

**GVHD: Trương Quỳnh Chi**

**Lớp: L05**

**Sinh viên:**

**Nguyễn Thị Trúc Ly 1710187**

**Nguyễn Đăng Hà Nam 1710195**

**Ngô Thanh Liêm 1711929  
Huỳnh Phạm Phước Linh 1710165**

**Trần Văn Tâm 1713057**

TP.Hồ Chí Minh, Tháng 12/2019

**HỆ CƠ SỞ DỮ LIỆU**

BÁO CÁO BÀI TẬP LỚN SỐ 2

**Topic 1: Design database for an Ecommerce system**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC – KỸ THUẬT MÁY TÍNH**

MỤC LỤC

[**I.** **PHẦN CHUNG** 3](#_Toc27025730)

[**1.** **Các câu lệnh tạo bảng và ràng buộc** 3](#_Toc27025731)

[**2.** **Các câu lệnh tạo chỉ mục** 3](#_Toc27025732)

[**3.** **Các câu lệnh insert dữ liệu** 3](#_Toc27025733)

[**II.** **PHẦN RIÊNG** 3](#_Toc27025734)

[**1.** **Huỳnh Phạm Phước Linh – 1710165** 3](#_Toc27025735)

[**1.1.** **Thủ tục insert dữ liệu** 3](#_Toc27025736)

[**1.2.** **Trigger** 3](#_Toc27025737)

[**1.3.** **Thủ tục chứa câu SQL** 3](#_Toc27025738)

[**1.4.** **Hàm** 3](#_Toc27025739)

[**2.** **Nguyễn Thị Trúc Ly – 1710187** 3](#_Toc27025740)

[**2.1.** **Thủ tục insert dữ liệu** 3](#_Toc27025741)

[**2.2.** **Trigger** 3](#_Toc27025742)

[**2.3.** **Thủ tục chứa câu SQL** 4](#_Toc27025743)

[**2.4.** **Hàm** 4](#_Toc27025744)

[**3.** **Nguyễn Đăng Hà Nam – 1710195** 4](#_Toc27025745)

[**3.1.** **Thủ tục insert dữ liệu** 4](#_Toc27025746)

[**3.2.** **Trigger** 4](#_Toc27025747)

[**3.3.** **Thủ tục chứa câu SQL** 4](#_Toc27025748)

[**3.4.** **Hàm** 4](#_Toc27025749)

[**4.** **Ngô Thanh Liêm – 1711929** 5](#_Toc27025750)

[**4.1.** **Thủ tục insert dữ liệu** 5](#_Toc27025751)

[**4.2.** **Trigger** 5](#_Toc27025752)

[**4.3.** **Thủ tục chứa câu SQL** 5](#_Toc27025753)

[**4.4.** **Hàm** 5](#_Toc27025754)

[**5.** **Trần Văn Tâm – 1713057** 5](#_Toc27025755)

[**5.1.** **Thủ tục insert dữ liệu** 5](#_Toc27025756)

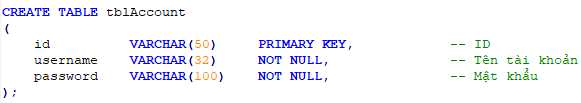
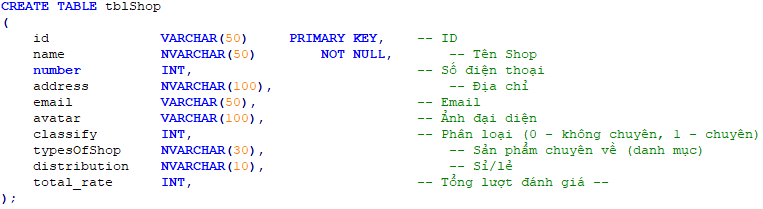
[**5.2.** **Trigger** 5](#_Toc27025757)

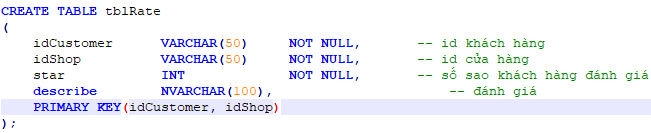
[**5.3.** **Thủ tục chứa câu SQL** 5](#_Toc27025758)

[**5.4.** **Hàm** 6](#_Toc27025759)

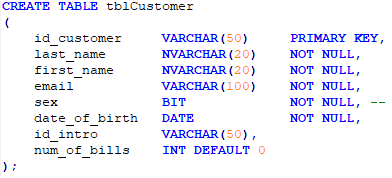
[**III.** **PHỤ LỤC** 6](#_Toc27025760)

[**Bảng phân công công việc** 6](#_Toc27025761)

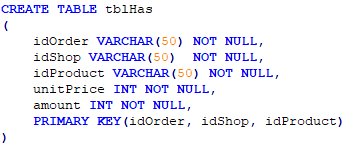
1. **PHẦN CHUNG**
2. **Các câu lệnh tạo bảng và ràng buộc**
   1. **Bảng Account**
   2. **Bảng Shop**

