CATEGORY\_CODE, CAT\_DESCRIPT)

2.3.4.PRODUCT(P\_CODE, P\_DESCRIPT, CATEGORY\_CODE, P\_QOH, P\_PRICE, P\_RATING, STT)

2.3.5.ORDERS(ORDER\_CODE, CUS\_CODE, ORD\_DATE, DUE\_DATE, SHIP\_COST, DELIVERY\_TYPE, PAYMENT\_TYPE, CUS\_NOTE, STT\_ORDER, COIN\_USED)

2.3.6.ORDER\_DETAILS(ORDER\_DETAIL\_NUM, ORDER\_CODE, P\_CODE, V\_CODE, UNITS, PRICES)

2.3.7.ADS(ADS\_CODE, P\_CODE, V\_CODE, ADS\_FULLNAME, ADS\_TYPE)

2.3.8.FEEDBACKS(FB\_ID, CUS\_CODE, P\_CODE, RATING, COMMENT)

2.3.9.EMPLOYEES(EMP\_CODE, EMP\_NAME, EMP\_PHONE, EMP\_DESCRIPT)

2.3.10.CUS\_SP(SP\_CODE, CUS\_CODE, EMP\_CODE, SP\_DESCRIPT, SP\_DATE)

2.3.11.SHIP(SHIP\_CODE, DELIVERY\_NAME, V\_CODE, CUS\_NOTE, V\_NOTE)

2.3.12.COIN(COIN\_ID, CUS\_CODE, UNITS\_COIN)

2.3.13.DISCOUNT(DISCOUNT\_ID, ORDER\_CODE, V\_CODE, PERCENT\_DC)

### 3. Query, View, Trigger and Procedure

#### 3.1.QUERY

-- 3.1.1. Finding out the product provided by Vietnamese's vendor

```
SELECT DISTINCT PRODUCT.P_DESCRIPT, VENDOR.COUNTRY  
  
FROM VENDOR JOIN ADS ON VENDOR.V_CODE = ADS.V_CODE JOIN PRODUCT ON ADS.P_CODE =  
PRODUCT.P_CODE  
  
WHERE VENDOR.COUNTRY = 'Vietnam'
```

	P_DESCRIPT	COUNTRY
1	Ấm siêu tốc Lock & Lock	Vietnam
2	Áo khoác jeans trơn	Vietnam
3	Bạch tuộc nhồi bông 2 mặt cảm xúc	Vietnam
4	Bàn phím Bluetooth	Vietnam
5	Cốc sứ cổ điển	Vietnam
6	Coca-ina	Vietnam
7	Giày Oxford đế cao	Vietnam
8	Gối ngủ chữ U kê cổ	Vietnam
9	iMac Pro	Vietnam
10	Iphone 12 PRO MAX	Vietnam
11	Ipod3	Vietnam

=> Finding out that products that Vietnamese's vendors haven't provided yet, there by, improving vendor's supply services, providing a wider variety of products

--3.1.2. Find the number of customers with service problems in 2021 sorted by month in ascending order

```
SELECT CUSTOMER.CUS_CODE, CUS_SP.SP_DESCRIPT, CUS_SP.SP_DATE  
  
FROM CUSTOMER JOIN CUS_SP ON CUSTOMER.CUS_CODE = CUS_SP.CUS_CODE  
  
WHERE YEAR(CUS_SP.SP_DATE) = 2021  
  
ORDER BY MONTH(CUS_SP.SP_DATE) ASC
```

	CUS_CODE	SP_DESCRIPT	SP_DATE
1	KH22	Lỗi Giao Dịch	2021-03-08 00:00:00
2	KH26	Lỗi Thanh Toán	2021-03-08 00:00:00
3	KH20	Lỗi Truy Cập	2021-03-08 00:00:00
4	KH40	Lỗi Truy Cập	2021-03-08 00:00:00
5	KH4	Lỗi Giao Dịch	2021-03-08 00:00:00
6	KH40	Lỗi Giao Dịch	2021-03-08 00:00:00
7	KH23	Lỗi Thanh Toán	2021-03-08 00:00:00
8	KH24	Lỗi Thanh Toán	2021-03-08 00:00:00



=> The goal is to find out which month customers ask the most about service problems, then improve the system to suit the needs of customers (For example, on March 8, 2021, there are many customers having problems with the service. payment problems and access errors)

### --3.1.3. Find item in BOOK'S category only

```
SELECT PRODUCT.P_DESCRIPT
```

```
FROM CATEGORIES JOIN PRODUCT ON CATEGORIES.CATEGORY_CODE =  
PRODUCT.CATEGORY_CODE
```

```
WHERE CATEGORIES.CAT_DESCRIPT = N'Sách'
```

	P_DESCRIPT
1	A brief history of time
2	A song of ice and fire
3	Set 30 tờ giấy ghi chú
4	Bút bi
5	Bút chì
6	Set 52 miếng dán graffiti màu hồng

=> The target is looking for any books that are not available on the web to add more

### --3.1.4. Find products that have not been purchased by customers

```
SELECT PRODUCT.P_CODE, PRODUCT.P_DESCRIPT,PRODUCT.P_PRICE
```

```
FROM PRODUCT
```

```
WHERE PRODUCT.P_CODE not in (
```

```
SELECT ORDER_DETAILS.P_CODE FROM ORDER_DETAILS )
```

	P_CODE	P_DESCRIPT	P_PRICE
1	P22	A brief history of time	250000.00
2	P26	CROCS Crocband	500000.00
3	P42	Quần đùi họa tiết dễ thương	110000.00
4	P43	Tắm nước mini Xiaomi Enpuly M6	500000.00
5	P44	Áo mưa phản quang vải dù	120000.00
6	P45	Áo nhung tắm lót lông	650000.00
7	P46	Xe mô hình Lamborghini LB770	2500000.00
8	P47	Nước Hoa Hồng Klairs Supple Preparation 180ml	250000.00
9	P48	Ốp điện thoại Iphone	22000.00
10	P49	Hộp khẩu trang y tế	60000.00
11	P50	Giày Oxford đế cao	350000.00

=>Find out the reason why the customer has not bought and fix it

### --3.1.5. Average rating of the product

```
SELECT AVG(FEEDBACKS.RATING) AS AVERAGE, PRODUCT.P_DESCRIPTOR
FROM PRODUCT JOIN FEEDBACKS ON PRODUCT.P_CODE = FEEDBACKS.P_CODE
GROUP BY PRODUCT.P_DESCRIPTOR
ORDER BY AVERAGE DESC
```

	AVERAGE	P_DESCRIPTOR
1	5	Mì Indomie
2	5	Tay cầm PS5
3	4	Hộp bánh Đài Loan 35 loại
4	4	Mô hình M4A4 - CS:GO
5	4	Súng đồ chơi
6	4	Tạ tay 15kg
7	4	A brief history of time
8	4	Chip máy tính
9	4	Iphone 12 PRO MAX
10	3	Ipod3
11	3	Kính mắt thời trang Hàn

=> Knowing which products have the highest average rating, evaluating the vendor providing good products, will put more product information on the web and limit the information of products with low ratings.

