

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÀI TẬP LỚN SỐ 2

**XÂY DỰNG HỆ CƠ SỞ DỮ LIỆU CHO HỆ THỐNG CHUỖI
CỬA HÀNG KINH DOANH QUẦN ÁO**

GVHD: TRƯƠNG QUỲNH CHI

NHÓM THỰC HIỆN: L02.F

STT	Họ & Tên	MSSV
1	Phạm Văn Thành	1814029
2	Nguyễn Hoàn Thành	1915146
3	Nguyễn Minh Tùng	1915831
4	Nguyễn Trường Sơn	1914968
5	Trần Văn Cường	1810852

MỤC LỤC

1. Phần chung	1
1.1 Các câu lệnh và tạo ràng buộc	2
1.2 Các câu lệnh tạo chỉ mục	2
1.3 Các câu lệnh Insert data	2
1.4 Database Diagram	2
Error! Bookmark not defined.	
5. Phần cá nhân – Nguyễn Trường Sơn – 1914968	2
5.1. Thủ tục INSERT và hiển thị dữ liệu.	2
5.1.1. Mô tả chức năng	2
5.1.2. Câu lệnh tạo thủ tục	3
5.1.3. Câu lệnh mẫu thực thi thủ tục	3
5.2. TRIGGER	4
5.2.1. Mô tả chức năng	4
5.2.2. Câu lệnh tạo TRIGGER	4
5.2.3. Câu lệnh kiểm tra TRIGGER hoạt động và kết quả	5
5.3. Câu SQL	6
5.3.1. Mô tả chức năng	6
5.3.2. Câu lệnh SQL	7
5.3.3. Kết quả thực thi câu lệnh SQL	7
5.4. Hàm	8
5.4.1. Mô tả chức năng	8
5.4.2. Câu lệnh tạo hàm	8
5.4.3. Câu lệnh SELECT minh họa gọi hàm	9
5.5. Giao diện WEB	10
7. Phụ lục	10
7.1 Báo cáo bài tập lớn số 1	10
7.2 Source code chương trình	10
7.3 Bảng phân công nhiệm vụ cho phần chung và bài tập lớn số 1	11

1. PHẦN CHUNG

1.1 Các câu lệnh và tạo ràng buộc

- File CREATE-ALTER-TABLE.sql

1.2 Các câu lệnh tạo chỉ mục

```
CREATE INDEX idx_date ON Order_product(order_date, receive_date);
CREATE INDEX ind_totalAmount ON Customer(totalAmount);
CREATE INDEX idx_signup_day ON Nomal_customer(signup_day);
CREATE INDEX idx_city ON Customer(city);
CREATE INDEX idx_zip_code ON Customer(zip_code);
CREATE INDEX idx_country ON Supplier_shop(country);
CREATE INDEX idx_email ON Supplier_contact(email);
CREATE INDEX idx_sname ON Shop(sname);
CREATE INDEX idx_facebook ON Shop_contact facebook);
CREATE INDEX idx_ename ON Employee(Name);
```

1.3 Các câu lệnh Insert data

- File INSERT.sql

1.4 Database Diagram

- File Databasediagram.png

5. PHẦN CÁ NHÂN – Nguyễn Trường Sơn – 1914968

5.1. Thủ tục INSERT và hiển thị dữ liệu.

5.1.1. Mô tả chức năng

Do ảnh hưởng nghiêm trọng của dịch bệnh Covid-19 hiện nay, nên cửa hàng thực hiện tạo một thủ tục nhằm:

- Insert thông tin khách hàng mua hàng lần đầu vào bảng ‘Customer’.
- Kiểm tra thông tin địa chỉ của khách hàng có phải khu vực bị ảnh hưởng nghiêm trọng bởi dịch bệnh Covid-19 (bị phong tỏa):
 - *Khu vực “Mỹ Phú, Châu Phú, An Giang” hiện đang bị phong tỏa nên không thể mua hàng (Insert thông tin khách hàng).*

