

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

*****  *****



BÁO CÁO CUỐI KÌ
IT3280
THỰC HÀNH CƠ SỞ DỮ LIỆU

Nhóm sinh viên thực hiện: **Nhóm 8**

Giảng viên hướng dẫn:

1. Hoàng Hương Giang – 20225619
2. Nguyễn Thu Trang – 20225938
3. Nguyễn Thị Hạ – 20225622
4. Bành Thị Thảo Trang – 20225768

MỤC LỤC

Phần I: Giới thiệu tổng quan.....	3
1. Đặt vấn đề và giới thiệu bài toán.....	3
2. Định nghĩa ‘Cơ sở dữ liệu giao vận’	3
3. Phân tích tổng quan chức năng hệ thống.....	3
3.1. Kịch bản đặt ra:	3
3.2. Các đối tượng người dùng:	3
Phần II: Thiết kế cơ sở dữ liệu.....	5
1.Thiết kế mô hình dữ liệu ER	5
2. Sơ đồ quan hệ	5
Phần III: Triển khai Cơ sở dữ liệu trên MySQL	7
1. Thiết kế mô hình vật lý	7
1.1. Bảng thực thể	7
1.2. Bảng thực thể liên kết	9
1.3. Trigger.....	10
1.4. Function	16
1.5. View	23
1.6. Procedure	24
2. Triển khai truy vấn.....	25

Phần I: Giới thiệu tổng quan

1. Đặt vấn đề và giới thiệu bài toán

Hiện nay việc vận chuyển chở hàng hóa đang có sự phát triển đột biến trên thị trường nhiều nước. Đặc biệt là trong bối cảnh toàn cầu hóa, sự phát triển mạnh mẽ của các sàn thương mại điện tử như shopee, lazada, tiki,..... đã tạo ra một cuộc cách mạng lớn đối với ngành giao vận. Giao vận giờ đây không chỉ là vấn đề vĩ mô như vận chuyển hàng từ tỉnh này qua tỉnh khác, hay từ nước này qua nước khác,...Mà người dùng có thể chỉ ngồi ở nhà là có thể trải nghiệm các dịch vụ giao vận. Cuộc cách mạng này còn sinh ra rất nhiều các nhân tố khác để đáp ứng yêu cầu của người dùng có thể nói như shipper, các kho hàng,... Vì vậy việc quản lý giao vận đặt ra thách thức lớn đối với các hãng vận chuyển hiện nay để đảm bảo sự hài lòng của khách hàng.

Vậy nên nhóm chúng em xin được đề xuất ‘Cơ sở dữ liệu giao vận’ để giải quyết vấn đề nan giải này.

2. Định nghĩa ‘Cơ sở dữ liệu giao vận’

Cơ sở dữ liệu quản lý quá trình giao hàng từ người gửi thông qua quá trình trung chuyển giữa các kho và shipper đến người nhận.

3. Phân tích tổng quan chức năng hệ thống

3.1. Kịch bản đặt ra:

Người gửi gửi hàng đi, hàng sẽ được vận chuyển qua các kho, các kho ghi lại ngày đơn hàng đến nếu đơn hàng chưa ra khỏi kho thì ngày xuất kho sẽ là không được thêm (Null). Sau khi đến kho đích, đơn hàng sẽ được đưa cho shipper, shipper sẽ vận chuyển đến tay người dùng. Kết quả vận chuyển sẽ được shipper cập nhật.

3.2. Các đối tượng người dùng:

- Người gửi: Mỗi người gửi đều có tài khoản có thể làm các thao tác cơ bản sau
 - i. Các thao tác với đơn hàng: Tạo đơn hàng, xem thông tin đơn hàng,....
 - ii. Shipper: xem thông tin shipper, xem thông tin đơn hàng,...
 - iii. Quản lý doanh thu: xem báo cáo tổng hợp tài chính,....
 - iv. Quản lý khách hàng(người nhận): tạo địa chỉ người nhận, xem địa chỉ người gửi,.....

- Người nhận: Có thể có tài khoản hoặc không cần tài khoản cần tài khoản có thể làm các thao tác cơ bản sau:
 - i. Nhập mã đơn hàng để xem thông tin đơn hàng, thông tin người gửi, thông tin shipper,...
- Shipper: mỗi shipper cần 1 tài khoản và có thể làm các thao tác sau:
 - ii. Đơn hàng: xem thông tin đơn hàng
 - iii. Người nhận: xem thông tin về người nhận
 - iv. Người gửi: xem thông tin về người gửi Người dùng:
- Người quản lí:
 - i. Xem được các yêu cầu nâng cao về xu hướng, đặc tính của các đơn hàng như thống kê tỷ lệ các đơn hàng vận chuyển thành công, shipper nào đạt chuẩn,.....
 - ii. Quản lý, xem, cập nhật được tất cả các thực thể và liên kết trong hệ thống.

Phần II: Thiết kế cơ sở dữ liệu

1. Thiết kế mô hình dữ liệu ER

- Các thực thể:
Product(OrderID, Payer, PickupCity, PickupDistrict, PickupWard, PickupAddress, RecipientName, DeliveryCity, DeliveryDistrict, DeliveryWard, DeliveryAddress, PhoneRecipient, ServiceID)
Service (ServiceID, ServiceName, MaxDistance, Price)
Warehouse (WarehouseID, WareName, Address, Ward, District, City)
OrderDetails(ItemID, ItemName, SurchargeID, Weight, itemprice)
Surcharge (SurchargeID, SurchargeName, Price)
AccUser (UserID, LastName, MiddleName, FirstName, Birthday, Gender, Phone, City, District, Ware, Address)
Shipper (EmployeeID, LastName, MiddleName, FirstName, Gender, Birthday, Phone, HomeTown)
- Liên kết giữa các thực thể:
Product $\xleftrightarrow[n]{1}$ Service
Product $\xleftrightarrow[n]{n}$ Warehouse
 Importexport(OrderID, WarehouseID, InboundDate, OutboundDate)
Product $\xleftrightarrow[n]{1}$ AccUser
 Ordercreate(OrderID, GiverID, ReciverID, OrderDate)
Product $\xleftrightarrow[n]{1}$ Shipper
 send(EmployeeID, OrderID, ReceiptDate, OrderDate EstimatedDate, ActualDate, SendStatus)
Product $\xleftrightarrow[n]{1}$ OrderDetails
OrderDetails $\xleftrightarrow[1]{1}$ Surcharge

2. Sơ đồ quan hệ

Phần III: Triển khai Cơ sở dữ liệu trên MySQL

1. Thiết kế mô hình vật lý

1.1. Bảng thực thể

Bảng AccUser: Lưu thông tin người dùng

Field	Type	Null	Key	Default	Extra
UserID	char(5)	NO	PRI	NULL	
LastName	varchar(10)	YES		NULL	
MiddleName	varchar(10)	YES		NULL	
FirstName	varchar(10)	NO		NULL	
Birthday	date	YES		NULL	
Gender	char(3)	YES		NULL	
Phone	varchar(10)	NO	UNI	NULL	
City	varchar(30)	YES		NULL	
District	varchar(45)	YES		NULL	
Ware	varchar(45)	YES		NULL	
Address	varchar(45)	YES		NULL	