### --3.1.6. Top selling products

```
SELECT SUM(ORDER_DETAILS.UNITS) AS TOTALUNITS, PRODUCT.P_DESCRIPTOR
FROM PRODUCT JOIN ORDER_DETAILS ON PRODUCT.P_CODE = ORDER_DETAILS.P_CODE
GROUP BY PRODUCT.P_DESCRIPTOR
ORDER BY TOTALUNITS DESC
```

	TOTALUNITS	P_DESCRIPTOR
1	52	Kính thực tế ảo
2	45	Tạ tay 15kg
3	43	Mô hình M4A4 - CS:GO
4	36	Mô hình đồ chơi
5	33	Gfuel - The Juice
6	31	Mũ rơm
7	27	Mắt kính
8	26	Set 21 mask Chando
9	26	iMac Pro
10	25	Chuột fuhlen
11	25	Ipod3

=> Predict customer buying trends

### --3.1.7. TOP most active service employee

```
SELECT DISTINCT TOP 10 COUNT(EMPLOYEES.EMP_CODE) AS TOTAL,EMPLOYEES.EMP_NAME
FROM EMPLOYEES JOIN CUS_SP ON EMPLOYEES.EMP_CODE = CUS_SP.EMP_CODE
GROUP BY EMPLOYEES.EMP_NAME
ORDER BY TOTAL DESC
```

	TOTAL	EMP_NAME
1	10	NguyenVanA
2	8	NguyenVanB
3	8	TranVanB
4	7	NguyenVanC
5	7	PhanThiB
6	6	PhanThiC
7	6	TranBaA
8	5	CaoBaG
9	5	NguyenThiA
10	5	CaoThiG

=> Aim to find the most active service employees to increase salary

### --3.1.8. The most chosen shipping company

```
SELECT DISTINCT TOP 3 COUNT(SHIP.DELIVERY_NAME) AS SL,SHIP.DELIVERY_NAME
FROM SHIP
GROUP BY SHIP.DELIVERY_NAME
ORDER BY SL DESC
```

	SL	DELIVERY_NAME
1	8	Grab Express
2	7	Fed-Ex
3	5	J&T Express

=> coverage of shipping carriers

### --3.1.9.What do customers born in 2001 usually buy?

```
SELECT DISTINCT PRODUCT.P_CODE,PRODUCT.P_DESCRIPT,PRODUCT.P_PRICE,CUSTOMER.DOB
FROM CUSTOMER JOIN ORDERS ON CUSTOMER.CUS_CODE = ORDERS.CUS_CODE JOIN
ORDER_DETAILS ON ORDERS.ORDER_CODE = ORDER_DETAILS.ORDER_CODE JOIN PRODUCT ON
ORDER_DETAILS.P_CODE = PRODUCT.P_CODE
WHERE YEAR(CUSTOMER.DOB) = 2001
ORDER BY PRODUCT.P_PRICE DESC
```

	P_CODE	P_DESCRIPT	P_PRICE	DOB
1	P15	iMac Pro	25000000.00	2001-08-16 00:00:00
2	P15	iMac Pro	25000000.00	2001-10-01 00:00:00
3	P19	Nintendo Switch	8900000.00	2001-02-16 00:00:00
4	P19	Nintendo Switch	8900000.00	2001-02-20 00:00:00
5	P19	Nintendo Switch	8900000.00	2001-03-12 00:00:00
6	P32	Camera Xiaomi	8500000.00	2001-03-12 00:00:00
7	P28	Diluc Costume - Genshin Impact	2000000.00	2001-10-01 00:00:00
8	P6	Ipod3	2000000.00	2001-02-10 00:00:00
9	P6	Ipod3	2000000.00	2001-02-11 00:00:00
10	P6	Ipod3	2000000.00	2001-08-16 00:00:00
11	P6	Ipod3	2000000.00	2001-12-23 00:00:00

=>Can predict what customers born in 2001 or buy and suggest for new customers born in 2001

--3.1.10.The vendor offers the most discount

```

SELECT TOP 10 COUNT(VENDOR.V_CODE) AS TOTAL,VENDOR.V_CODE
FROM VENDOR JOIN DISCOUNT ON VENDOR.V_CODE = DISCOUNT.V_CODE
GROUP BY VENDOR.V_CODE
ORDER BY TOTAL DESC

```

	TOTAL	V_CODE
1	4	V10
2	3	V13
3	2	V12
4	2	V2
5	2	V14
6	2	V4
7	2	V8
8	2	V9
9	1	V5
10	1	V3

=> The goal is to find the vendor with the most discounts, suggest vendor

## 3.2. VIEW

---3.2.1.VIEW OF THE TOP 10 BEST SELLING PRODUCTS

```

CREATE VIEW TOP_PRODUCT AS
    SELECT TOP(10) P_DESCRIPT, UNITS
    FROM ORDER_DETAILS AS O

```

```

FULL JOIN PRODUCT AS P
ON O.P_CODE = P.P_CODE
ORDER BY UNITS DESC;

```

```

SELECT * FROM TOP_PRODUCT;

```

	P_DESCRIPT	UNITS
1	Diluc Costume - Genshin Impact	18
2	Mũ rơm	16
3	Mũ rơm	15
4	Mô hình M4A4 - CS:GO	14
5	Converse c? cao	13
6	God of War - 2018	12
7	Thần đồng đất việt - tập 222	10
8	Peppa the Pig Plushie	10
9	Mì Indomie	10
10	Bộ 3 hộp tẩy trang Silcot 82 miếng/hộp	10

=>Top 10 best-selling products => buying trends' customers.