**1.3. Bảng Rate**

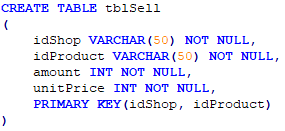
**1.4. Bảng Customer**



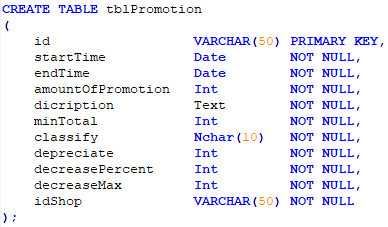
**1.5. Bảng Has**

****

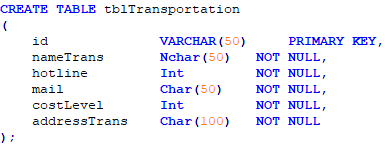
**1.6. Bảng Sell**

****

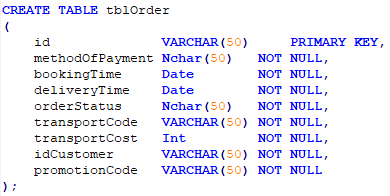
**1.7. Bảng Promotion**

****

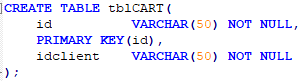
**1.8. Bảng Transportation**

****

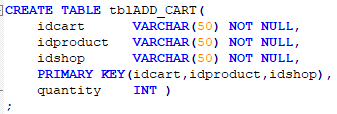
**1.9. Bảng Order**

****

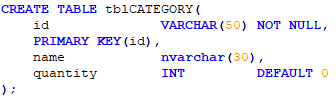
**1.10. Bảng CART**

****

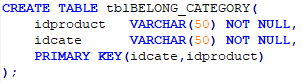
**1.11. Bảng ADD\_CART**

****

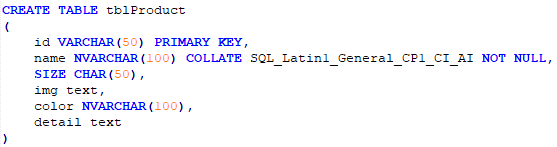
**1.12. Bảng CATEGORY**



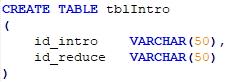
**1.13. Bảng BELONG\_CATEGORY**



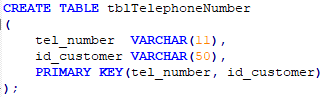
**1.14. Bảng Product**

****

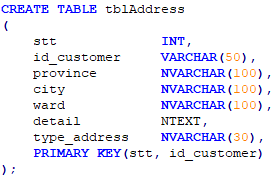
**1.15. Bảng Intro**



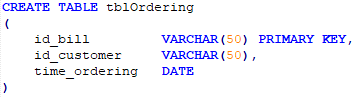
**1.16. Bảng TelephoneNumber**



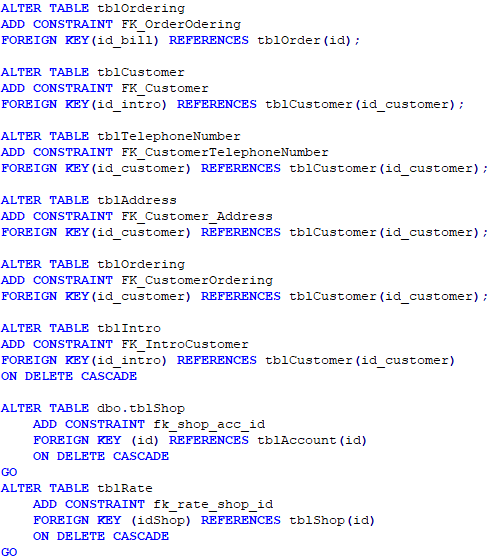
**1.17. Bảng Address**

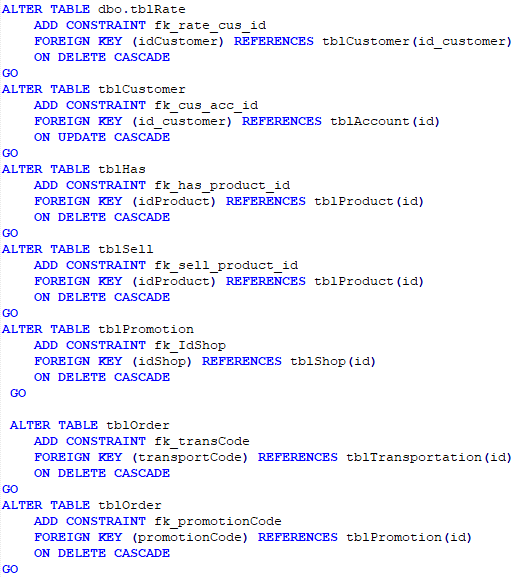


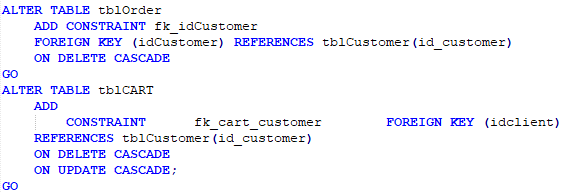
**1.18. Bảng Ordering**

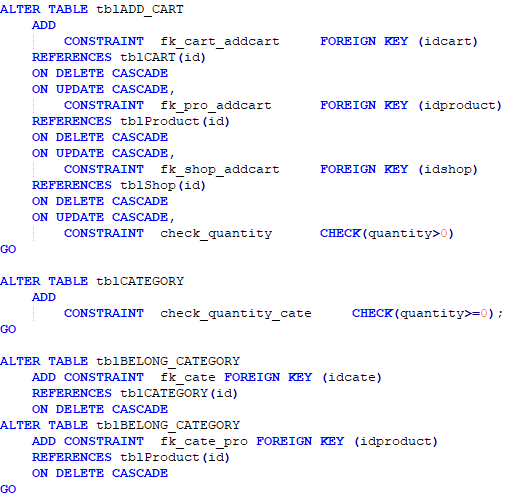
****

**1.19. Các ràng buộc**

****

****

****

****

1. **Các câu lệnh tạo chỉ mục**

CREATE INDEX indexAccount

ON tblAccount (id)