5.1.2. Câu lệnh tạo thủ tục

--"Do tình hình diễn biến phức tạp của dịch bệnh Covid-19, tạm thời cửa hàng sẽ không nhận đăng ký khách hàng mới từ khách hàng trong vùng nguy hiểm."

```
CREATE OR ALTER PROCEDURE Insert_customer_COVID (  
    @customer_id INT,  
    @lname VARCHAR(15),  
    @minit VARCHAR(15),  
    @fname VARCHAR(15),  
    @birthday DATE,  
    @gender CHAR,  
    @street VARCHAR(30),  
    @district VARCHAR(30),  
    @city VARCHAR(20),  
    @zip_code CHAR(6),  
    @total_amount DECIMAL(11,2)  
)  
AS  
BEGIN  
    IF (@street = 'My Phu' AND @district = 'Chau Phu' AND @city = 'An Giang')  
        THROW 50005, N'Do dien bien phuc tap cua dich benh  
        covid-19 nen dia chi nay khong duoc ho tro!', 1  
    ELSE  
        BEGIN  
            INSERT INTO Customer VALUES  
                (@customer_id, @lname, @minit, @fname, @birthday, @gender,  
                @street, @district, @city, @zip_code, @total_amount)  
        END  
END  
GO
```

5.1.3. Câu lệnh mẫu thực thi thủ tục

--"Câu lệnh mẫu thực thi thủ tục hiển thị thông tin báo lỗi có nghĩa".

```
EXEC Insert_customer_COVID  
@customer_id = 202100008,  
@lname = 'Nguyen',  
@minit = 'Hong',  
@fname = 'Phuc',  
@birthday = '1994-01-25',  
@gender = 'F',  
@street = 'My Phu',  
@district = 'Chau Phu',  
@city = 'An Giang',  
@zip_code = 921496,  
@total_amount = 0;  
GO
```

%

Messages

Msg 50005, Level 16, State 1, Procedure Insert_customer_COVID, Line 17 [Batch Start Line 134]
Do dien bien phuc tap cua dich benh
covid-19 nen dia chi nay khong duoc ho tro!

Completion time: 2021-11-30T11:57:58.7980417+07:00

5.2. TRIGGER

5.2.1. Mô tả chức năng

➤ TRIGGER [KIỂM TRA MÃ KHÁCH HÀNG]

Mã khách hàng tại cửa hàng được quy định là một chuỗi số mã hóa gồm chín số, trong đó bốn số đầu tiên của mỗi mã khách hàng biểu thị cho Năm mà khách hàng được cung cấp mã (mua hàng lần đầu tại hệ thống chuỗi cửa hàng). *Trigger* này nhằm:

- Kiểm soát Mã khách hàng được Insert hay Update vào database: *Chuỗi của hàng được mở vào năm 2021 nên không thể có mã khách hàng nào nhỏ hơn "202100000"*.

➤ TRIGGER [CẬP NHẬT DISCOUNT]

- Sử dụng TRIGGER để cập nhật discount trên Bảng Vip_customer từ câu IF để kiểm tra giá trị của Tổng số tiền khách hàng đã mua hàng tại cửa hàng trên bảng Customer. Nếu Tổng số tiền trong khoảng từ 15-20 triệu đồng thì cập nhật discount là 0.05. Nếu tổng số tiền lớn hơn hoặc bằng 20 triệu thì cập nhật discount là 0.1.