Bảng Product: Lưu thông tin đơn hàng

Field	Type	Null	Key	Default	Extra
OrderID	char(5)	NO	PRI	NULL	
Payer	varchar(15)	NO		NULL	
PickupCity	varchar(50)	NO		NULL	
PickupDistrict	varchar(30)	YES		NULL	
PickupWard	varchar(30)	YES		NULL	
PickupAddress	varchar(45)	NO		NULL	
RecipientName	varchar(30)	NO		NULL	
PhoneRecipient	varchar(15)	NO		NULL	
DeliveryCity	varchar(50)	YES		NULL	
DeliveryDistrict	varchar(45)	YES		NULL	
DeliveryWard	varchar(45)	YES		NULL	
DeliveryAddress	varchar(45)	NO		NULL	
ServiceID	char(5)	YES	MUL	NULL	

Bảng Shipper: Lưu thông tin nhân viên Shipper

Field	Type	Null	Key	Default	Extra
EmployeeID	char(5)	NO	PRI	NULL	
LastName	varchar(10)	YES		NULL	
MiddleName	varchar(10)	YES		NULL	
FirstName	varchar(10)	NO		NULL	
Gender	char(6)	YES		NULL	
Birthday	date	YES		NULL	
Phone	varchar(10)	NO	UNI	NULL	
HomeTown	varchar(45)	NO		NULL	

Bảng Warehouse: Lưu thông tin kho

Field	Type	Null	Key	Default	Extra
WarehouseID	char(4)	NO	PRI	NULL	
WareName	varchar(30)	NO		NULL	
City	varchar(40)	NO		NULL	
District	varchar(30)	YES		NULL	
Ward	varchar(30)	YES		NULL	
Address	varchar(30)	YES		NULL	

Bảng Service: Lưu thông tin dịch vụ

Field	Type	Null	Key	Default	Extra
ServiceID	char(5)	NO	PRI	NULL	
ServiceName	varchar(20)	YES		NULL	
Price	decimal(12,3)	NO		NULL	
MaxDistance	char(6)	YES		NULL	

Bảng Surcharge: Lưu thông tin phân loại hàng hóa

Field	Type	Null	Key	Default	Extra
SurchargeID	char(5)	NO	PRI	NULL	
SurchargeName	varchar(20)	YES		NULL	
Price	decimal(5,2)	YES		NULL	

Bảng OrderDetails: Lưu thông tin chi tiết đơn hàng

Field	Type	Null	Key	Default	Extra
ItemID	char(10)	NO	PRI	NULL	
OrderID	char(5)	NO	MUL	NULL	
ItemName	varchar(30)	NO		NULL	
SurchargeID	varchar(20)	YES	MUL	NULL	
Weight	decimal(18,2)	NO		NULL	
itemprice	decimal(20,3)	YES		NULL	

1.2. Bảng thực thể liên kết

Bảng OrderCreate: Lưu thông tin liên quan đến việc người dùng tạo đơn

Field	Type	Null	Key	Default	Extra
OrderID	char(5)	NO	PRI	NULL	
GiverID	char(5)	NO	PRI	NULL	
ReceiverID	char(5)	YES		NULL	
OrderDate	date	NO		NULL	

Bảng Send: Lưu thông tin liên quan đến việc shipper giao đơn hàng

Field	Type	Null	Key	Default	Extra
EmployeeID	char(5)	NO	PRI	NULL	
OrderID	char(5)	NO	PRI	NULL	
ReceiptDate	date	YES		NULL	
EstimatedDate	date	YES		NULL	
ActualDate	date	YES		NULL	
SendStatus	varchar(30)	YES		NULL	

Bảng ImportExport: Lưu thông tin liên quan đến việc xuất nhập kho của đơn hàng

Field	Type	Null	Key	Default	Extra
OrderID	char(5)	NO	PRI	NULL	
WarehouseID	char(5)	NO	PRI	NULL	
InboundDate	date	NO		NULL	
OutboundDate	date	YES		NULL	

1.3. Trigger

Tạo trigger để lưu trữ thông tin cập nhật vào bảng log sau mỗi lần cập nhật bảng Warehouse

```
CREATE TABLE WarehouseLog (  
    LogID INT AUTO_INCREMENT PRIMARY KEY,  
    WarehouseID CHAR(4),  
    OldWareName VARCHAR(30),  
    NewWareName VARCHAR(30),  
    OldCity VARCHAR(40),  
    NewCity VARCHAR(40),  
    OldDistrict VARCHAR(30),  
    NewDistrict VARCHAR(30),  
    OldWard VARCHAR(30),  
    NewWard VARCHAR(30),  
    OldAddress VARCHAR(30),  
    NewAddress VARCHAR(30),  
    ChangeTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DELIMITER //
```

```
CREATE TRIGGER after_warehouse_update  
AFTER UPDATE ON Warehouse  
FOR EACH ROW  
BEGIN
```

```
INSERT INTO WarehouseLog (WarehouseID, OldWareName,  
NewWareName, OldCity, NewCity, OldDistrict, NewDistrict, OldWard,  
NewWard, OldAddress, NewAddress)
```

```
VALUES (OLD.WarehouseID, OLD.WareName, NEW.WareName,  
OLD.City, NEW.City, OLD.District, NEW.District, OLD.Ward,  
NEW.Ward, OLD.Address, NEW.Address);
```

```
END //
```

```
DELIMITER ;
```

*Người dùng không thể tự đặt đơn hàng của mình, trigger để không thể add
ReciverID là Giver vào bảng OrderCreate*

```
# Tao trigger de ngan khi add du lieu vao 2 cot reciverID = OrderID
```

```
#DROP TRIGGER before_insert_OrderCreate;
```

```
DELIMITER $$
```

```
CREATE TRIGGER before_insert_OrderCreate
```

```
BEFORE insert ON OrderCreate
```

```
for each row
```

```
BEGIN
```

```
IF new.GiverID = new.ReciverID
```

```
then
```

```
signal SQLSTATE '45000'
```

```
SET MESSAGE_TEXT = 'Nguoi gui khong the tu dat don hang cua  
minh!';
```

```
end if;
```

```
END $$
```

```
DELIMITER ;
```

```
#DROP TRIGGER before_update_OrderCreate
```

```
DELIMITER $$
```

```
CREATE TRIGGER before_update_OrderCreate
```

```
BEFORE update ON OrderCreate
```

```
for each row
```

```
BEGIN
```

```

    IF new.GiverID = old.ReciverID or old.GiverID = new.ReciverID or
new.GiverID = new.ReciverID
    then
    signal SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Nguoi gui khong the tu dat don hang cua
minh!';
    end if;
END $$
DELIMITER ;