CREATE INDEX indexADD\_CART

ON tblADD\_CART (idcart)

CREATE INDEX tblAdrress

ON tblAdrress (addr)

CREATE INDEX tblBELONG\_CATEGORY

ON tblBELONG\_CATEGORY (idproduct)

CREATE INDEX tblCATEGORY

ON tblCATEGORY (id)

CREATE INDEX tblCustomer

ON tblCustomer (id\_customer)

CREATE INDEX tblHas

ON tblHas (idOrder)

CREATE INDEX tblOrder

ON tblOrder (id)

CREATE INDEX tblProduct

ON tblProduct (id)

CREATE INDEX tblPromotion

ON tblPromotion (id)

CREATE INDEX tblRate

ON tblRate (idCustomer)

CREATE INDEX tblSell

ON tblSell (idShop)

CREATE INDEX tblShop

ON tblShop (id)

CREATE INDEX tblTelephoneNumber

ON tblTelephoneNumber (tel\_number)

CREATE INDEX tblTransportation

ON tblTransportation (id)

1. **Các câu lệnh insert dữ liệu**

**INSERT** **INTO** dbo.tblTelephoneNumber **VALUES** ('0834562109', 'KH0006')

**INSERT** **INTO** dbo.tblTelephoneNumber **VALUES** ( '0395914514', 'KH0002')

**INSERT** **INTO** dbo.tblTelephoneNumber **VALUES** ('0123456789', 'KH0003')

**INSERT** **INTO** dbo.tblTelephoneNumber **VALUES** ('0998656689', 'KH0005')

**INSERT** **INTO** dbo.tblOrdering **VALUES**('MDH001', 'KH0006', '')

**INSERT** **INTO** dbo.tblOrdering **VALUES**('MDH002', 'KH0002', '20190312')

**INSERT** **INTO** dbo.tblOrdering **VALUES**('MDH003', 'KH0002', '20190312')

**INSERT** **INTO** dbo.tblOrdering **VALUES**('MDH004', 'KH0002', '20190312')

**INSERT** **INTO** dbo.tblOrdering **VALUES**('MDH005', 'KH0002', '20191119')

**INSERT** **INTO** tblOrder **VALUES** ('MDH001','Trực tiếp','2019-12-09','2019-12-15','Đang xử lý','VIETTEL',30000,'','BLACKFRIDAY')

**INSERT** **INTO** tblOrder **VALUES** ('MDH002','Chuyển khoản','2019-11-09','2019-11-12','Đã giao','GHN',10000,'','TET2020')

**INSERT** **INTO** tblOrder **VALUES** ('MDH003','Chuyển khoản','2019-12-01','2019-12-05','Đã giao','GRAB',23000,'','TIPEEMOMO')

**INSERT** **INTO** tblOrder **VALUES** ('MDH004','Trực tiếp','2019-12-09','2019-12-16','Đang giao','VNPOST',50000,'','NOEL')

**INSERT** **INTO** tblOrder **VALUES** ('MDH005','Trực tiếp','2019-11-19','2019-11-21','Đang xử lý','ALOGN',15000,'','FREESHIP')

1. **PHẦN RIÊNG**
2. **Huỳnh Phạm Phước Linh – 1710165**
   1. **Thủ tục insert dữ liệu**
      1. **Mô tả**
      2. **Câu lệnh tạo**

CREATE PROCEDURE insertShopAccount

@id VARCHAR(6),

@username VARCHAR(32),

@password VARCHAR(50),

@name NVARCHAR(50),

@number INT,

@address NVARCHAR(100),

@email VARCHAR(50),

@avatar VARCHAR(100),

@classify INT,

@typesOfShop NVARCHAR(30),

@distribution NVARCHAR(10)

AS

BEGIN

DECLARE @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

DECLARE @afterHashPassword VARCHAR(100) = CONVERT(VARCHAR(500), @hashPass)

DECLARE @count\_id INT = (SELECT COUNT(id) FROM dbo.tblAccount WHERE id = @id)

DECLARE @count\_user INT = (SELECT COUNT(username) FROM dbo.tblAccount WHERE username = @username)

IF @count\_id = 0 AND @count\_user = 0

BEGIN

INSERT INTO dbo.tblAccount(id, username, password) VALUES (@id, @username, @afterHashPassword)

IF @@ROWCOUNT > 0

BEGIN

INSERT INTO dbo.tblShop(id, name, number, address, email, avatar, classify, typesOfShop, distribution, total\_rate) VALUES (@id, @name, @number, @address, @email, @avatar, @classify, @typesOfShop, @distribution, 0)

END

END

END

* + 1. **Câu lệnh thực thi**
    2. **Kết quả**
  1. **Trigger**
     1. **Mô tả**
     2. **Câu lệnh tạo**

CREATE TRIGGER trgCountRate ON dbo.tblRate

AFTER INSERT

AS

BEGIN

DECLARE @count\_idCustomer INT = (SELECT COUNT(idCustomer) FROM dbo.tblRate WHERE idShop = (SELECT idShop FROM inserted))

UPDATE dbo.tblShop SET total\_rate = @count\_idCustomer WHERE id = (SELECT idShop FROM inserted)

END

GO

------------------------------------------------------------------------------

CREATE TRIGGER trgDeleteRate ON dbo.tblRate

FOR DELETE

AS

BEGIN

DECLARE @cur\_idShop VARCHAR(6) = (SELECT idShop FROM deleted)

DECLARE @cur\_idCustomer VARCHAR(6) = (SELECT idCustomer FROM deleted)

DECLARE @curTotalRate INT = (SELECT total\_rate FROM tblShop WHERE id = @cur\_idShop) - 1