### ---3.2.2.VIEW OF THE TOTAL PROFIT MADE

```

CREATE VIEW VIEW_DOANHTHU AS

```

```

    SELECT TOP(100) V.V_CODE, V.V_NAME, SUM(UNITS*PRICES) AS TOTAL
    FROM VENDOR V
    FULL JOIN ORDER_DETAILS O ON V.V_CODE = O.V_CODE
    FULL JOIN ORDERS ON O.ORDER_CODE = ORDERS.ORDER_CODE
    WHERE NOT STT_ORDER = N'Dơn đã hủy'
    GROUP BY V.V_CODE, V.V_NAME
    ORDER BY TOTAL DESC;

```

```

SELECT *FROM VIEW_DOANHTHU;

```

	V_CODE	V_NAME	TOTAL
1	V18	Nmplol	803840000
2	V16	Dnututt	678986000
3	V22	Phạm Trang	648320000
4	V4	Nguyễn Đức Trung	439450000
5	V13	Vũ Minh Tuấn	368175000
6	V9	Trần Thị Hồng	277920000
7	V15	Asmon	241645000
8	V10	Đinh Phương Thảo	240715000
9	V1	Nguyễn Nam Anh	214850000
10	V17	Jankos	212270000
11	V14	Cat Anh	139700000
12	V24	Lê Thị Thu Thảo	134590000
13	V19	Charlie	97991000
14	V23	Đặng Mai	81724000
15	V21	Trần Thảo Nguyên	74200000
16	V20	Nguyễn Thị Nguyệt	66420000
17	V2	Hồ Ánh Dương	53780000
18	V11	Lê Thị Thường	50600000
19	V25	Lê Thu Hương	30636000
20	V6	Hồ Đức Duy	25150000
21	V5	Trịnh Việt Hưng	24870000
22	V3	Nguyễn Bằng Hiệp	11800000
23	V7	Trịnh Khắc Dương	9840000
24	V8	Dokutan	9000000
25	V12	Trần Tuấn Quang	6200000

**=>From the revenue of vendors, we can know which suppliers operate effectively on the online sales platform.**

### **---3.2.3.VIEW OF MONTHLY PROFIT**

CREATE VIEW DOANHTHUTHANG AS

```

    SELECT V.V_CODE, V.V_NAME, SUM(UNITS*PRICES) AS TOTAL,
MONTH(DUE_DATE) AS MONTHS
    FROM VENDOR V
    FULL JOIN ORDER_DETAILS O ON V.V_CODE = O.V_CODE
    FULL JOIN ORDERS ON O.ORDER_CODE = ORDERS.ORDER_CODE
    WHERE NOT STT_ORDER = N'Dơn đã hủy'
    GROUP BY V.V_CODE, V.V_NAME, MONTH(DUE_DATE);

```

SELECT \*FROM DOANHTHUTHANG;

	V_CODE	V_NAME	TOTAL	MONTHS
1	V1	Nguyễn Nam Anh	200000	1
2	V1	Nguyễn Nam Anh	250000	5
3	V1	Nguyễn Nam Anh	214400000	6
4	V10	Đinh Phương Thảo	2400000	3
5	V10	Đinh Phương Thảo	3000000	5
6	V10	Đinh Phương Thảo	235315000	6
7	V11	Lê Thị Thường	2250000	3
8	V11	Lê Thị Thường	5250000	5
9	V11	Lê Thị Thường	43100000	6
10	V12	Trần Tuấn Quang	2000000	3
11	V12	Trần Tuấn Quang	1500000	5
12	V12	Trần Tuấn Quang	2700000	6
13	V13	Vũ Minh Tuấn	60000	3
14	V13	Vũ Minh Tuấn	1500000	5
15	V13	Vũ Minh Tuấn	366615000	6
16	V14	Cat Anh	200000	3
17	V14	Cat Anh	1200000	5
18	V14	Cat Anh	138300000	6
19	V15	Asmon	100000000	4
20	V15	Asmon	75000000	5

**=>View the vendor's monthly revenue => which month the shop has the highest revenue**  
**Identify which month suppliers usually have large sales => have better methods of promoting purchases**

#### **---3.2.4. VIEW OF PRODUCTS SOLD**

CREATE VIEW SANPHAM AS

```

    SELECT P.P_CODE, P_P_DESCRIPT, SUM(UNITS) AS QUANTITY,
    MONTH(DUE_DATE) AS MONTHS
    FROM PRODUCT P
    FULL JOIN ORDER_DETAILS O ON P.P_CODE = O.P_CODE
    FULL JOIN ORDERS OS ON O.ORDER_CODE = OS.ORDER_CODE
    WHERE NOT STT_ORDER = N'Đơn đã hủy'
    GROUP BY P.P_CODE, MONTH(DUE_DATE), P_P_DESCRIPT;
```

SELECT \*FROM SANPHAM;

	P_CODE	P_DESCRIPT	QUANTITY	MONTHS
1	P1	Súng đồ chơi	4	1
2	P1	Súng đồ chơi	5	5
3	P10	Mô hình đồ chơi	8	3
4	P10	Mô hình đồ chơi	28	6
5	P11	Chuột fuhlen	15	3
6	P11	Chuột fuhlen	10	6
7	P12	Cây máy tính	4	3
8	P12	Cây máy tính	3	5
9	P12	Cây máy tính	15	6
10	P13	isdeD	3	3
11	P13	isdeD	15	6
12	P14	Mắt kính	1	3
13	P14	Mắt kính	6	5
14	P14	Mắt kính	20	6
15	P15	iMac Pro	4	4
16	P15	iMac Pro	4	5
17	P15	iMac Pro	18	6
18	P16	Tạ tay 15kg	45	6
19	P17	Gfuel - The Jui...	33	6
20	P18	Tay cầm PS5	20	6
21	P19	Nintendo Switch	25	6
22	P2	Iphone 12 PR...	1	1

=>View how much the product can be sold each month.

### ---3.2.5VIEW OF CUSTOMERS SPENDING

CREATE VIEW CHITIEUKHACHHANG AS

```

    SELECT TOP(100) C.CUS_CODE, C.FULL_NAME, SUM(UNITS*PRICES) AS
TOTAL_SPENDING
    FROM CUSTOMER C
    FULL JOIN ORDERS OS ON C.CUS_CODE = OS.CUS_CODE
    FULL JOIN ORDER_DETAILS O ON OS.ORDER_CODE = O.ORDER_CODE
    WHERE STT_ORDER = N'Đã giao'
    GROUP BY C.CUS_CODE, C.FULL_NAME
    ORDER BY TOTAL_SPENDING DESC;
```

SELECT \*FROM CHITIEUKHACHHANG;