```

Tạo trigger để làm ra bảng Statusofproduct để lưu OrderID, CurrentWarehouseID (kho hiện tại), OrderStatus(trạng thái của đơn hàng). Các đơn hàng đang xử lý thì có kho mặc định là 0000, khi Đơn hàng đến kho cuối cùng và được giao cho shipper thì CurrentWarehouseID = Done.

```

# Trigger for Statusofproduct
create table Statusofproduct(
    OrderID CHAR(5) PRIMARY KEY,
    CurrentWarehouseID CHAR(5),
    OrderStatus VARCHAR(30)
);
# select * from Statusofproduct;

#DROP TRIGGER after_create_product;
# mỗi khi tạo một đơn hàng trong product sẽ thêm vào bảng Statusofproduct
các thông tin dưới đây
DELIMITER $$
CREATE TRIGGER after_create_product
    AFTER INSERT ON product
    for each row
BEGIN
    INSERT INTO Statusofproduct
    SET OrderID = new.OrderID,
        CurrentWarehouseID = '0000',
        OrderStatus = 'Dang xu ly';
END$$

```

DELIMITER ;

#drop TRIGGER after_delete_product;

#khi don hang bi xoa di khoi bang product -> update OrderStatus = 'don hang da bi huy'

DELIMITER \$\$

CREATE TRIGGER after_delete_product

AFTER delete ON product

for each row

BEGIN

UPDATE Statusofproduct as sta

SET OrderStatus = 'Don hang da bi huy'

where OLD.OrderID = sta.OrderID and sta.CurrentWarehouseID = '0000';

END\$\$

DELIMITER ;

#Drop TRIGGER after_insert_Warehouse;

mỗi khi thêm vào importexport thì update bảng statusofproduct

DELIMITER \$\$

CREATE TRIGGER after_insert_Warehouse

AFTER INSERT ON importexport

FOR EACH ROW

BEGIN

IF new.OutboundDate is not null

THEN

UPDATE Statusofproduct as Sta

SET CurrentWarehouseID = NEW.WarehouseID,
OrderStatus = 'Da roi kho'

WHERE NEW.OrderID = Sta.OrderID;

ELSE

UPDATE Statusofproduct as Sta

SET CurrentWarehouseID = NEW.WarehouseID,
OrderStatus = 'Dang trong kho'

WHERE NEW.OrderID = Sta.OrderID;

```

        end if;
    END$$
    DELIMITER ;

#DROP TRIGGER after_updateexportdate_Warehouse;
#MOI KHI UPDATE VAO importexport, ExportDate != Null -> update
Statusofproduct
DELIMITER $$
CREATE TRIGGER after_updateexportdate_Warehouse
AFTER Update  ON importexport
FOR EACH ROW
BEGIN
    IF new.OutboundDate is not null
    THEN
        UPDATE Statusofproduct as Sta
        SET OrderStatus = 'Da roi kho'
        WHERE NEW.OrderID = Sta.OrderID or Old.OrderID = sta.OrderID;
    end if;
END$$
DELIMITER ;

```

```

#DROP TRIGGER after_insert_send;
# moi khi insert vao bang send thi update bang statusofproduct
DELIMITER $$
CREATE TRIGGER after_insert_send
AFTER INSERT ON send
FOR EACH ROW
BEGIN
    IF NEW.SendStatus IS NULL
    THEN
        UPDATE Statusofproduct as sta
        SET CurrentWarehouseID = 'Done',
            OrderStatus = 'Dang giao hang'
        WHERE NEW.OrderID = sta.OrderID;
    ELSE

```

```
UPDATE Statusofproduct as sta
SET CurrentWarehouseID = 'Done',
    OrderStatus = NEW.SendStatus
WHERE NEW.OrderID = sta.OrderID;
end if;
END$$
DELIMITER ;
```

```
#Mỗi khi update mà actualdate != null thì hàng đã giao thành công
DELIMITER $$
CREATE TRIGGER after_update_send
AFTER UPDATE ON send
FOR EACH ROW
BEGIN
    IF OLD.SendStatus <> NEW.SendStatus THEN
        UPDATE Statusofproduct as sta
        SET OrderStatus = NEW.SendStatus
        where NEW.OrderID = sta.OrderID or Old.OrderID = sta.OrderID;
    end if;
END$$
DELIMITER ;
#select * from Statusofproduct;
```

	OrderID	CurrentWarehouseID	OrderStatus
►	P001	Done	Thanh cong
	P002	W064	Dang trong kho
	P003	Done	Dang giao hang
	P004	Done	Thanh cong
	P005	0000	Dang xu ly
	P006	Done	Thanh cong
	P007	0000	Don hang da bi huy
	P008	Done	Dang giao hang
	P009	W058	Da roi kho
	P010	Done	Thanh cong
	P011	Done	Thanh cong
	P012	Done	Dang giao hang
	P013	Done	Thanh cong
	P014	0000	Dang xu ly
	P015	W024	Dang trong kho
	P016	Done	Thanh cong
	P017	Done	Hoan
	P018	0000	Don hang da bi huy
	P019	Done	Dang giao hang
	P020	W000	Dang xu ly

1.4. Function

Function tính giá ship được niêm yết

- Tạo function ShipPrice tính tiền ship dựa trên giá niêm yết được lưu trong bảng Service tùy theo dịch vụ của đơn hàng có ServiceID được truyền vào.
- Câu lệnh:

DELIMITER \$\$

CREATE FUNCTION ShipPrice(OrderServiceID char(5))

RETURNS decimal(12, 3)

DETERMINISTIC

BEGIN

declare OrderShipPrice decimal(12,3) ;

select Price into OrderShipPrice from Service

where OrderServiceID = ServiceID;

return (OrderShipPrice);

END\$\$

DELIMITER ;

Function tính giá trị đơn

- Tạo function OrderPrice tính giá trị đơn bằng tổng giá trị các thành phần của đơn được lưu trong OrderDetails.
- Câu lệnh:

DELIMITER \$\$

CREATE FUNCTION ShipPrice(OrderServiceID char(5))

RETURNS decimal(12, 3)

DETERMINISTIC

BEGIN

declare OrderShipPrice decimal(12,3) ;

select Price into OrderShipPrice from Service

where OrderServiceID = ServiceID;

return (OrderShipPrice);

END\$\$

DELIMITER ;

Function tính phụ phí ship Cod

- Tạo function OrderCod tính tiền phụ phí cod. Nếu giá trị Payer là người nhận thì sẽ thực hiện tính phụ phí bằng 40% phí ship niêm yết, ngược lại thì phụ phí cod bằng 0 (Người gửi trả tiền → Không phải ship cod)
- Câu lệnh:

DELIMITER \$\$

CREATE FUNCTION OrderCod(OrderItemID char(5), Payer varchar(15))

RETURNS decimal(12, 3)