PRINT @curTotalRate

UPDATE dbo.tblShop

SET total\_rate = @curTotalRate

WHERE id = @cur\_idShop

END

GO

------------------------------------------------------------------------------

CREATE TRIGGER trgCheckEmail ON dbo.tblShop

AFTER INSERT

AS

BEGIN

DECLARE @cur\_email VARCHAR(50) = (SELECT email FROM inserted)

DECLARE @check AS BIT;

IF @cur\_email NOT LIKE '%@%.%'

BEGIN

PRINT 'Format of email is %@%.%'

DELETE FROM dbo.tblAccount WHERE id = (SELECT id FROM inserted)

--ROLLBACK

--UPDATE dbo.tblShop SET email = (SELECT email FROM tblShop WHERE id = (SELECT id FROM inserted)) + '@gmail.com' WHERE id = (SELECT id FROM inserted)

END

ELSE

PRINT 'SUCCESS'

END

* + 1. **Kết quả**
  1. **Thủ tục chứa câu SQL**
     1. **Mô tả**
     2. **Câu lệnh tạo**
     3. **Câu lệnh thực thi**

CREATE PROCEDURE insertShopAccount

@id VARCHAR(6),

@username VARCHAR(32),

@password VARCHAR(50),

@name NVARCHAR(50),

@number INT,

@address NVARCHAR(100),

@email VARCHAR(50),

@avatar VARCHAR(100),

@classify INT,

@typesOfShop NVARCHAR(30),

@distribution NVARCHAR(10)

AS

BEGIN

DECLARE @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

DECLARE @afterHashPassword VARCHAR(100) = CONVERT(VARCHAR(500), @hashPass)

DECLARE @count\_id INT = (SELECT COUNT(id) FROM dbo.tblAccount WHERE id = @id)

DECLARE @count\_user INT = (SELECT COUNT(username) FROM dbo.tblAccount WHERE username = @username)

IF @count\_id = 0 AND @count\_user = 0

BEGIN

INSERT INTO dbo.tblAccount(id, username, password) VALUES (@id, @username, @afterHashPassword)

IF @@ROWCOUNT > 0

BEGIN

INSERT INTO dbo.tblShop(id, name, number, address, email, avatar, classify, typesOfShop, distribution, total\_rate) VALUES (@id, @name, @number, @address, @email, @avatar, @classify, @typesOfShop, @distribution, 0)

END

END

END

GO

CREATE PROCEDURE writeReviewShop

@idShop VARCHAR(6),

@username VARCHAR(32),

@password VARCHAR(100),

@star INT,

@describe NVARCHAR(100)

AS

BEGIN

DECLARE @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

DECLARE @afterHashPassword VARCHAR(100) = CONVERT(VARCHAR(500), @hashPass)

IF (@afterHashPassword = (SELECT password FROM dbo.tblAccount WHERE username = @username)) AND (@username LIKE 'kh%')

BEGIN

IF @describe = ''

PRINT 'Write comment'

ELSE

BEGIN

DECLARE @countRate AS INT

DECLARE @idCustomer VARCHAR(6) = (SELECT id FROM dbo.tblAccount WHERE username = @username)

SET @countRate = (SELECT COUNT(idShop) FROM dbo.tblRate WHERE idCustomer = @idCustomer AND idShop = @idShop)

IF @countRate = 0

BEGIN

INSERT INTO dbo.tblRate(idShop, idCustomer, star, describe)

VALUES (@idShop, @idCustomer, @star, @describe)

PRINT 'Success'

END

ELSE

BEGIN

UPDATE dbo.tblRate

SET star = @star, describe = @describe

WHERE idShop = @idShop AND idCustomer = @idCustomer

PRINT 'Success'

END

END

END

ELSE

PRINT 'Password wrong!'

END

GO

CREATE PROCEDURE deleteReviewShop

@idShop VARCHAR(6),

@username VARCHAR(32),

@password VARCHAR(100)

AS

BEGIN

DECLARE @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

DECLARE @afterHashPassword VARCHAR(100) = CONVERT(VARCHAR(500), @hashPass)

IF @afterHashPassword != (SELECT password FROM dbo.tblAccount WHERE username = @username)

PRINT 'Password wrong!'

ELSE

BEGIN

DECLARE @idCustomer VARCHAR(6) = (SELECT id FROM dbo.tblAccount WHERE username = @username)

DELETE FROM dbo.tblRate WHERE idCustomer = @idCustomer

END

END

GO

CREATE PROCEDURE insertCustomerAccount

@id\_customer VARCHAR(6),

@username VARCHAR(32),

@password VARCHAR(50),

@last\_name NVARCHAR(20),

@first\_name NVARCHAR(20),

@email VARCHAR(100),

@sex BIT,

@date\_of\_birth DATE,

@id\_intro VARCHAR(15),

@id\_reduce VARCHAR(15)

AS

BEGIN

DECLARE @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

DECLARE @afterHashPassword VARCHAR(100) = CONVERT(VARCHAR(500), @hashPass)

DECLARE @count\_id INT = (SELECT COUNT(id) FROM dbo.tblAccount WHERE id = @id\_customer)

DECLARE @count\_user INT = (SELECT COUNT(username) FROM dbo.tblAccount WHERE username = @username)

IF @count\_id = 0 AND @count\_user = 0

BEGIN

INSERT INTO dbo.tblAccount(id, username, password) VALUES (@id\_customer, @username, @afterHashPassword)