	CUS_CODE	FULL_NAME	TOTAL_SPENDING
1	KH35	Do Minh	63750000
2	KH19	Cao Tu Uyen	60830000
3	KH34	Le Van Anh	30600000
4	KH5	Tran Minh Hoang	25000000
5	KH40	Chu Anh Duc	19160000
6	KH6	Cao Ba Thien	16000000
7	KH37	Le Phuong Anh	13500000
8	KH26	Ha Minh Duc	12000000
9	KH7	Tran Huyen Trang	8400000
10	KH17	Nguyen Thi Hanh	4500000
11	KH20	Doan Thi Bich N...	4400000
12	KH24	Nguyen Van Bao	4000000
13	KH12	Nguyen Minh Hieu	2000000
14	KH39	Nguyen Ngoc T...	1920000
15	KH9	Le Thi Tu Van	1800000
16	KH30	Tran Thi Tu	1680000
17	KH32	Nguyen Thi Nga	1500000
18	KH11	Le Tien Bang	1200000
19	KH10	Ha Long Giang	1200000
20	KH22	Bui Minh Nguyet	500000
21	KH36	Tran Thuc Hien	450000
22	KH28	Le Thi Ha	250000
23	KH15	Ma Duy Son	200000
24	KH1	Cao Ba Hoang	150000
25	KH21	Trinh Thi Hoai	60000

View how much customers spend on online shopping

### ---3.2.6.Create a view of customer spending by month

CREATE VIEW CHITIEUTHANG AS

```

    SELECT C.CUS_CODE, C.FULL_NAME, SUM(UNITS*PRICES) AS
TOTAL_SPENDING, MONTH(DUE_DATE) AS MONTHS
    FROM CUSTOMER C
    FULL JOIN ORDERS OS ON C.CUS_CODE = OS.CUS_CODE
    FULL JOIN ORDER_DETAILS O ON OS.ORDER_CODE = O.ORDER_CODE
    WHERE STT_ORDER = N'Đã giao'
    GROUP BY C.CUS_CODE, C.FULL_NAME, MONTH(DUE_DATE)

```

SELECT \*FROM CHITIEUTHANG;

	CUS_CODE	FULL_NAME	TOTAL_SPENDING	MONTHS
1	KH10	Ha Long Giang	1200000	2
2	KH11	Le Tien Bang	1200000	2
3	KH6	Cao Ba Thien	16000000	2
4	KH12	Nguyen Minh Hieu	2000000	3
5	KH20	Doan Thi Bich Ngoc	1200000	3
6	KH21	Trinh Thi Hoai	60000	3
7	KH24	Nguyen Van Bao	4000000	4
8	KH28	Le Thi Ha	250000	5
9	KH30	Tran Thi Tu	1200000	5
10	KH32	Nguyen Thi Nga	1500000	5
11	KH35	Do Minh	5250000	5
12	KH1	Cao Ba Hoang	150000	6
13	KH15	Ma Duy Son	200000	6
14	KH17	Nguyen Thi Hanh	4500000	6
15	KH19	Cao Tu Uyen	60830000	6
16	KH20	Doan Thi Bich Ngoc	3200000	6
17	KH22	Bui Minh Nguyet	500000	6
18	KH26	Ha Minh Duc	12000000	6
19	KH30	Tran Thi Tu	480000	6
20	KH34	Le Van Anh	30600000	6
21	KH35	Do Minh	58500000	6
22	KH36	Tran Thuc Hien	450000	6
23	KH37	Le Phuong Anh	13500000	6
24	KH39	Nguyen Ngoc Thai	1920000	6
25	KH40	Chu Anh Duc	19160000	6
26	KH5	Tran Minh Hoang	25000000	6
27	KH7	Tran Huyen Trang	8400000	6
28	KH9	Le Thi Tu Van	1800000	6

View how much customers spend on online shopping each month => assess their willingness to pay for buying online => improve platform more suitable

### 3.3. TRIGGER

**3.3.1.TRIGGER DETAILS:** When an order is placed, the number of goods in stock is reduced and while the order is canceled, the number of goods is increased

**--UPDATE STOCK AFTER ORDER OR UPDATE**

```
CREATE OR ALTER TRIGGER UPDATE_QUANTITY
ON ORDER_DETAILS
AFTER INSERT
AS
DECLARE @pcode VARCHAR(10);
DECLARE @unit INT;

SELECT @pcode = I.P_CODE FROM INSERTED AS I;
```

```
SELECT @unit = I.UNITES FROM INSERTED AS I;
```

```
BEGIN
```

```
    UPDATE PRODUCT
```

```
    SET P_QOH = P_QOH - @unit
```

```
    WHERE PRODUCT.P_CODE = @pcode
```

```
END;
```

```
INSERT INTO ORDER_DETAILS VALUES ('OD1310', 'O18', 'P1','V1',10, 120000.00)
```

```
INSERT INTO ORDER_DETAILS VALUES ('OD1320', 'O17', 'P10','V2',5, 120000.00)
```

```
INSERT INTO ORDER_DETAILS VALUES ('OD1330', 'O16', 'P11','V5',3, 120000.00)
```

```
INSERT INTO ORDER_DETAILS VALUES ('OD1340', 'O15', 'P12','V3',1, 120000.00)
```

```
INSERT INTO ORDER_DETAILS VALUES ('OD1350', 'O12', 'P15','V9',5, 120000.00)
```

```
SELECT * FROM PRODUCT
```

	P_CODE	P_DESCRIPTOR	CATEGORY_CODE	P_QOH	P_PRICE	P_RATING	STT
1	P1	Súng đồ chơi	CA1	170	50000,00	4	Còn Hàng
2	P10	Mô hình đồ chơi	CA1	1255	300000,00	4,1	Còn Hàng
3	P11	Chuột fuhlen	CA4	97	150000,00	4,1	Còn Hàng
4	P12	Cây máy tính	CA4	19	500000,00	3,6	Còn Hàng
5	P13	isdeD	CA6	100	20000,00	4,4	Còn Hàng
6	P14	Mắt kính	CA5	1000	200000,00	3,1	Còn Hàng
7	P15	iMac Pro	CA4	13	25000000,00	4,1	Còn Hàng

## -- UPDATE THE UNITS IN PRODUCT WHEN THE ORDER DETAILS IS UPDATED

```
CREATE OR ALTER TRIGGER UPDATE_ORDERS
```

```
ON ORDER_DETAILS
```

```
AFTER UPDATE
```

```
AS
```

```
BEGIN
```

```
    UPDATE PRODUCT
```

```
    SET P_QOH = P_QOH - (SELECT UNITS FROM INSERTED WHERE P_CODE =PRODUCT.P_CODE)
```

```
        + (SELECT UNITS FROM DELETED WHERE P_CODE = PRODUCT.P_CODE)
```

```
FROM PRODUCT
```

```
JOIN DELETED ON PRODUCT.P_CODE = DELETED.P_CODE
```

```
END;
```