DETERMINISTIC

BEGIN

declare ShipPrice decimal(12,3);

```

        declare OrderCod decimal(12,3) ;
    if Payer = 'nguoi nhan' then
        select ShipPrice(ServiceID) into ShipPrice from product
    where OrderItemID = OrderID;
    set OrderCod = ShipPrice*0.4;
    else set OrderCod = 0.00;
    end if;
return (OrderCod);
END$$
DELIMITER ;

```

Function tính phụ phí các loại hàng

- Tạo function OrderSurcharge tính phụ phí loại hàng. Phụ phí loại hàng được tính bằng công thức:

$$\text{OrderSurcharge}(PP \text{ loại hàng}) = (\text{ShipPrice}(\text{giá ship}) + \text{OrderCod}(PP \text{ cod})) * \text{ItemSurcharge}(\text{Tổng phần trăm phụ phí loại hàng}).$$
 Với ShipPrice, OrderCod được tính bằng function ShipPrice, OrderCod và Item Surcharge được tính bằng tổng % phụ phí loại hàng có trong đơn được truy vấn từ bảng Surcharge và OrderDetails
- Câu lệnh:

```

DELIMITER $$
CREATE FUNCTION OrderSurcharge(OrderItemID char(5))
RETURNS decimal(12, 3)
DETERMINISTIC
BEGIN
    declare OrderSurcharge decimal(12,3);
    declare ItemSurcharge decimal(12,3) ;
    declare ShipPrice decimal(12,3) ;
    declare OrderCod decimal(12,3);

```

```
select sum(Surcharge.price) into ItemSurcharge from Surcharge,  
OrderDetails where OrderItemID = OrderID and OrderDetails.SurchargeID =  
Surcharge.SurchargeID
```

```
group by OrderID;
```

```
select ShipPrice(ServiceID) into ShipPrice from product where OrderItemID =  
OrderID;
```

```
select OrderCod(OrderID, Payer) into OrderCod from product where  
OrderItemID = OrderID;
```

```
set OrderSurcharge = (ShipPrice+OrderCod)*ItemSurcharge;
```

```
return (OrderSurcharge);
```

```
END$$
```

```
DELIMITER ;
```

Function tính tổng tiền cước:

- Tạo function Total tính tổng tiền cước theo công thức:
$$\text{TotalPrice}(\text{Tổng tiền}) = \text{ShipPrice}(\text{giá ship}) + \text{OrderPrice}(\text{giá trị đơn}) + \text{OrderCod}(\text{PP cod}) + \text{OrderSurcharge}(\text{PP loại hàng}).$$

Với các giá trị ShipPrice, OrderPrice, OrderCod, OrderSurcharge được tính từ các function cùng tên.
- Câu lệnh:

```
DELIMITER $$
```

```
CREATE FUNCTION Total(OrderItemID char(5))
```

```
RETURNS decimal(12, 3)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
declare ShipPrice decimal(12,3);
```

```
declare OrderPrice decimal(12,3);
```

```
declare OrderCod decimal(12,3);
```

```
declare ShipSurcharge decimal(12,3);
```

```

declare Total decimal(12,3) ;

#Xac dinh % phu phi

select ShipPrice(ServiceID) into ShipPrice from product where OrderItemID =
OrderID;

select OrderPrice(OrderID) into OrderPrice from product where OrderItemID =
OrderID;

select OrderCod(OrderID, Payer) into OrderCod from product where
OrderItemID = OrderID;

select OrderSurcharge(OrderID) into ShipSurcharge from product where
OrderItemID = OrderID;

set total = ShipPrice + OrderPrice + OrderCod + ShipSurcharge;

return (total);

END$$

DELIMITER ;

```

Function tính số tiền người nhận phải trả

- Tạo function RecipientPay để tính số tiền người nhận phải trả. Hàm trả về giá trị bằng Total(OrderID) nếu tham số Payer được truyền vào là người nhận, nếu không phải thì hàm trả về giá trị 0.
- Câu lệnh:

```

DELIMITER $$

CREATE FUNCTION RecipientPay( OrderItemID char(5), Payer varchar(15))
RETURNS decimal(12, 3)
DETERMINISTIC
BEGIN
    declare RecipientPay decimal(12,3);
    declare total decimal(12,3);

    if Payer = 'nguoi nhan' then
        select total(OrderID) into total from product

```

```

where OrderItemID = OrderID;
set RecipientPay = total;
else set RecipientPay = 0.00;
end if;
return (RecipientPay);
END$$

```

DELIMITER ;

Function tính tỷ lệ giao hàng đúng hạn

- Tạo function tính tỷ lệ giao hàng đúng hạn của mỗi shipper.
- Câu lệnh:

```

DELIMITER $$
CREATE FUNCTION EnableOrderPercent( PEmployeeID char(6) )
RETURNS decimal(5,2)
DETERMINISTIC
BEGIN
    declare OK int;
    declare total int;
    declare percent decimal(5,2);
    select count(orderid)
    into total
    from send
    where PEmployeeID = EmployeeID
    group by EmployeeID;
    select count(orderid)
    into ok
    from send
    where PEmployeeID = EmployeeID and EstimatedDate <= ActualDate
    group by EmployeeID;
    if (total = 0) then set percent = 0;
    else set percent = ok/total*100;
    end if;
    return (percent);
END$$

```

DELIMITER ;

Function tính tỷ lệ phân phối đơn hàng của các tỉnh

- Tính tỷ lệ phân phối đơn hàng nhận của các tỉnh

- Câu lệnh:

DELIMITER \$\$

Create function PercentCity(Cityname varchar(20))

returns decimal(5,3)

DETERMINISTIC

BEGIN

declare NumberofOrder smallint;

declare NumberOrderCity smallint;

declare PercentCity decimal(5,3);

select count(*) into NumberofOrder from Product;

select count(OrderID) into NumberOrderCity from Product

where Cityname = DeliveryCity

group by Cityname;

set PercentCity = NumberOrderCity/NumberofOrder*100;

return (PercentCity);

END\$\$

DELIMITER ;

Function tính số ngày kể từ khi tạo đơn hàng đến khi giao thành công.

DELIMITER \$\$

CREATE FUNCTION SUM_OF_DAY(Ma_dich_vu CHAR(5))

RETURNS INT

DECLARE Sumofday INT;

SELECT DATEDIFF((SELECT ActualDate from send where OrderID =
Ma_dich_vu), OrderDate)

INTO Sumofday

FROM ordercreate

WHERE OrderID = Ma_dich_vu;

return Sumofday;

END \$\$

Function tính số ngày trung bình của các đơn hàng giao thành công thuộc một dịch vụ vận chuyển.

- Câu lệnh:

```
CREATE FUNCTION tr(Ma_dich_vu CHAR(5))
RETURNS INT
DECLARE Trung_binh INT;
select AVG(SUM_OF_DAY(P.OrderID))
into Trung_binh
from (Product as P
      inner join statusofproduct as status on P.OrderID = status.OrderID)
where OrderStatus = 'Thanh cong' and
      s.ServiceID = Ma_dich_vu;
#select * from ordercreate order by OrderID;
#select * from send order by OrderID;
#select * from product where ServiceID = 'S303';
return Trung_binh;
END $$
```

1.5. View

- Tạo bảng view lưu từng thành phần giá tiền cước của đơn hàng.