5.2.2. Câu lệnh tạo TRIGGER

➤ KIỂM TRA MÃ KHÁCH HÀNG

-- Kiểm tra mã khách hàng gồm 9 số trong đó 4 số đầu tiên thể hiện năm mà khách hàng được cung cấp mã (Điều kiện: 'customer_id' > 202100000).

```
CREATE OR ALTER TRIGGER Check_customer_id ON Customer
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @customer_id INT;
    SELECT @customer_id = customer_id FROM INSERTED;
    IF (@customer_id < 202100000)
    BEGIN
        RAISERROR(N'Mã khách hàng phải lớn hơn số 202100000
        vui lòng tạo lại mã customerID khác!', 16, 1);
        ROLLBACK;
    END;
END;
GO
```

➤ CẬP NHẬT DISCOUNT (VIP_CUSTOMER)

```
CREATE OR ALTER TRIGGER Update_discount ON Vip_customer
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS ( SELECT * FROM Customer Cus WHERE totalAmount > 15000000 AND totalAmount < 20000000)
    BEGIN
        UPDATE Vip_customer
        SET discount = 0.05
        WHERE customer_id IN (SELECT C.customer_id
                             FROM Customer C, Vip_customer V
                             WHERE C.customer_id = V.customer_id
                             AND TotalAmount > 15000000 AND TotalAmount < 20000000)
    END;

    BEGIN
        UPDATE Vip_customer
        SET discount = 0.10
        WHERE customer_id IN (SELECT C.customer_id
                             FROM Customer C, Vip_customer V
                             WHERE C.customer_id = V.customer_id AND totalAmount >= 20000000)
    END;
END;
GO
```

5.2.3. Câu lệnh kiểm tra TRIGGER hoạt động và kết quả

➤ KIỂM TRA MÃ KHÁCH HÀNG

--Sử dụng câu lệnh Store Procedure đã tạo ở câu 5.1 để insert dữ liệu không hợp lệ để kiểm tra hoạt động của TRIGGER.

```
EXEC Insert_customer_COVID
@customer_id = 201900011,
@lname = 'Nguyen',
@minit = 'Thanh',
@fname = 'Son',
@birthday = '1992-01-25',
@gender = 'M',
@street = 'Kha Van Can',
@district = 'Thu Duc',
@city = 'Ho Chi Minh',
@zip_code = 700000,
@total_amount = 0;
GO
```

Messages

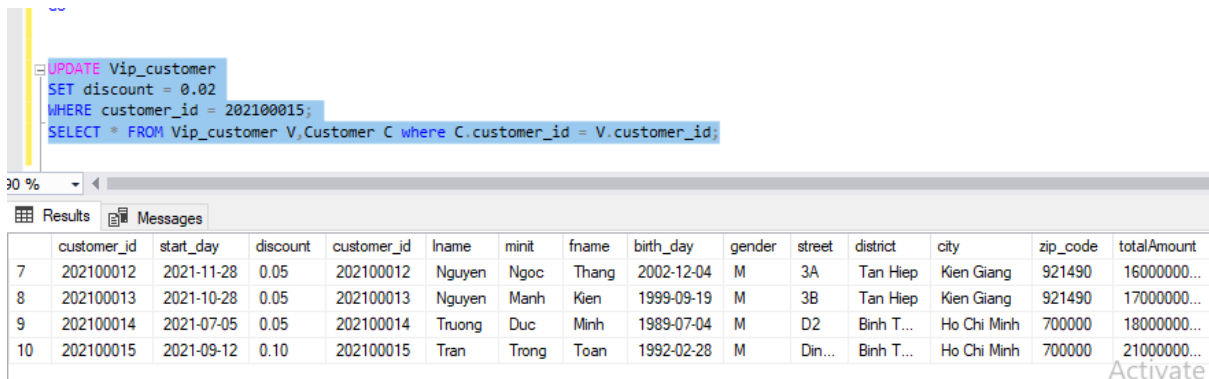
Msg 50000, Level 16, State 1, Procedure Check_customer_id, Line 9 [Batch Start Line 167]
Mã khách hàng phải lớn hơn số 202100000 vui lòng tạo lại mã customerID khác!

Msg 3609, Level 16, State 1, Procedure Insert_customer_COVID, Line 21 [Batch Start Line 167]
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2021-11-30T12:11:09.4760609+07:00

➤ KIỂM TRA CẬP NHẬT DISCOUNT

--Sử dụng câu lệnh UPDATE để kiểm tra TRIGGER. Discount vẫn là 0.1 mà không bị thay đổi.