UPDATE ORDER\_DETAILS SET UNITS = 5 WHERE ORDER\_DETAIL\_NUM = 'OD1310'

UPDATE ORDER\_DETAILS SET UNITS = 15 WHERE ORDER\_DETAIL\_NUM = 'OD1320'

SELECT \* FROM PRODUCT

	P_CODE	P_DESCRIPTOR	CATEGORY_CODE	P_QOH	P_PRICE	P_RATING	STT
1	P1	Súng đồ chơi	CA1	175	50000,00	4	Còn Hàng
2	P10	Mô hình đồ chơi	CA1	1255	300000,00	4,1	Còn Hàng
3	P11	Chuột fuhlen	CA4	97	150000,00	4,1	Còn Hàng
4	P12	Cây máy tính	CA4	19	500000,00	3,6	Còn Hàng
5	P13	isdeD	CA6	100	20000,00	4,4	Còn Hàng

### -- UPDATE THE UNITS IN PRODUCT WHEN THE ORDER IS CANCELLED

CREATE OR ALTER TRIGGER DELETE\_ORDERS

ON ORDER\_DETAILS

AFTER DELETE

AS

DECLARE @p\_code VARCHAR(10);

DECLARE @unit INT;

SELECT @p\_code = D.P\_CODE FROM DELETED AS D;

SELECT @unit = D.UNITS FROM DELETED AS D;

BEGIN

    UPDATE PRODUCT

    SET P\_QOH = P\_QOH + @unit

    WHERE PRODUCT.P\_CODE = @p\_code

END;

DELETE FROM ORDER\_DETAILS WHERE ORDER\_DETAIL\_NUM = 'OD1310'

DELETE FROM ORDER\_DETAILS WHERE ORDER\_DETAIL\_NUM = 'OD1320'

	P_CODE	P_DESCRIPTOR	CATEGORY_CODE	P_QOH	P_PRICE	P_RATING	STT
1	P1	Súng đồ chơi	CA1	180	50000,00	4	Còn Hàng
2	P10	Mô hình đồ chơi	CA1	1260	300000,00	4,1	Còn Hàng
3	P11	Chuột fuhlen	CA4	97	150000,00	4,1	Còn Hàng
4	P12	Cây máy tính	CA4	19	500000,00	3,6	Còn Hàng
5	P13	isdeD	CA6	100	20000,00	4,4	Còn Hàng

### 3.3.2.TRIGGER FEEDBACKS: UPDATE THE RATING FOR PRODUCT(S) WHEN FEEDBACK IS SUBMITTED

```

CREATE OR ALTER TRIGGER UPDATE_RATE
ON FEEDBACKS
AFTER INSERT
AS
DECLARE @pcode VARCHAR(10);
DECLARE @rating INT;

SELECT @pcode = I.P_CODE FROM INSERTED AS I;
SELECT @rating = I.RATING FROM INSERTED AS I;

BEGIN
    UPDATE PRODUCT
        SET P_RATING = ROUND((@rating + P_RATING)/2,1)
        WHERE PRODUCT.P_CODE = @pcode
END;

INSERT INTO FEEDBACKS VALUES ('1011','KH12','P1','5','Excellent')
INSERT INTO FEEDBACKS VALUES ('1211','KH2','P10','4','Good')
INSERT INTO FEEDBACKS VALUES ('1217','KH36','P11','3','Neutral')
INSERT INTO FEEDBACKS VALUES ('1217','KH36','P12','2','Bad')
INSERT INTO FEEDBACKS VALUES ('1217','KH36','P13','3','Neutral')

SELECT * FROM PRODUCT

```

	P_CODE	P_DESCRIPTOR	CATEGORY_CODE	P_QOH	P_PRICE	P_RATING	STT
1	P1	Súng đồ chơi	CA1	180	50000,00	4,5	Còn Hàng
2	P10	Mô hình đồ chơi	CA1	1260	300000,00	4	Còn Hàng
3	P11	Chuột fuhlen	CA4	97	150000,00	3,5	Còn Hàng
4	P12	Cây máy tính	CA4	19	500000,00	2,8	Còn Hàng
5	P13	isdeD	CA6	100	20000,00	3,7	Còn Hàng
6	P14	Mắt kính	CA5	1000	200000,00	3,1	Còn Hàng

### 3.3.3.TRIGGER ORDERS: UPDATE THE NUMBER OF COINS WHEN THE CUSTOMER USES COINS TO PAY IN THE ORDER AND REFUND THE COINS WHEN THE ORDER IS CANCELED

--INSERT THE ORDER TABLE WHEN CUSTOMER(S) USE COINS TO GET A DISCOUNT ON THEIR ORDER(S)

```

CREATE OR ALTER TRIGGER UPDATE_COIN

```

```
ON ORDERS
AFTER INSERT
AS
```

```
DECLARE @cus_code VARCHAR(10);
DECLARE @coin_used INT;
```

```
SELECT @cus_code = I.CUS_CODE FROM INSERTED AS I;
SELECT @coin_used = I.COIN_USED FROM INSERTED AS I;
```

```
BEGIN
IF @coin_used <= (SELECT UNITS_COIN FROM COIN WHERE COIN.CUS_CODE =
@cus_code)
    UPDATE COIN
    SET UNITS_COIN = UNITS_COIN - @coin_used
    WHERE COIN.CUS_CODE = @cus_code
END;
```

```
SELECT * FROM COIN
```

	COIN_ID	CUS_CODE	UNITS_COIN
1	C1	KH1	10000
2	C10	KH10	6000
3	C11	KH11	2100
4	C12	KH12	13700
5	C13	KH13	2000
6	C14	KH14	4600
7	C15	KH15	800
8	C16	KH16	50000
9	C17	KH17	33400

```
INSERT INTO ORDERS VALUES ('O1112', 'KH1', '2021-06-17', '2021-06-20', 45000.00, N'Vận chuyển nhanh', 'Cash', N'Tránh xa tầm tay trẻ nhỏ', N'Đang giao',1000)
```

```
INSERT INTO ORDERS VALUES ('O1113', 'KH10', '2021-06-17', '2021-06-20', 20000.00, N'Vận chuyển tiết kiệm', 'Credit Card', N'Tránh xa tầm tay trẻ nhỏ', N'Đang giao',5000)
```

```
INSERT INTO ORDERS VALUES ('O1114', 'KH11', '2021-06-18', '2021-06-20', 30000.00, N'Vận chuyển nhanh', 'Credit Card', N'Tránh xa tầm tay trẻ nhỏ', N'Đang giao',5000)
```

```
INSERT INTO ORDERS VALUES ('O1115', 'KH12', '2021-06-19', '2021-06-20', 15000.00, N'Vận chuyển tiết kiệm', 'Cash', N'Tránh xa tầm tay trẻ nhỏ', N'Đã giao',100)
```

```
INSERT INTO ORDERS VALUES ('O1116', 'KH13', '2021-06-07', '2021-06-20', 20000.00, N'Vận chuyển tiết kiệm', 'Credit Card', N'Tránh xa tầm tay trẻ nhỏ', N'Đang giao',700)
```

```
INSERT INTO ORDERS VALUES ('O1117', 'KH14', '2021-06-18', '2021-06-20', 15000.00, N'Vận
chuyển tiết kiệm', 'Credit Card', N'Tránh xa tầm tay trẻ nhỏ', N'Đang giao',1000)
SELECT * FROM COIN
```

