

**Trường CD Công nghệ Thông tin – Đại học Đà Nẵng**  
**Khoa Công nghệ Thông tin**  
**--oo0oo--**

**BÀI TẬP THỰC HÀNH**  
**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

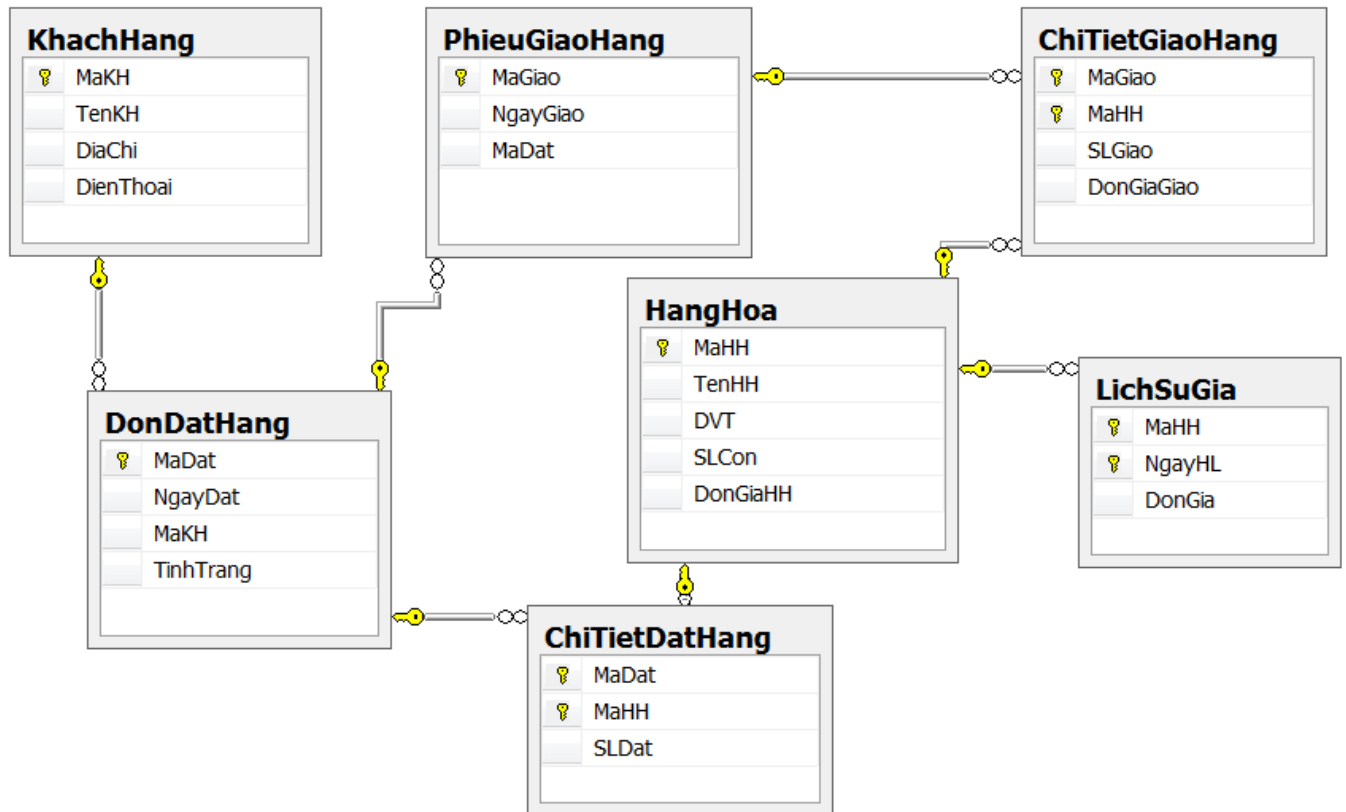
Môi trường thực hành: Microsoft SQL Server 2005 (hoặc phiên bản mới hơn)

Số tiết thực hành: 30 tiết

<b>PHÂN BỐ THỜI GIAN THỰC HÀNH TẠI LỚP</b>		
<b>Stt</b>	<b>Nội dung thực hành</b>	<b>Số tiết</b>
1	Database	4
2	Query	4
3	View + Cursor	4
4	Procedure	4
5	Function	4
6	Trigger	4
7	Xử lý truy xuất đồng thời	4
8	Kiểm tra thực hành	2
<b>Tổng Cộng</b>		<b>30</b>