The screenshot shows a SQL IDE with a query editor at the top and a results pane at the bottom. The query in the editor is:

```
UPDATE Vip_customer  
SET discount = 0.02  
WHERE customer_id = 202100015;  
SELECT * FROM Vip_customer V, Customer C where C.customer_id = V.customer_id;
```

The results pane shows a table with 15 columns: customer_id, start_day, discount, customer_id, lname, minit, fname, birth_day, gender, street, district, city, zip_code, and totalAmount. The table contains 4 rows of data, with the discount for customer_id 202100015 updated to 0.10.

	customer_id	start_day	discount	customer_id	lname	minit	fname	birth_day	gender	street	district	city	zip_code	totalAmount
7	202100012	2021-11-28	0.05	202100012	Nguyen	Ngoc	Thang	2002-12-04	M	3A	Tan Hiep	Kien Giang	921490	16000000...
8	202100013	2021-10-28	0.05	202100013	Nguyen	Manh	Kien	1999-09-19	M	3B	Tan Hiep	Kien Giang	921490	17000000...
9	202100014	2021-07-05	0.05	202100014	Truong	Duc	Minh	1989-07-04	M	D2	Binh T...	Ho Chi Minh	700000	18000000...
10	202100015	2021-09-12	0.10	202100015	Tran	Trong	Toan	1992-02-28	M	Din...	Binh T...	Ho Chi Minh	700000	21000000...

5.3. Câu SQL

5.3.1. Mô tả chức năng

➤ **Truy xuất danh sách khách hàng mua hàng lần đầu trong năm 2021 theo thứ tự thời gian từ sớm nhất đến trễ nhất.**

- Hiện thị thông tin về Họ, Tên đệm, Tên, Ngày tạo mã.
- Có sử dụng mệnh đề WHERE và ORDER BY.
- Dữ liệu được truy xuất từ hai bảng “Customer” và “Nomal_customer”.

Mục đích thực hiện câu truy xuất này nhằm xác định các khách hàng mà cửa hàng đã tiếp cận được trong năm đầu hoạt động để triển khai hoạt động tri ân khách hàng.

➤ **Truy xuất Tổng doanh thu (totalAmount) phân theo từng tỉnh được xếp theo thứ tự giảm dần từ các khách hàng VIP được cấp Mã khách hàng trong năm 2021.**

- Sử dụng hàm SUM tính tổng doanh thu từ các khách hàng VIP được cấp mã trong năm 2021; sử dụng hàm COUNT để đếm số lượng khách VIP có trong bảng cho mệnh đề having.
- Sử dụng mệnh đề WHERE, GROUP BY để truy xuất Tổng doanh thu đến từ khách hàng VIP phân theo từng tỉnh.
- Sử dụng mệnh đề HAVING để loại trừ đi những tỉnh mà tổng doanh thu chỉ đến từ một khách hàng VIP.
- Sử dụng mệnh đề ORDER BY để sắp xếp bảng thông tin trả về theo thứ tự giảm dần của giá trị Tổng doanh thu.

Mục đích thực hiện câu truy xuất này nhằm xác định các tỉnh có tiềm năng nhất (khả năng bán được hàng, nhận doanh thu cao) để từ đó ra quyết định áp dụng các chính sách, chương trình khuyến mãi phù hợp để đẩy mạnh phát triển cửa hàng.

5.3.2. Câu lệnh SQL

- **Truy xuất danh sách khách hàng mua hàng lần đầu trong năm 2021 theo thứ tự thời gian từ sớm nhất đến trễ nhất.**

-- Hiển thị họ, tên đệm, tên, ngày cấp mã của các khách hàng được cấp mã trong năm 2021 theo thứ tự ngày tạo mã.

```
SELECT lname, minit, fname, signup_day
FROM Customer as Cus, Nomal_customer as NomalCUS
WHERE Cus.customer_id = NomalCUS.customer_id AND
      Cus.customer_id > 202100000 AND Cus.customer_id < 202200000
ORDER BY NomalCUS.signup_day;
GO
```