	COIN_ID	CUS_CODE	UNITS_COIN
1	C1	KH1	9000
2	C10	KH10	1000
3	C11	KH11	2100
4	C12	KH12	13600
5	C13	KH13	1300
6	C14	KH14	3600

#### **--UPDATE THE COIN TABLE WHEN AN ORDER IS CANCELLED**

```
CREATE OR ALTER TRIGGER _COIN
```

```
ON ORDERS
```

```
AFTER DELETE
```

```
AS
```

```
DECLARE @cus_code VARCHAR(10);
```

```
DECLARE @coin_used INT;
```

```
SELECT @cus_code = I.CUS_CODE FROM DELETED AS I;
```

```
SELECT @coin_used = I.COIN_USED FROM DELETED AS I;
```

```
BEGIN
```

```
    UPDATE COIN
```

```
    SET UNITS_COIN = UNITS_COIN + @coin_used
```

```
    WHERE COIN.CUS_CODE = @cus_code
```

```
END;
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1112'
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1113'
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1114'
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1115'
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1116'
```

```
DELETE FROM ORDERS WHERE ORDER_CODE = 'O1117'
```

```
SELECT * FROM COIN
```

	COIN_ID	CUS_CODE	UNITS_COIN
1	C1	KH1	10000
2	C10	KH10	6000
3	C11	KH11	2100
4	C12	KH12	13700
5	C13	KH13	2000
6	C14	KH14	4600
7	C15	KH15	800
8	C16	KH16	50000
9	C17	KH17	33400

### 3.3.4. TRIGGER CUSTOMER: WHEN A CUSTOMER CHANGES INFORMATION OR ADDS NEW INFORMATION, THAT INFORMATION WILL BE UPDATED AND SAVED IN THE AUDIT TABLE

--CREATE AN AUDIT TABLE TO TRACK CHANGES IN THE TABLE CUSTOMER

```
CREATE TABLE AUDIT_CUS (
CUS_CODE VARCHAR(10),
FIELD NVARCHAR(255),
OLD_DATA VARCHAR(50),
NEW_DATA VARCHAR(50),
ACTION_TIME SMALLDATETIME );
```

--WHEN ADDING DATA OF A NEW CUSTOMER, ALL THE NEW AND OLD DATA WILL BE INSERTED INTO THE AUDIT TABLE ALONG WITH THE CORRESPONDING ID

```
CREATE OR ALTER TRIGGER UPDATE_INF
ON CUSTOMER
AFTER UPDATE,INSERT
AS
DECLARE @new_cuscode VARCHAR(10);
DECLARE @new_fullname NVARCHAR(255);
DECLARE @new_cus_phone VARCHAR(20);
DECLARE @new_address NVARCHAR(255);
DECLARE @new_dob SMALLDATETIME;

DECLARE @old_cuscode VARCHAR(10);
DECLARE @old_fullname NVARCHAR(255);
DECLARE @old_cus_phone VARCHAR(20);
```



```
DECLARE @old_address NVARCHAR(255);
```

```
DECLARE @old_dob SMALLDATETIME;
```

```
SELECT @new_cuscode = I.CUS_CODE FROM inserted AS I;
```

```
SELECT @new_fullname = I.FULL_NAME FROM inserted AS I;
```

```
SELECT @new_cus_phone = I.CUS_PHONE FROM inserted AS I;
```

```
SELECT @new_address = I.CUS_ADDRESS FROM inserted AS I;
```

```
SELECT @new_dob = I.DOB FROM inserted AS I;
```

```
SELECT @old_cuscode = I.CUS_CODE FROM deleted AS I;
```

```
SELECT @old_fullname = I.FULL_NAME FROM deleted AS I;
```

```
SELECT @old_cus_phone = I.CUS_PHONE FROM deleted AS I;
```

```
SELECT @old_address = I.CUS_ADDRESS FROM deleted AS I;
```

```
SELECT @old_dob = I.DOB FROM deleted AS I;
```

```
BEGIN
```

```
IF @new_cuscode <> @old_cuscode OR (@old_cuscode IS NULL AND @new_cuscode IS NOT NULL) OR (@old_cuscode IS NOT NULL AND @new_cuscode IS NULL)
```

```
INSERT INTO AUDIT_CUS VALUES
```

```
(@new_cuscode,'CUS_CODE',@old_cuscode,@new_cuscode, SYSDATETIME())
```

```
IF @new_fullname <> @old_fullname OR (@old_fullname IS NULL AND @new_fullname IS NOT NULL) OR (@old_fullname IS NOT NULL AND @new_fullname IS NULL)
```

```
INSERT INTO AUDIT_CUS VALUES
```

```
(@new_cuscode,'CUS_NAME',@old_fullname,@new_fullname, SYSDATETIME())
```

```
IF @new_cus_phone <> @old_cus_phone OR (@old_cus_phone IS NULL AND
```

```
@new_cus_phone IS NOT NULL) OR (@old_cus_phone IS NOT NULL AND
```

```
@new_cus_phone IS NULL)
```

```
INSERT INTO AUDIT_CUS VALUES
```

```
(@new_cuscode,'CUS_PHONE',@old_cus_phone,@new_cus_phone, SYSDATETIME())
```

```
IF @new_address <> @old_address OR (@old_address IS NULL AND @new_address IS NOT NULL) OR (@old_address IS NOT NULL AND @new_address IS NULL)
```

```
INSERT INTO AUDIT_CUS VALUES
```

```
(@new_cuscode,'CUS_ADDRESS',@old_address,@new_address, SYSDATETIME())
```

```

IF @new_dob <> @old_dob OR (@old_dob IS NULL AND @new_dob IS NOT NULL) OR
(@old_dob IS NOT NULL AND @new_dob IS NULL)
INSERT INTO AUDIT_CUS VALUES (@new_cuscode,'CUS_DOB',@old_dob,@new_dob,
SYSDATETIME())
END;