- Câu lệnh:

```
create view PriceTotal as
select OrderID, ServiceID, ShipPrice(ServiceID) as ShipPrice,
OrderPrice(OrderID) as OrderPrice,
OrderCod(OrderID, Payer) as OrderCod, OrderSurcharge(OrderID) as
ShipSurcharge, Total(OrderID) as Total,
RecipientPay(OrderID, Payer) as RecipientPay
from product;
```

- Kết quả minh họa:

OrderID	ServiceID	ShipPrice	OrderPrice	OrderCod	ShipSurcharge	Total	RecipientPay
P001	S302	49000.000	8100000.000	19600.000	41160.000	8209760.000	8209760.000
P002	S207	54000.000	25190000.000	0.000	15120.000	25259120.000	0.000
P003	S203	29000.000	1070000.000	11600.000	11368.000	1121968.000	1121968.000
P004	S303	63000.000	4000000.000	0.000	12600.000	4075600.000	0.000
P005	S101	10000.000	10000000.000	4000.000	2800.000	10016800.000	10016800.000

1.6. Procedure

Procedure tra trạng thái đơn hàng

- Tra trạng thái đơn hàng và kho hiện tại của mình
- Câu lệnh:

```
DELIMITER $$
```

```
create procedure trang_thai(Ma_van_don Char(5))
```

```
BEGIN
```

```
SELECT *
```

```
from Statusofproduct
```

```
where OrderID = Ma_van_don;
```

```
END $$
```

```
DELIMITER ;
```

Procedure tra show các đơn hàng đã được vận chuyển qua kho từ ngày a đến ngày b

- Show các đơn hàng đã được vận chuyển qua kho từ ngày a đến ngày b:
- Câu lệnh:

```
Create index name_of_warehouse on warehouse(WareName);
```

```
Create index ID_In_out_Warehouse on importexport(WarehouseID,  
InboundDate, OutboundDate);
```

```
#drop procedure show_don_hang;
```

```
DELIMITER $$
```

```
CREATE procedure Show_don_hang(tenkho VARCHAR(30), ngaya DATE,  
ngayb DATE)
```

```
BEGIN
```



```

SELECT OrderID AS `Đơn hàng`

from importexport AS i

WHERE i.WarehouseID in (select WarehouseID from warehouse where
WareName = tenkho) AND

i.InboundDate >= ngaya AND

i.OutboundDate <= ngayb;

END$$

DELIMITER ;

```

Proceduce tra chặng đường đi (tất cả các kho mà đơn hàng đã đi qua)

```
DELIMITER $$
```

```
Create procedure Show_Warehouse( OrderItemID char(5))
```

```
BEGIN
```

```

select OrderID, Warehouse.WarehouseID, WareName
from Warehouse inner join ImportExport
on Warehouse.WarehouseID = ImportExport.WarehouseID
where OrderID = OrderItemID;

```

```
END$$
```

```
DELIMITER ;
```

2. Triển khai truy vấn

Câu 1: Tính tỷ lệ đơn hàng giao thành công của từng account người gửi.

- Đối tượng sử dụng: Người quản lí

```
DELIMITER $$
```

```
CREATE FUNCTION Successpercent(UserID CHAR(5))
```

```
RETURNS DECIMAL(5,2)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
DECLARE success_rate DECIMAL(5,2) DEFAULT 0.00;
```

```

# bang luu cac don da tao cua nguoi dung UserID
WITH CTEtable AS (
    SELECT POG.OrderID AS Orde, S.OrderStatus AS Sttus
    FROM Statusofproduct AS S
    INNER JOIN (
        SELECT OrderID
        FROM OrderCreate
        WHERE GiverID = UserID
    ) AS POG ON S.OrderID = POG.OrderID
)
SELECT ROUND((CAST((SELECT COUNT(Orde) FROM CTEtable
WHERE Sttus = 'Thanh cong') AS DECIMAL)
/ COUNT(Orde)) * 100, 2)
    INTO success_rate
    FROM CTEtable;
RETURN success_rate;

```

END\$\$

DELIMITER ;

#accouunt da tao va ty le giao hang thanh cong cua ho

create view `Bảng tỷ lệ bán hàng thành công` as

SELECT distinct UserID,

CASE

When userid in (select distinct giverid from ordercreate) then

Successpercent(UserID)

else 0.00

end AS `Tỷ lệ đơn hàng được giao thành công`

FROM accuser

Order by UserID;

select * from `Bảng tỷ lệ bán hàng thành công`;

- Giải thích:

- Hàm Successpercent(UserID) tính tỷ lệ thành công của các đơn hàng trên số đơn hàng đã được đặt của người gửi UserID. Giá trị trả về là decimal(5,2)
- View `Bảng tỷ lệ bán hàng thành công`: trả về kết quả UserID của người gửi cùng với tỷ lệ phần trăm thành công của đơn hàng.

- Kết quả minh họa:

UserID	Tỷ lệ đơn hàng được giao thành công
U007	100.00
U008	100.00
U011	100.00
U021	100.00
U030	100.00
U025	66.67
U010	50.00
U015	50.00

Câu 2: Trong tất cả các đơn hàng đã gửi thì tỷ lệ số đơn hàng thành công, hoàn, hủy lần lượt là?

- Đối tượng sử dụng: Người quản lí

- Câu lệnh truy vấn:

```
select * from Statusofproduct;
```

```
select Round(((select count(orderID) from Statusofproduct
where OrderStatus = 'Thanh cong'
group by OrderStatus) / count(OrderID)) * 100, 2)
as `Tỷ lệ đơn hàng được giao thành công`,
```

```
Round(((select count(orderID) from Statusofproduct
where OrderStatus = 'Hoan'
group by OrderStatus) / count(OrderID)) * 100, 2)
as `Tỷ lệ đơn hàng bị hoàn lại`,
```

```
Round(((select count(orderID) from Statusofproduct
where OrderStatus = 'Don hang da bi huy'
group by OrderStatus) / count(OrderID)) * 100, 2)
as `Tỷ lệ đơn hàng đã bị hủy`
```

```
from Statusofproduct;
```

- Trước khi sử dụng index:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	PRIMARY	Statusofproduct	<small>NULL</small>	index	<small>NULL</small>	PRIMARY	20	<small>NULL</small>	50	100.00	Using index
	4	SUBQUERY	Statusofproduct	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	50	10.00	Using where
	3	SUBQUERY	Statusofproduct	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	50	10.00	Using where
	2	SUBQUERY	Statusofproduct	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	50	10.00	Using where

- Sau khi sử dụng index

+ Index: create index order_status on Statusofproduct(OrderStatus);