- **Truy xuất Tổng doanh thu (totalAmount) phân theo từng tỉnh được xếp theo thứ tự giảm dần từ các khách hàng VIP được cấp Mã khách hàng trong năm 2021.**

-- Hiển thị mã Zip_code, Tỉnh, Tổng doanh thu phân theo tỉnh đến từ khách hàng VIP và Số khách hàng Vip có địa chỉ tại từng tỉnh.

```
SELECT zip_code, city, SUM(totalAmount) AS N'Total revenue from City', COUNT(*) as N'No of Vip customer'
FROM Customer Cus, Vip_customer VipCUS
WHERE Cus.customer_id = VipCUS.customer_id
      AND Cus.customer_id > 202100000 AND Cus.customer_id < 202200000
GROUP BY city, zip_code
HAVING COUNT(*) >= 2
ORDER BY SUM(totalAmount) DESC;
GO
```

5.3.3. Kết quả thực thi câu lệnh SQL

- **Truy xuất danh sách khách hàng mua hàng lần đầu trong năm 2021 theo thứ tự thời gian từ sớm nhất đến trễ nhất.**

-- Execut và hiển thị kết quả câu truy vấn.

```
SELECT lname, minit, fname, signup_day
FROM Customer as Cus, Nomal_customer as NomalCUS
WHERE Cus.customer_id = NomalCUS.customer_id AND
      Cus.customer_id > 202100000 AND Cus.customer_id < 202200000
ORDER BY NomalCUS.signup_day;
GO
```

	lname	minit	fname	signup_day
1	Nguyen	Huu	Thang	2021-01-01
2	Nguyen	Trung	Kien	2021-01-02
3	Nguyen	Ngoc	Chau	2021-02-03
4	Nguyen	Thanh	Tuyen	2021-02-22
5	Tran	Duc	Tho	2021-04-11
6	Tran	Quoc	Tung	2021-04-19
7	Dinh	Quang	Quyet	2021-04-19
8	Nguyen	Huu	Thinh	2021-04-26
9	Truong	Duc	Tai	2021-05-10
10	Tran	Trong	Nhan	2021-05-23

- **Truy xuất Tổng doanh thu (totalAmount) phân theo từng tỉnh được xếp theo thứ tự giảm dần từ các khách hàng VIP được cấp Mã khách hàng trong năm 2021.**

-- Exec và hiển thị kết quả câu truy vấn.

```

SELECT zip_code, city, SUM(totalAmount) AS N'Total revenue from City', COUNT(*) as N'No of Vip customer'
FROM Customer Cus, Vip_customer Vipcus
WHERE Cus.customer_id = Vipcus.customer_id
      AND Cus.customer_id > 202100000 AND Cus.customer_id < 202200000
GROUP BY city, zip_code
HAVING COUNT(*) >= 2
ORDER BY SUM(totalAmount) DESC;
GO

```

	zip_code	city	Total revenue from City	No of Vip customer
1	700000	Ho Chi Minh	38000000.00	4
2	921490	Kien Giang	33000000.00	6

5.4. Hàm

5.4.1. Mô tả chức năng

- **Hàm xác định mã khách hàng từ một Email đầu vào**
 - Nhập tham số đầu vào là một Email
 - Hàm trả về kết quả là một Mã khách hàng
- **Hàm tính tuổi khách hàng từ một mã khách hàng cho trước**
 - Nhập tham số đầu vào là một Mã khách hàng
 - Hàm tính toán và trả về tuổi của khách hàng đó

5.4.2. Câu lệnh tạo hàm

- **Hàm xác định mã khách hàng từ một email đầu vào**

```

CREATE OR ALTER FUNCTION determine_customer_from_email (@customer_email VARCHAR(50))
RETURNS INT
AS
BEGIN
    IF EXISTS (SELECT * FROM Customer_contact WHERE cust_email = @customer_email)
    BEGIN
        DECLARE @determine_customer_id INT
        SELECT @determine_customer_id = (SELECT DISTINCT customer_id
                                         FROM Customer_contact
                                         WHERE @customer_email = cust_email )
        RETURN @determine_customer_id
    END
    RETURN N'Thông tin về Email này không nằm trong hệ thống'
END;
GO