```

```

INSERT INTO CUSTOMER VALUES ('KH1212', 'Nguyen Thi Hoa', '0123458745','Hai Ba
Trung,Ha Noi','2001-05-22')

```

```

SELECT * FROM AUDIT_CUS

```

	CUS_CODE	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	KH1212	CUS_CODE	NULL	KH1212	2021-05-31 21:31:00
2	KH1212	CUS_NAME	NULL	Nguyen Thi Hoa	2021-05-31 21:31:00
3	KH1212	CUS_PHONE	NULL	0123458745	2021-05-31 21:31:00
4	KH1212	CUS_ADDRESS	NULL	HaiBaTrung,Ha Noi	2021-05-31 21:31:00
5	KH1212	CUS_DOB	NULL	May 22 2001 12:00AM	2021-05-31 21:31:00

```

UPDATE CUSTOMER SET FULL_NAME = 'Tran Thanh Giang' WHERE CUS_CODE =
'KH1212'

```

```

SELECT * FROM AUDIT_CUS

```

	CUS_CODE	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	KH1212	CUS_CODE	NULL	KH1212	2021-05-31 21:31:00
2	KH1212	CUS_NAME	NULL	Nguyen Thi Hoa	2021-05-31 21:31:00
3	KH1212	CUS_PHONE	NULL	0123458745	2021-05-31 21:31:00
4	KH1212	CUS_ADDRESS	NULL	HaiBaTrung,Ha Noi	2021-05-31 21:31:00
5	KH1212	CUS_DOB	NULL	May 22 2001 12:00AM	2021-05-31 21:31:00
6	KH1212	CUS_NAME	Nguyen Thi Hoa	Tran Thanh Giang	2021-05-31 21:32:00

**--WHEN DELETING A CUSTOMER, ALL CORRESPONDING COLUMNS WILL RETURN TO NULL AND BE STORED IN THE AUDIT TABLE**

```

CREATE OR ALTER TRIGGER DELETE_NOTE
ON CUSTOMER
AFTER DELETE
AS

```

```

DECLARE @cuscode VARCHAR(10);
DECLARE @fullname NVARCHAR(255);

```

```

DECLARE @cus_phone VARCHAR(20);
DECLARE @address NVARCHAR(255);
DECLARE @dob SMALLDATETIME;

SELECT @cuscode = I.CUS_CODE FROM deleted AS I;
SELECT @fullname = I.FULL_NAME FROM deleted AS I;
SELECT @cus_phone = I.CUS_PHONE FROM deleted AS I;
SELECT @address = I.CUS_ADDRESS FROM deleted AS I;
SELECT @dob = I.DOB FROM deleted AS I;

BEGIN
IF @cuscode IS NOT NULL
INSERT INTO AUDIT_CUS VALUES(@cuscode,'CUS_CODE',
@cus_phone,NULL,SYSDATETIME())
IF @fullname IS NOT NULL
INSERT INTO AUDIT_CUS VALUES(@cuscode,'CUS_NAME',
@fullname,NULL,SYSDATETIME())
IF @cus_phone IS NOT NULL
INSERT INTO AUDIT_CUS VALUES(@cuscode,'CUS_PHONE',
@cus_phone,NULL,SYSDATETIME())
IF @address IS NOT NULL
INSERT INTO AUDIT_CUS VALUES(@cuscode,'CUS_ADDRESS',
@address,NULL,SYSDATETIME())
IF @dob IS NOT NULL
INSERT INTO AUDIT_CUS VALUES(@cuscode,'CUS_DOB', @dob,NULL,SYSDATETIME())
END;

DELETE FROM CUSTOMER WHERE CUS_CODE = 'KH1212'
SELECT * FROM AUDIT_CUS

```

KH1212	CUS_CODE	0123458745	NULL	2021-06-01 13:32:00
KH1212	CUS_NAME	Tran Thanh Giang	NULL	2021-06-01 13:32:00
KH1212	CUS_PHONE	0123458745	NULL	2021-06-01 13:32:00
KH1212	CUS_ADDRESS	HaiBaTrung,Ha ...	NULL	2021-06-01 13:32:00
KH1212	CUS DOB	Mav 22 2001 12:...	NULL	2021-06-01 13:32:00

## 3.4.Procedures

### 3.4.1.Shows the feedback of products based on their product code (p\_code)

#### CREATE OR ALTER PROCEDURE FEEDBACK

@PROD VARCHAR(10)

AS

SELECT \* FROM FEEDBACKS WHERE @PROD = P\_CODE

GO

EXEC FEEDBACK P1

	FB_ID	CUS_CODE	P_CODE	RATING	COMMENT
1	1	KH1	P1	4	Good
2	4	KH3	P1	4	Good
3	31	KH5	P1	4	Good
4	36	KH35	P1	4	Good
5	39	KH25	P1	4	Good
6	41	KH18	P1	5	Excellent

**Purpose:** Allows customers to read the comments and ratings of certain product(s) in order to assist in the purchasing process

### 3.4.2.Show products in selected category within certain price range

#### CREATE OR ALTER PROCEDURE PRICE

@MIN MONEY, @MAX MONEY, @CA VARCHAR(10)

AS

SELECT \* FROM PRODUCT

WHERE P\_PRICE >= @MIN AND P\_PRICE <= @MAX

AND CATEGORY\_CODE = @CA ORDER BY P\_PRICE

GO

EXEC PRICE 20000, 100000000, CA2

	P_CODE	P_DESCRIPT	CATEGORY_CODE	P_QOH	P_PRICE	STT
1	P48	Ốp điện thoại Iphone	CA2	65	22000.00	Còn Hàng
2	P62	Sạc dự phòng Xiaomi Gen 3	CA2	290	300000.00	Còn Hàng
3	P6	Ipod3	CA2	100	2000000.00	Còn Hàng
4	P2	Iphone 12 PRO MAX	CA2	30	25000000.00	Còn Hàng

**Purpose: Allow customers to easily find product(s) within certain price range based on their category**

### 3.4.3 Show the status of your order

#### CREATE OR ALTER PROCEDURE ORDER\_STAT

@OD\_CODE VARCHAR(10)

AS

SELECT STT\_ORDER FROM ORDERS WHERE ORDER\_CODE = @OD\_CODE

GO

EXEC ORDER\_STAT O1

	STT_ORDER
1	Chờ lấy hàng

**Purpose: Allow customers to check the status of their order**

### 3.4.4. Show the details of the vendor(s) based on the product they provide

#### CREATE OR ALTER PROCEDURE VENDOR\_DETAILS

@P\_DESCRIPT NVARCHAR(255)