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	Statusofproduct	NULL	index	NULL	order_status	123	NULL	50	100.00	Using index
	4	SUBQUERY	Statusofproduct	NULL	ref	order_status	order_status	123	const	5	100.00	Using index
	3	SUBQUERY	Statusofproduct	NULL	ref	order_status	order_status	123	const	2	100.00	Using index
	2	SUBQUERY	Statusofproduct	NULL	ref	order_status	order_status	123	const	18	100.00	Using index

- Kết quả minh họa:

Tỷ lệ đơn hàng được giao thành công	Tỷ lệ đơn hàng bị hoàn lại	Tỷ lệ đơn hàng đã bị hủy
35.29	3.92	9.80

Câu 3: Những dịch vụ nào được sử dụng nhiều nhất

- Đối tượng sử dụng: Người gửi, Shipper, Revenue Manager, Người quản lí, Người nhận
- Câu lệnh truy vấn:
with CTETable as(
select p.ServiceID ,servicename,count(OrderID) as c
from product as p
inner join Service as S
on S.ServiceID = p.ServiceID
group by S.ServiceID
order by c desc)
select * from CTETable
where c in (select max(c) from CTETable);
- Kết quả minh họa:

ServiceID	servicename	c
S207	Fast	6

Câu 4: Tính trung bình thời gian xử lý đơn hàng từ lúc tạo đến lúc giao thành công của từng dịch vụ vận chuyển

- Đối tượng sử dụng: Người gửi, Shipper, Người quản lí, Người nhận
- Câu truy vấn
DELIMITER \$\$
CREATE FUNCTION SUM_OF_DAY(Ma_van_don CHAR(5))
RETURNS INT

```

DETERMINISTIC
BEGIN
    DECLARE Sumofday INT;
    SELECT DATEDIFF((SELECT ActualDate from send where OrderID
= Ma_van_don), OrderDate)
    INTO Sumofday
    FROM ordercreate
    WHERE OrderID = Ma_van_don;
    return Sumofday;
END $$
DELIMITER ;
DELIMITER $$
CREATE FUNCTION tr(Ma_dich_vu CHAR(5))
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE Trung_binh INT;
    select AVG(SUM_OF_DAY(P.OrderID))
    into Trung_binh
    from (Product as P inner join statusofproduct as status on P.OrderID =
status.OrderID)
    where OrderStatus = 'Thanh cong' and
    ServiceID = Ma_dich_vu;
#select * from ordercreate order by OrderID;
#select * from send order by OrderID;
#select * from product where ServiceID = 'S303';
    return Trung_binh;
END $$
DELIMITER ;
#Bang show ket qua
select serviceid, servicename,
CASE
    when tr(serviceid) is not null THEN tr(serviceid)
    else 'Chua duoc su dung'
end as `Số ngày giao trung binh`

```

from service;

- Giải thích

- Sum_of_day(Ma_dich_vu): Function tính số ngày kể từ khi tạo đến khi nhận được đơn hàng.
- Tr(Ma_dich_vu): Function tính trung bình số ngày đã giao của các đơn hàng thành công của 1 dịch vụ.

- Kết quả minh họa:

serviceid	servicename	Số ngày giao trung bình
S101	Economy	Chưa được sử dụng
S102	Economy	2
S103	Economy	Chưa được sử dụng
S104	Economy	2
S105	Economy	Chưa được sử dụng
S106	Economy	10
S107	Economy	9
S201	Fast	8

Câu 5: Lọc ra những khách hàng đã tạo đơn và số đơn của họ từ đầu năm 2024 đến 01-06-2024, sắp xếp theo thứ tự số đơn giảm dần

- Đối tượng sử dụng: Người quản lý

- Câu lệnh truy vấn:

```
create index Order_Date on OrderCreate(OrderDate);
select GiverID, count(orderID) as So_Don from OrderCreate
where orderDate >= '2024-01-01'
group by GiverID
order by count(orderID) desc;
```

- So sánh hiệu suất khi dùng Index:

• Dùng Index:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
1	SIMPLE	OrderCreate	NULL	range	PRIMARY,OrderID,Order_Date	Order_Date	3	NULL	32	100.00

• Không dùng Index:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
1	SIMPLE	OrderCreate	NULL	index	PRIMARY,OrderID	PRIMARY	40	NULL	45	33.33

- Kết quả minh họa:

GiverID	So_Don
U015	3
U019	3
U024	2
U007	2
U010	2
U033	2
U002	2
U009	2
U003	2
U028	2
U020	2
U026	1

Câu 6: Biết những khách hàng có tổng số tiền ship (Không tính giá trị đơn) trên 150.000 là những khách hàng tiềm năng. Hãy lọc ra thông tin của những khách hàng này. (Tổng số tiền ship bao gồm giá ship, phụ phí cod, phụ phí phân loại hàng)

- Đối tượng sử dụng: Người quản lý
- Câu lệnh truy vấn:

```
select * from accuser
where userid in (
    select giverid
    from ordercreate inner join PriceTotal
    on ordercreate.OrderID = PriceTotal.OrderID
    group by GiverID
    having (sum(ShipPrice) + sum(OrderCod) + sum(ShipSurcharge)) >
    150000);
```
- Kết quả minh họa:

UserID	LastName	MiddleName	FirstName	Birthday	Gender	Phone	City	District	Ware	Address
U009	Tran	Thi	Bich	2002-02-16	Nu	0969273815	TP. Ho Chi Minh	Quan 1	Phuong 9	789 Duong Nguyen Van Nghi
U010	Hoang	Van	Phong	1992-07-10	Nam	0947162930	TP. Ho Chi Minh	Go Vap	Phuong Phu Nhuan	185 Hung Vuong
U012	Vo	Thi	Em	2002-12-05	Nu	0918937460	Can Tho	Cai Rang	An Khanh	So 14 An Khanh
U015	Nguyen	Thi	Hong	2007-11-05	Nu	0972839415	Ha Noi	Long Bien	Phuong Ngoc Lam	24 Ngo Tram
U019	Le	Van	Hai	1988-04-10	Nam	0942326688	Da Nang	Lien Chieu	Hoa Khanh	886 Duong Hoang Van Thai
U025	Bui	Van	Hung	1987-10-05	Nam	0917414710	Bac Giang	Yen The	Xa Yen The	99 Duong Dong Lac
U028	Pham	Van	Luan	1994-11-11	Nam	0917649751	Quang Binh	Dong Hoi	Hai Dinh	117 Duong Hai Dinh

Câu 7: Lọc ra thông tin những khách hàng đã tạo tài khoản nhưng chưa gửi một đơn hàng nào.