```

➤ *Hàm tính tuổi khách hàng từ một mã khách hàng cho trước*

```
CREATE OR ALTER FUNCTION find_customer_years_old (@customer_id INT)
RETURNS INT
AS
BEGIN
    DECLARE @customer_years_old INT
    SELECT @customer_years_old = DATEDIFF(yyyy, Cus.birth_day, GETDATE())
    FROM Customer Cus
    WHERE @customer_id = Cus.customer_id
    RETURN @customer_years_old
END;
GO
```

5.4.3. Câu lệnh SELECT minh họa gọi hàm

➤ *Hàm xác định mã khách hàng từ một email đầu vào*

```
SELECT dbo.determine_customer_from_email ('thangnguyen141998@gmail.com');
GO
```

(No column name)	
1	202100001

➤ *Hàm tính tuổi khách hàng từ một mã khách hàng cho trước*

```
SELECT dbo.find_customer_years_old ('202100001');
GO
```

(No column name)	
1	20

5.5. Giao diện WEB

Customer [+ add customer](#)

ID	Last Name	Mid Name	First Name	Birth Date	Gender	Street	District	City	ZipCode	Total Amount	
202100001	Nguyen	Huu	Thang	Tue Dec 04 2001 00:00:00 GMT+0700 (Indochina Time)	M	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100002	Nguyen	Trung	Kien	Mon Sep 19 2005 00:00:00 GMT+0700 (Indochina Time)	M	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100003	Nguyen	Ngoc	Chau	Sun May 06 2001 00:00:00 GMT+0700 (Indochina Time)	F	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100004	Nguyen	Thanh	Tuyen	Sun Oct 30 1994 00:00:00 GMT+0700 (Indochina Time)	F	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100005	Tran	Duc	Tho	Sat May 19 2001 00:00:00 GMT+0700 (Indochina Time)	M	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100006	Tran	Quoc	Tung	Sat Sep 15 2001 00:00:00 GMT+0700 (Indochina Time)	M	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete
202100007	Nguyen	Huu	Thinh	Thu Jan 04 2001 00:00:00 GMT+0700 (Indochina Time)	M	3A	Tan Hiep	Kien Giang	921490	0	Edit Delete

Hình 5.5 a) Giao diện

[Customer](#) / [New Customer](#)

ID

Last Name

Mid Name

First Name

Birth Date

Gender

Street

District

City

ZipCode

Total Amount

[Submit](#)

Hình 5.5 b) Insert thông tin khách hàng mua hàng lần đầu.

[customer](#) / [Edit customer](#)

ID: 202100001

Last Name:

Mid Name:

First Name:

Birth Date:

Gender:

Street:

District:

City:

ZipCode:

Total Amount:

[Submit](#)

Hình 5.5 c) Chỉnh sửa thông tin khách hàng

7. PHỤ LỤC

7.1 Báo cáo bài tập lớn số 1

- Đã nộp

7.2 Source code chương trình

- File AS2.zip

7.3 Bảng phân công nhiệm vụ cho phần chung và bài tập lớn số 1

Tên	Trần Văn Cường	Phạm Văn Thành	Nguyễn Hoàn Thành	Nguyễn Trường Sơn	Nguyễn Minh Tùng
Hợp nhóm	Đầy đủ	Đầy đủ	Đầy đủ	Đầy đủ	Đầy đủ
Viết mô tả hệ thống	20%	20%	20%	20%	20%
Vẽ ERD	20%	20%	20%	20%	20%
Relational mapping	20%	20%	20%	20%	20%
Tạo bảng và ràng buộc	20%	20%	20%	20%	20%
Tạo dữ liệu	20%	20%	20%	20%	20%