AS

SELECT V\_CODE, V\_NAME, V\_ADDRESS, COUNTRY, V\_PHONE, V\_EMAIL FROM VENDOR

FULL JOIN ORDER\_DETAILS ON VENDOR.V\_CODE = ORDER\_DETAILS.V\_CODE

FULL JOIN PRODUCT ON ORDER\_DETAILS.P\_CODE = PRODUCT.P\_CODE

WHERE PRODUCT.P\_DESCRIPT = @P\_DESCRIPT

GO

EXEC VENDOR\_DETAILS 'iMac Pro'

	V_NAME	V_ADDRESS	COUNTRY	V_PHONE	V_EMAIL
1	Asmon	Austin	America	05551337	otk@gmail.com
2	Drututt	Warsaw	Poland	0665556948	Dr.ututt@gmail.com
3	Jankos	Warsaw	Poland	0455558226	marcin@gmail.com
4	Lê Thu Huong	Hà Nam	Vietnam	03958777	lth@gmail.com
5	Nmplol	Austin	America	NULL	nmpol@gmail.com

**Purpose:** Allow the customers to find information on the vendors that provided their product(s) -> promotes good vendors/blacklist bad ones.

#### 3.4.5.Show the product(s) the vendor provides

##### CREATE OR ALTER PROCEDURE PRODUCT\_DETAILS

@V\_CODE VARCHAR(10)

AS

SELECT DISTINCT PRODUCT.P\_CODE, P\_DESCRIPT , CATEGORY\_CODE FROM  
VENDOR

FULL JOIN ORDER\_DETAILS ON VENDOR.V\_CODE = ORDER\_DETAILS.V\_CODE

FULL JOIN PRODUCT ON ORDER\_DETAILS.P\_CODE = PRODUCT.P\_CODE

WHERE VENDOR.V\_CODE = @V\_CODE

GO

EXEC PRODUCT\_DETAILS V3

	P_CODE	P_DESCRIPT	CATEGORY_CODE
1	P17	Gfuel - The Juice	CA7
2	P3	Mi Indomie	CA3

**Purpose:** Allow the customers to find the product(s) provided by certain vendor(s) -> helps customer(s) easily find product from a vendor they like/ helps vendor(s) expand their horizon.

#### 3.4.6.Show the top products sold in selected month

##### CREATE OR ALTER PROCEDURE TOP\_PROD\_MONTH

@MONTH DATETIME

AS

SELECT TOP 5 PRODUCT.P\_CODE, P\_DESCRIPT, SUM(UNITS) AS TOTAL FROM  
ORDER\_DETAILS

```

FULL JOIN ORDERS ON ORDER_DETAILS.ORDER_CODE = ORDERS.ORDER_CODE
FULL JOIN PRODUCT ON ORDER_DETAILS.P_CODE = PRODUCT.P_CODE
WHERE MONTH(ORDERS.ORD_DATE) = @MONTH
GROUP BY UNITS, PRODUCT.P_CODE, P_DESCRIPT
ORDER BY TOTAL DESC
GO
EXEC TOP_PROD_MONTH 5

```

	P_CODE	P_DESCRIPT	TOTAL
1	P41	Set 21 mask Chando	18
2	P38	Hộp bánh Đài Loan 35 loại	16
3	P3	Mì Indomie	10
4	P40	Bộ 3 hộp tẩy trang Silcot 82 miếng/hộp	10
5	P10	Mô hình đồ chơi	9

**Purpose:** Helps shopee identify the trend of purchase on the website -> Assist in the placement of ads, discount system, notify customer(s) of the quality product(s)

### 3.4.7.Show top products sold based on month and category

#### CREATE OR ALTER PROCEDURE TOP\_PROD\_CA

@MONTH DATETIME, @CA VARCHAR(10)

AS

```

SELECT TOP 5 PRODUCT.P_CODE, P_DESCRIPT, SUM(UNITS) AS TOTAL FROM
ORDER_DETAILS

```

```

FULL JOIN ORDERS ON ORDER_DETAILS.ORDER_CODE = ORDERS.ORDER_CODE

```

```

FULL JOIN PRODUCT ON ORDER_DETAILS.P_CODE = PRODUCT.P_CODE

```

```

WHERE MONTH(ORDERS.ORD_DATE) = @MONTH AND CATEGORY_CODE = @CA

```

```

GROUP BY PRODUCT.P_CODE, P_DESCRIPT

```

```

ORDER BY TOTAL DESC

```

GO

```

EXEC TOP_PROD_CA 5, CA8

```

	P_CODE	P_DESCRIPT	TOTAL
1	P20	Kính thực tế ảo	22
2	P18	Tay cầm PS5	3

**Purpose: Assist in the placement of ads, discount system, notifying customer(s) of the quality of product(s) in certain categories**

### 3.4.8 Show the top selling categories of the month

#### CREATE OR ALTER PROCEDURE TOP\_CA

@MONTH DATETIME

AS

SELECT TOP 5 CAT\_DESCRIPT, SUM(UNITS) AS TOTAL FROM ORDER\_DETAILS

FULL JOIN ORDERS ON ORDER\_DETAILS.ORDER\_CODE = ORDERS.ORDER\_CODE

FULL JOIN PRODUCT ON ORDER\_DETAILS.P\_CODE = PRODUCT.P\_CODE

FULL JOIN CATEGORIES ON PRODUCT.CATEGORY\_CODE =  
CATEGORIES.CATEGORY\_CODE

WHERE MONTH(ORDERS.ORD\_DATE) = @MONTH

GROUP BY CAT\_DESCRIPT

ORDER BY TOTAL DESC

GO

EXEC TOP\_CA 5

	CAT_DESCRIPT	TOTAL
1	Sức khỏe và Sắc đẹp	47
2	Đồ chơi	36
3	Đồ ăn	33
4	Thiết bị điện tử	25
5	Máy tính & Laptop	24

**Purpose: Assist in the placement of ads, discount system, notifying website of the weight of certain categories in certain months**

## 4.CONCLUSION

Through the analysis and design of shopee's database of operation, we have gained a more thorough understanding of online shopping as well as solving the unsolved problems that plague traditional commerce. Clients are able to purchase their product(s) faster, more



convenient thus saving time and resources. At the same time, it allows the vendors to compile the number of popular product(s), feedbacks or the number of surplus product(s) leading to rank the most popular product(s).

**⇒ In today's growing society, it is very appropriate and common for "sales management" as well as other management activities to be born.**