- Đối tượng sử dụng: Người quản lý
- Câu lệnh truy vấn:

```
select * from accuser
where userid not in (
    select giverid
```

from ordercreate);

- Kết quả minh họa:

UserID	LastName	MiddleName	FirstName	Birthday	Gender	Phone	City	District	Ware	Address
U001	Le	Thi	Cuc	2003-03-25	Nu	0948719302	Hue	Phu Hoi	Phu Nhuan	So 6 Phu Nhuan
U004	Pham	Van	Dung	1975-04-14	Nam	0920475186	TP. Ho Chi Minh	Binh Thanh	Phuong 11	123 Duong D2
U013	Duong	Thi	Lan	1997-09-25	Nu	0957681029	Hai Phong	Ngo Quyen	Le Chan	So 23A Le Chan
U014	Tran	Van	Minh	1996-09-22	Nam	0949302781	Lao Cai	Huyen Sa Pa	Xa Sa Pa	333 Duong Phan Xi Pang
U016	Nguyen	Thi	Lan	1990-05-16	Nu	0906946044	Da Nang	Hai Chau	Thuan Phuoc	So 45 Thuan Phuoc
U017	Tran	Van	Son	1985-11-02	Nam	0961742337	Hai Phong	Le Chan	An Duong	108 Duong An Duong
U018	Pham	Thi	Hue	1992-08-23	Nu	0987396094	Can Tho	Ninh Kieu	An Hoa	294 Duong 3/2
U023	Ly	Van	Thanh	2003-07-25	Nam	0960372201	Nam Dinh	Nam Truc	Truc Hung	505 Duong Truc Hung
U027	Tran	Thi	Hanh	2001-03-22	Nu	0901064092	Phu Tho	Viet Tri	Tien Cat	86 Duong Tien Cat
U032	Vu	Van	Tuan	1990-12-29	Nam	0907113310	Quang Ngai	Son Tinh	Tinh Son	176 Duong Tinh Son
U034	Tran	Van	Kiet	1988-05-18	Nam	0974627239	Binh Duong	Di An	Tan Dong Hiep	845 Duong Tan Dong Hiep
U035	Pham	Thi	Hong	1995-07-25	Nu	0939109692	Ben Tre	Chau Thanh	Tam Phuoc	41 Duong Tam Phuoc

Câu 8: Lọc ra ID kho, tên kho và số đơn của top 10 kho hàng có nhiều đơn hàng đi qua nhất.

- Đối tượng sử dụng: Người quản lí
- Câu lệnh truy vấn:
select warehouse.WarehouseID, Warename, count(orderid) as SoDon
from Warehouse, importexport
where warehouse.warehouseid = importexport.warehouseid
group by warehouseid
order by SoDon desc, warename asc
limit 10;
- Kết quả minh họa:

WarehouseID	Warename	SoDon
W029	Kho Quang Binh	12
W062	Kho Da Nang	11
W030	Kho Quang Tri	11
W032	Kho Thua Thien Hue	11
W042	Kho Ho Chi Minh	10

Câu 9: Biết shipper phải giao đúng hạn 50% số đơn và số đơn lớn hơn bằng 2 mới đạt tiêu chuẩn. Lọc ra thông tin của những shipper đạt tiêu chuẩn.

- Đối tượng sử dụng: Người quản lí, Shipper
- Câu lệnh truy vấn:
select distinct shipper.employeeid, CONCAT(LastName, ' ', MiddleName, ' ',
FirstName) as HoTen, count(orderid) as SoDon,
EnableOrderPercent(shipper.EmployeeID) as SuccessPercent


```

from shipper inner join send
on shipper.employeeid = send.employeeid
group by shipper.employeeid
having SuccessPercent >= 50.00;

```

- Giải thích:
 - Sử dụng Function EnableOrderPercent, truy vấn từ Shipper, Send và lọc ra những EmployeeID tương ứng có EnableOrderPercent > 50%.
- Kết quả minh họa:

employeeid	HoTen	SoDon	SuccessPercent
SP001	Nguyen Van Anh	6	50.00
SP009	Chu Thi Mai	2	50.00
SP006	Tran Van Hai	2	50.00
SP012	Pham Minh Nhat	1	100.00

Câu 10: Lọc ra tỷ lệ phân phối đơn hàng của các địa chỉ nhận ở các tỉnh (Không hiện các tỉnh có tỷ lệ bằng 0)

- Đối tượng sử dụng: Người quản lí, Người gửi
- Câu lệnh truy vấn:


```

select distinct DeliveryCity as City, count(OrderID) as TongSoDon,
PercentCity(DeliveryCity) as Percent from Product
group by DeliveryCity
order by percent desc;

```
- Kết quả minh họa:

City	TongSoDon	Percent
Da Nang	7	15.556
TP. Ho Chi Minh	6	13.333
Hai Phong	5	11.111
Ha Noi	4	8.889
Quang Ninh	3	6.667
Thai Binh	3	6.667
Bac Giang	2	4.444
Ha Nam	2	4.444
Ninh Binh	2	4.444
Can Tho	1	2.222
Nghe An	1	2.222
Vinh Phuc	1	2.222
Lang Son	1	2.222
Gia Lai	1	2.222
Hai Duong	1	2.222
Thai Nguyen	1	2.222

Câu 11: Liệt kê số lượng đơn hàng đã giao thành công theo từng nhân viên giao hàng trong tháng 5/2024

- Đối tượng sử dụng: Shipper, Người quản lí

- Câu lệnh truy vấn:

```
SELECT s.EmployeeID, CONCAT(e.FirstName, ' ', e.MiddleName, ' ',
e.LastName) AS ShipperName, COUNT(*) AS SoLuongDonHang
```

```
FROM Send s
```

```
JOIN Shipper e ON s.EmployeeID = e.EmployeeID
```

```
WHERE s.SendStatus = 'Thanh cong'
```

```
AND YEAR(s. ActualDate) = 2024
```

```
AND MONTH(s. ActualDate) = 5
```

```
GROUP BY s.EmployeeID;
```

- Kết quả minh họa:

EmployeeID	ShipperName	SoLuongDonHang
SP009	Mai Thi Chu	1

Câu 12: Lấy thông tin về các đơn hàng đã qua kho có tên 'Kho Ninh Binh' vào ngày 29/5/2024

- Đối tượng sử dụng: Người quản lí
- Câu lệnh truy vấn:

With temp as(

```
select p.OrderID, CONCAT(a.LastName, ' ', a.MiddleName, ' ', a.FirstName)
AS `Tên người gửi`, p.PickupCity, p.RecipientName, p.DeliveryCity, p.ServiceID
```

```
from (OrderCreate as OC
```

```
inner join accuser as a on UserID = GiverID)
```

```
inner join product as p on OC.OrderID = p.OrderID)
```

```
SELECT ie.OrderID, `Tên người gửi`, PickupCity, RecipientName as `Tên người
nhận`, DeliveryCity, ServiceID, ie.WarehouseID, ie.InboundDate,
ie.OutboundDate
```

```
FROM (ImportExport as ie
```

```
inner join
```

```
(select WarehouseID from Warehouse where WareName = 'Kho Ninh Binh') as
w ON ie.WarehouseID = w.WarehouseID)
```

```
inner join temp on ie.OrderID = temp.OrderID
```

```
WHERE DATE(ie.InboundDate) = '2024-05-29';
```

- Kết quả minh họa:

OrderID	Tên người gửi	PickupCity	Tên người nhận	DeliveryCity	ServiceID	WarehouseID	InboundDate	OutboundDate
P027	Nguyen Van Hung	Thanh Hoa	Duong Thi Lan	Hai Phong	S204	W024	2024-05-29	2024-05-29
P045	Tran Van Quang	Ha Noi	Le Van Hai	Da Nang	S206	W024	2024-05-29	2024-05-29

Câu 13: Shipper giao nhiều đơn hàng nhất vào năm 2023 sẽ được trao danh hiệu “Shipper Star 2023”

- Đối tượng sử dụng: Người quản lí
- Câu lệnh truy vấn:

```
ALTER TABLE Shipper
```

```
ADD Status VARCHAR(30);
```

```

create view ShipperDeliveries AS (
    SELECT s.EmployeeID, COUNT(sd.OrderID) AS NumberOfDeliveries
    FROM Shipper s
    JOIN Send sd ON s.EmployeeID = sd.EmployeeID
    WHERE YEAR(sd.ReceiptDate) = 2023
    GROUP BY s.EmployeeID
);

with star as(
    select EmployeeID from ShipperDeliveries
    where NumberOfDeliveries in (select max(NumberOfDeliveries) from
ShipperDeliveries)
)

UPDATE Shipper
SET Status = 'Shipper Star 2023'
WHERE EmployeeID =
    (select employeeID from star);

SELECT
    EmployeeID,
    CONCAT(LastName, ' ', MiddleName, ' ', FirstName) AS ShipperName,
    IF(Status IS NULL, '0000', Status) AS ShipperStatus
FROM Shipper;

```

- Kết quả minh họa:

EmployeeID	ShipperName	ShipperStatus
SP001	Nguyen Van Anh	Shipper Star 2023

Câu 14: Tính tổng số đơn hàng đã được tạo bởi từng người gửi

- Đối tượng sử dụng: Người dùng.
- Câu lệnh truy vấn:
SELECT u.UserID, COALESCE(COUNT(DISTINCT o.OrderID), 0) AS
TotalOrders
FROM AccUser u
LEFT JOIN OrderCreate oc ON u.UserID = oc.GiverID
LEFT JOIN Product o ON oc.OrderID = o.OrderID
GROUP BY u.UserID;
- Giải thích: Đếm số lượng đơn hàng đã được tạo bởi mỗi người dùng.
- Kết quả minh họa:

UserID	TotalOrders
U001	0
U002	2
U003	2
U004	0
U005	1
U006	1

Câu 15: Thống kê các khách hàng có sinh nhật trong tháng hiện tại để tặng ưu đãi

- Đối tượng sử dụng: Người quản lý
- Câu lệnh truy vấn:
SELECT UserID, FirstName, LastName, Birthday
FROM AccUser
WHERE MONTH(Birthday) = MONTH(CURRENT_DATE()) AND
DAY(Birthday) >= 1
ORDER BY DAY(Birthday)
- Giải thích: Thống kê các khách hàng có sinh nhật trong tháng hiện tại
- Kết quả minh họa:

UserID	FirstName	LastName	Birthday
U026	Hung	Nguyen	1990-06-12

Câu 16: Thống kê số lượng đơn hàng theo từng tháng của một năm cụ thể

- Đối tượng sử dụng: Người dùng, Người quản lý
- Câu lệnh truy vấn:

```

SELECT YEAR(OrderDate) AS Year, MONTH(OrderDate) AS Month,
COUNT(OrderID) AS TotalOrders
FROM OrderCreate
GROUP BY YEAR(OrderDate), MONTH(OrderDate)
ORDER BY Year, Month;

```

- Giải thích: Truy vấn này thống kê số lượng đơn hàng theo từng tháng của một năm cụ thể
- Kết quả minh họa:

Year	Month	TotalOrders
2021	3	1
2021	6	1
2021	9	1
2021	12	1
2022	6	2
2022	8	1
2022	9	1
2022	12	1

Câu 17: Thống kê các đơn hàng giao và nhận tại mỗi tỉnh/thành

- Đối tượng sử dụng: Người quản lý
- Câu lệnh truy vấn:

```

SELECT City,
COUNT(CASE WHEN Type = 'Giao hang' THEN 1 ELSE NULL END)
AS `Số lượng đơn hàng giao`,
COUNT(CASE WHEN Type = 'Nhận hàng' THEN 1 ELSE NULL END)
AS `Số lượng đơn hàng nhận`
FROM (
SELECT PickupCity AS City, 'Giao hang' AS Type FROM Product
UNION ALL
SELECT DeliveryCity AS City, 'Nhận hàng' AS Type FROM Product
) AS CombinedData
GROUP BY City;

```
- Giải thích: Truy vấn số đơn hàng tại mỗi tỉnh/thành
- Kết quả minh họa:

City	Số lượng đơn hàng giao	Số lượng đơn hàng nhận
Bac Giang	7	2
Quang Binh	3	0
Can Tho	5	1
Thai Nguyen	2	1
Ha Noi	10	4
Quang Tri	4	0
TP. Ho Chi Minh	4	6
Ninh Binh	1	2
Thanh Hoa	1	0
Quang Ninh	1	3
Da Nang	2	8
Thai Binh	1	3
Hue	2	1
Ha Nam	1	2
Nha Trang	1	1
Hai Phong	1	5
Nghe An	0	1
Vinh Phuc	0	1

Câu 18: Giả sử vào các ngày đặc biệt (ngày bằng tháng hoặc ngày đầu tháng) có chương trình khuyến mãi. Thống kê các đơn hàng được đặt vào ngày đặc biệt trong năm 2022

- Đối tượng sử dụng: Người quản lí
- Câu lệnh truy vấn:

```
SELECT DATE(OrderDate) AS `Ngày đặt hàng`,
        COUNT(*) AS `Số lượng đơn hàng`
FROM OrderCreate
WHERE (
        DAY(OrderDate) = MONTH(OrderDate) or DAY(OrderDate) = 1
        AND YEAR(OrderDate) = 2022
    )
GROUP BY DATE(OrderDate)
ORDER BY `Ngày đặt hàng` DESC;
```
- Giải thích: Thống kê số đơn hàng được đặt vào ngày có số ngày bằng tháng hoặc đầu tháng.
- Kết quả minh họa:

Ngày đặt hàng	Số lượng đơn hàng
2022-09-09	1
2022-06-01	2

Câu 19: Tra các kho mà đơn hàng P008 đã đi qua

- Đối tượng sử dụng: Người gửi, Người nhận, Người quản lý
- Câu lệnh truy vấn:
call Show_Warehouse('P008');
- Kết quả minh họa:

OrderID	WarehouseID	WareName
P008	W007	Kho Bac Giang
P008	W027	Kho Ha Tinh
P008	W032	Kho Thua Thien Hue
P008	W042	Kho Ho Chi Minh
P008	W048	Kho Can Tho
P008	W057	Kho Binh Dinh