IF @@ROWCOUNT > 0

BEGIN

INSERT INTO dbo.tblCustomer(id\_customer, last\_name, first\_name, email, sex, date\_of\_birth, id\_intro, id\_reduce) VALUES (@id\_customer, @last\_name, @first\_name, @email, @sex, @date\_of\_birth, @id\_intro, @id\_reduce)

END

END

END

GO

CREATE PROCEDURE procedureChangePassword

@username VARCHAR(50),

@curpassword VARCHAR(32),

@newpassword VARCHAR(100)

AS

BEGIN

DECLARE @hashCurPass AS VARCHAR(500)

SET @hashCurPass = CONVERT(VARCHAR(500), HASHBYTES('SHA2\_512', @curpassword))

DECLARE @password AS VARCHAR(500)

SET @password = (SELECT password FROM dbo.tblAccount WHERE username = @username)

IF @hashCurPass = @password

BEGIN

UPDATE dbo.tblAccount SET password = CONVERT(VARCHAR(500), HASHBYTES('SHA2\_512', @newpassword)) WHERE username = @username;

print 'Change Password Success!'

END

ELSE

BEGIN

print 'Incorrect current password'

END

END

GO

CREATE PROCEDURE procedureChangeProfileShop

@id VARCHAR(6),

@username VARCHAR(32),

@curpassword VARCHAR(100),

@name NVARCHAR(50),

@number INT,

@address NVARCHAR(50),

@email VARCHAR(50),

@avatar VARCHAR(100),

@classify INT,

@typesOfShop NVARCHAR(30),

@distribution NVARCHAR(10)

AS

BEGIN

DECLARE @hashCurPass AS VARCHAR(500)

SET @hashCurPass = CONVERT(VARCHAR(500), HASHBYTES('SHA2\_512', @curpassword))

DECLARE @password AS VARCHAR(500)

SET @password = (SELECT password FROM dbo.tblAccount WHERE username = @username)

IF @hashCurPass = @password

BEGIN

--UPDATE dbo.tblAccount SET password = CONVERT(VARCHAR(500), HASHBYTES('SHA2\_512', @newpassword)) WHERE username = @username;

UPDATE DBO.tblShop SET

name = @name,

number = @number,

address = @address,

email = @email,

avatar = @avatar,

classify = @classify,

typesOfShop = @typesOfShop,

distribution = @distribution

WHERE

id = @id;

print 'Change profile shop success!'

END

ELSE

BEGIN

print 'Incorrect current password'

END

END

GO

* + 1. **Kết quả**
  1. **Hàm**
     1. **Mô tả**
     2. **Câu lệnh tạo**
     3. **Câu lệnh thực thi**
     4. **Kết quả**

1. **Nguyễn Thị Trúc Ly – 1710187**
   1. **Thủ tục insert dữ liệu**
      1. **Mô tả**

Thêm các thông tin cơ bản của một khách hàng

* + 1. **Câu lệnh tạo**

**CREATE** **PROCEDURE** insertCustomerAccount

@id\_customer **VARCHAR**(6),

@username **VARCHAR**(32),

@password **VARCHAR**(50),

@last\_name NVARCHAR(20),

@first\_name NVARCHAR(20),

@email **VARCHAR**(100),

@sex BIT,

@date\_of\_birth **DATE**,

@tel\_num **VARCHAR**(11),

@stt **INT**,

@province NVARCHAR(100),

@city NVARCHAR(100),

@ward NVARCHAR(100)

**AS**

**BEGIN**

**DECLARE** @hashPass VARBINARY(500) = HASHBYTES('SHA2\_512', @password)

**DECLARE** @afterHashPassword **VARCHAR**(100) = **CONVERT**(**VARCHAR**(500), @hashPass)

**DECLARE** @count\_id **INT** = (**SELECT** **COUNT**(id) **FROM** dbo.tblAccount **WHERE** id = @id\_customer)

**DECLARE** @count\_user **INT** = (**SELECT** **COUNT**(username) **FROM** dbo.tblAccount **WHERE** username = @username)

**IF** @email **NOT** **LIKE** '%@%'

**BEGIN**

RAISERROR (N'Email sai định dạng', 1, 1)

**END**

**ELSE** **IF** **EXISTS**(**SELECT** \* **FROM** dbo.tblCustomer **WHERE** @email = email)

**BEGIN**

RAISERROR (N'Email này đã tồn tại!', 1, 1)

**END**

**ELSE**

**IF** @count\_id = 0 **AND** @count\_user = 0

**BEGIN**

**INSERT** **INTO** dbo.tblAccount(id, username, password) **VALUES** (@id\_customer, @username, @afterHashPassword)

**INSERT** **INTO** dbo.tblCustomer(id\_customer, last\_name, first\_name, email, sex, date\_of\_birth)

**VALUES** (@id\_customer, @last\_name, @first\_name, @email, @sex, @date\_of\_birth)

**INSERT** **INTO** dbo.tblTelephoneNumber **VALUES** (@tel\_num, @id\_customer)

**INSERT** **INTO** dbo.tblAddress **VALUES**(@stt,@id\_customer,@province,@city,@ward,**NULL**,'')

**END**

**ELSE**

**BEGIN**

RAISERROR (N'Tên tài khoản hoặc ID đã tồn tại!', 1, 1)

**END**

**END**

* + 1. **Câu lệnh thực thi**

**EXEC** insertCustomerAccount 'KH0007','kh07','kh07','Nguyen', 'Ly','truclybk.cse@gmail.com', 0,'19990218','0834562109',1,'An Giang','Long Xuyên','Mỹ Hòa Hưng';

* + 1. **Kết quả**
  1. **Trigger**
     1. **Trigger 1**

**Mô tả**

Sau khi một đơn hàng được đặt, thì tăng tổng số đơn hàng của khách hàng

**Câu lệnh tạo**

**GO**

**CREATE** **TRIGGER** updateTotalBills **ON** tblOrdering AFTER **INSERT** **AS**

**BEGIN**

**DECLARE** @count\_id\_bill **INT** =

(

**SELECT** **COUNT**(id\_bill)

**FROM** tblOrdering

**WHERE** id\_customer = (**SELECT** id\_customer **FROM** Inserted)

)

**UPDATE** tblCustomer **SET** num\_of\_bills = @count\_id\_bill

**WHERE** id\_customer = (**SELECT** id\_customer **FROM** Inserted)

**END**

**Kết quả**

* + 1. **Trigger 3**

**Mô tả**

Sau khi khách hàng nhập mã giới thiệu bằng id của một khách hàng khác, thì kiểm tra mã nhập có hợp lệ không, đồng thời thêm mã giảm vào người giới thiệu

**Câu lệnh tạo**

**GO**

**CREATE** **TRIGGER** checkIntro **ON** tblCustomer **FOR** **UPDATE** **AS**

**BEGIN**

**DECLARE** @count\_bill\_old **INT** = (**SELECT** num\_of\_bills **FROM** Deleted)

**DECLARE** @count\_bill\_cur **INT** = (**SELECT** num\_of\_bills **FROM** Inserted)

**IF** (@count\_bill\_cur = @count\_bill\_old)

**BEGIN**

**DECLARE** @id\_customer **VARCHAR**(6)

**DECLARE** @id\_cur\_intro **VARCHAR**(6)

**DECLARE** @id\_intro **VARCHAR**(6)

**SET** @id\_customer = (**SELECT** id\_customer **FROM** inserted)

**SET** @id\_intro = (**SELECT** id\_intro **FROM** inserted)

**SET** @id\_cur\_intro = (**SELECT** id\_intro **FROM** deleted)

**IF** (@id\_cur\_intro **IS** **NOT** **NULL**)

**BEGIN**

PRINT 'Tài khoản này đã được giới thiệu'

**ROLLBACK**

**END**

**IF** (@id\_customer = @id\_intro)

**BEGIN**

PRINT 'Không được tự giới thiệu chính mình'

**ROLLBACK**

**END**

**INSERT** **INTO** tblIntro(id\_intro, id\_reduce) **VALUES** (@id\_intro, @id\_customer)

**END**

**END**

**Câu lệnh thực thi**

**UPDATE** dbo.tblCustomer **SET** id\_intro = 'KH0005' **WHERE** id\_customer = 'KH0001'

**Kết quả**

* 1. **Thủ tục chứa câu SQL**
     1. **Thủ tục 1**

**Mô tả**

Truy vấn tất cả khách hàng ở cùng một tỉnh với tỉnh được nhập vào và sắp xếp theo id

**Câu lệnh tạo**

**GO**

**CREATE** **PROCEDURE** queryCustomersInOneProvince

@province nvarchar(100)

**AS**

**BEGIN**

**SELECT** tblCustomer.id\_customer, first\_name, tel\_number

**FROM** tblAddress, tblCustomer, tblTelephoneNumber

**WHERE** province = @province

**AND** tblAddress.id\_customer = tblCustomer.id\_customer

**AND** tblAddress.id\_customer = tblTelephoneNumber.id\_customer

**ORDER** **BY** tblCustomer.id\_customer

**END**

**Câu lệnh thực thi**

**EXEC** queryCustomersInOneProvince 'An Giang'

**Kết quả**

* + 1. **Thủ tục 2**

**Mô tả**

Truy vấn tất cả các hóa đơn của một khách hàng trước một thời gian nhập vào và có trạng thái là đang xử lý

**Câu lệnh tạo**

**GO**

**CREATE** **PROCEDURE** queryBillsBeforeOneDate

@**DATE** **DATE**,

@id\_customer **VARCHAR**(6)

**AS**

**BEGIN**

**SELECT** id\_bill, time\_ordering

**FROM** tblOrdering, tblOrder

**WHERE** id\_customer = @id\_customer **AND** DATEDIFF(**DAY**, @**DATE**, time\_ordering) > 0

**GROUP** **BY** time\_ordering, orderStatus, id\_bill

**HAVING** orderStatus = 'Đang xử lý'

**ORDER** **BY** time\_ordering

**END**

**Câu lệnh thực thi**

**INSERT** **INTO** tblOrder **VALUES** ('MDH001','Trực tiếp','2019-12-09','2019-12-15','Đang xử lý','VIETTEL',30000,'','BLACKFRIDAY')

**Kết quả**

* 1. **Hàm**
     1. **Hàm 1**

**Mô tả**

Tính điểm của một khách hàng nào đó dựa vào tổng hóa đơn đã đặt.

Với các điều kiện như sau:

Hóa đơn từ 0 đến 10 thì mỗi hóa đơn tương ứng 1 điểm

Hóa đơn từ 11 đến 50 thì mỗi hóa đơn tương ứng 2 điểm

Hóa đơn từ 51 đến 100 thì mỗi hóa đơn tương ứng 3 điểm

Hóa đơn từ 101 trở lên thì mỗi hóa đơn tương ứng 5 điểm

**Câu lệnh tạo**

**GO**

**CREATE** **FUNCTION** getScoreOfCustomer(@id\_customer **VARCHAR**(6))

**RETURNS** **INT**

**AS**

**BEGIN**

**DECLARE** @scores **INT**

**SET** @scores = 0

**DECLARE** @num\_of\_bills **INT** = (**SELECT** num\_of\_bills **FROM** dbo.tblCustomer **WHERE** id\_customer = @id\_customer)

**IF** @num\_of\_bills <= 10

**BEGIN**

**SET** @scores = @num\_of\_bills

**END**

**ELSE** **IF** @num\_of\_bills <= 50

**BEGIN**

**SET** @scores = 10 + (@num\_of\_bills - 10) \* 2

**END**

**ELSE** **IF** @num\_of\_bills <= 100

**BEGIN**

**SET** @scores = 10 + 40 \* 2 + (@num\_of\_bills - 50) \* 3

**END**

**ELSE**

**BEGIN**

**SET** @scores = 10 + 40 \* 2 + 50 \* 3 + (@num\_of\_bills - 100) \* 5

**END**

**RETURN** @scores

**END**

**Câu lệnh thực thi**

**GO**

**SELECT** dbo.getScoreOfCustomer('KH0001')

**Kết quả**

* + 1. **Hàm 2**

**Mô tả**

Hàm trả ra bảng các khách hàng có độ tuổi trong một khoảng và có cùng một tháng sinh, với biên trên, biên dưới của tuổi và tháng được nhập vào

**Câu lệnh tạo**

**GO**

**CREATE** **FUNCTION** queryCusInMonthBirth(@**MONTH** **INT**, @age\_low **INT**, @age\_high **INT**)

**RETURNS** @tblCustomerBirth **TABLE**

(

id\_customer **VARCHAR**(6),

last\_name NVARCHAR(20),

first\_name NVARCHAR(20),

email **VARCHAR**(100),

sex BIT,

date\_of\_birth **DATE**

)

**AS**

**BEGIN**

**IF** @**MONTH** < 0 **OR** @**MONTH** > 12

**BEGIN**

**RETURN**

**END**

**INSERT** **INTO** @tblCustomerBirth

**SELECT** id\_customer, last\_name, first\_name, email, sex, date\_of\_birth

**FROM** tblCustomer

**WHERE** **MONTH**(date\_of\_birth) = @**MONTH** **AND** @age\_low < (**YEAR**(GETDATE()) - **YEAR**(date\_of\_birth)) **AND** (**YEAR**(GETDATE()) - **YEAR**(date\_of\_birth)) < @age\_high

**RETURN**

**END**

**Câu lệnh thực thi**

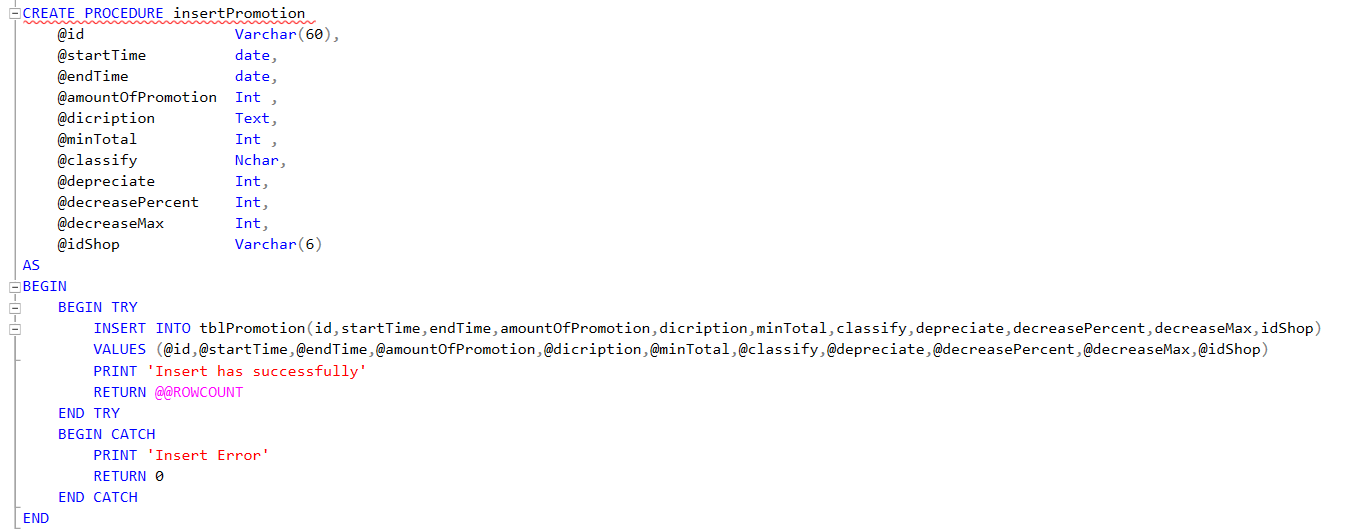
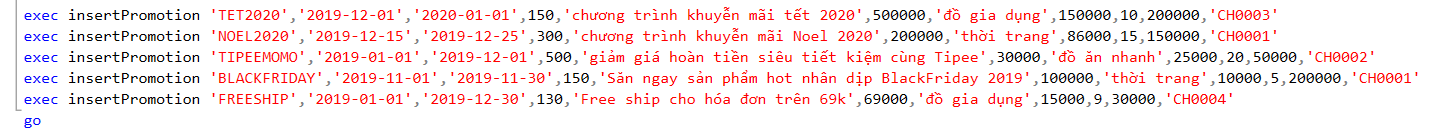
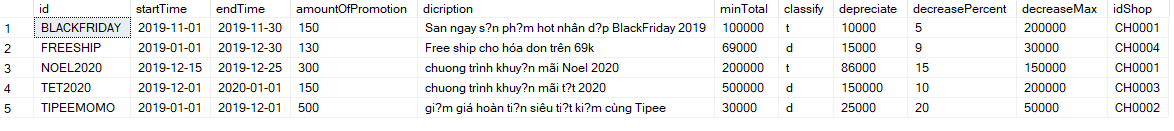
**GO**

**SELECT** \* **FROM** queryCusInMonthBirth(2, 18, 30)

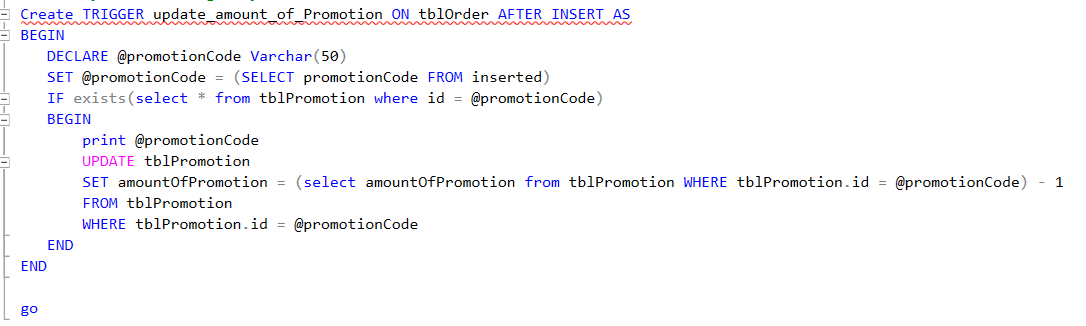
**Kết quả**

1. **Nguyễn Đăng Hà Nam – 1710195**
   1. **Thủ tục insert dữ liệu**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   2. **Trigger**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   3. **Thủ tục chứa câu SQL**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   4. **Hàm**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
2. **Ngô Thanh Liêm – 1711929**
   1. **Thủ tục insert dữ liệu**
      1. **Mô tả**

Thêm dữ liệu cơ bản cho chương trình khuyễn mãi

* + 1. **Câu lệnh tạo**
    2. **Câu lệnh thực thi**
    3. **Kết quả**
  1. **Trigger**
     1. **Mô tả**

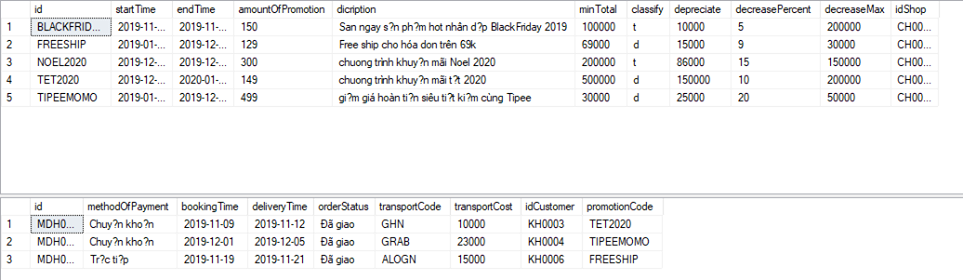
Cập nhật số lượng áp dụng khuyễn mãi còn lại

* + 1. **Câu lệnh tạo**
    2. **Câu lệnh thực thi**

exec insertOrder 'MDH014','Chuyển khoản','2019-12-01','2019-12-05','Đã giao','GRAB',23000,'','BLACK'

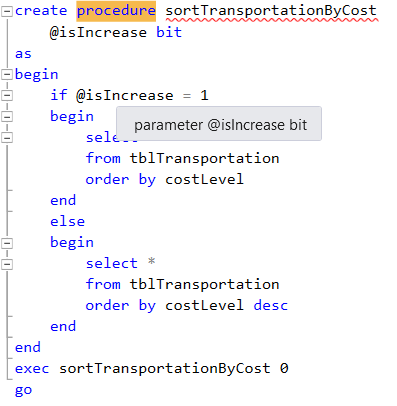
select \* from tblPromotion

select \* from tblOrder

* + 1. **Kết quả**
  1. **Thủ tục chứa câu SQL**
     1. **Mô tả**

Sắp xếp lại theo phí vận chuyển của các đơn vị

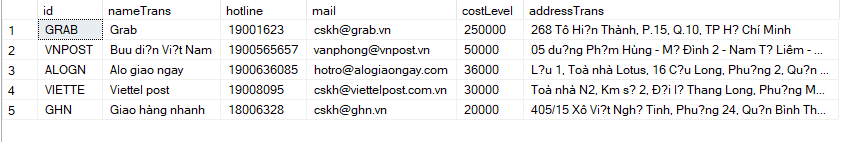
* + 1. **Câu lệnh tạo**

****

* + 1. **Câu lệnh thực thi**

exec sortTransportationByCost 0

go

* + 1. **Kết quả**
  1. **Hàm**
     1. **Mô tả**

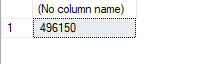
Tính tổng tiền theo phần trăm được giảm

* + 1. **Câu lệnh tạo**
    2. **Câu lệnh thực thi**

SELECT dbo.totalMoneyByPercent('MDH001')

go

* + 1. **Kết quả**



1. **Trần Văn Tâm – 1713057**
   1. **Thủ tục insert dữ liệu**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   2. **Trigger**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   3. **Thủ tục chứa câu SQL**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
   4. **Hàm**
      1. **Mô tả**
      2. **Câu lệnh tạo**
      3. **Câu lệnh thực thi**
      4. **Kết quả**
2. **PHỤ LỤC**

**Bảng phân công công việc**

|  |  |
| --- | --- |
| **Thành viên** | **Công việc** |
| Linh |  |
| Ly |  |
| Nam |  |
| Liêm |  |
| Tâm |  |