❖ **Lưu ý:** Giáo viên hướng dẫn sẽ chấm bài 5 sinh viên bất kỳ vào cuối mỗi buổi thực hành

Cho cơ sở dữ liệu *Quản lý đơn đặt hàng* sau:



### Các qui định hoạt động của hệ thống:

- QĐ1.** Khi thêm mới một *đơn đặt hàng* thì *tình trạng* của đơn đặt hàng mặc định là chưa giao ( $TinhTrang=0$ ), khi đơn đặt hàng được giao thì *tình trạng* đơn đặt hàng được cập nhật lại là đã giao ( $TinhTrang=1$ )
- QĐ2.** Mỗi *đơn đặt hàng* chỉ có tối đa 1 *phiếu giao hàng* (cũng có những đơn đặt không được giao), *ngày giao hàng* phải bằng hoặc sau *ngày đặt hàng* nhưng không được quá 30 ngày.
- QĐ3.** *Số lượng giao* của một hàng hóa trong *chi tiết phiếu giao hàng* phải nhỏ hơn hoặc bằng *số lượng đặt* của *chi tiết đặt hàng* ứng với phiếu giao hàng đó. Khi cập nhật (thêm, xóa, sửa) một chi tiết phiếu giao hàng phải cập nhật lại *số lượng còn* (SLCon) của hàng hóa được giao.
- QĐ4.** *DonGiaHH* trong bảng *HangHoa* là đơn giá hiện hành, đơn giá này dùng để tham khảo khi giao hàng và được cập nhật theo *lịch sử giá* của hàng hóa đó. Chỉ được phép thêm (hay sửa) lịch sử giá của hàng hóa mà *ngày hiệu lực* của dòng dữ liệu được thêm (hay sửa) phải là lớn hơn so với tất cả các ngày hiệu lực còn lại của lịch sử giá ứng với hàng hóa đó.

# 1. DATABASE (CƠ SỞ DỮ LIỆU)

- Cài đặt CSDL Quản lý đơn đặt hàng với tên CSDL là QLDDH\_TenSinhVien, kết quả cuối cùng là có được diagram như hình trên. Lưu ý, trước khi tạo CSDL nên kiểm tra CSDL đã tồn tại chưa, nếu đã tồn tại rồi thì xóa CSDL đó đi rồi mới tạo.
- Thêm ràng buộc duy nhất (UNIQUE) cho trường TenHH trong bảng HangHoa, thử nhập dữ liệu để kiểm tra ràng buộc.
- Thêm ràng buộc kiểm tra (CHECK) cho trường SLCon, yêu cầu là trường này chỉ nhận giá trị  $\geq 0$ , thử nhập dữ liệu để kiểm tra ràng buộc.
- Thêm ràng buộc mặc định (DEFAULT) cho cột NgayDat trong DonDatHang với giá trị mặc định là ngày hiện tại, thử nhập dữ liệu để kiểm tra ràng buộc.
- Xóa bảng KHACHHANG? Nếu không xóa được thì nêu lý do? Muốn xóa được thì phải làm sao?
- Xóa cột DiaChi trong bảng KhachHang, sau đó tạo lại cột này với ràng buộc mặc định là 'Đà Nẵng'
- Xóa khóa ngoại MaDat trong PHIEUGIAHANG tham chiếu tới MaDat trong DonDatHang, sau đó tạo lại khóa ngoại này.
- Nhập dữ liệu cho các bảng như sau:

**HangHoa**

MaHH	TenHH	DVT	SLCon	DonGiaHH
BU	Bàn ỳ Philip	Cái	60	350000
CD	Nồi cơm điện Sharp	Cái	100	700000
DM	Đầu máy Sharp	Cái	75	1200000
MG	Máy giặt SanYo	Cái	10	4700000
MQ	Máy quạt ASIA	cái	40	400000
TL	Tủ lạnh Hitachi	Cái	50	5500000
TV	Tivi JVC 14WS	Cái	33	7800000

**KhachHang**

MaKH	TenKH	DiaChi	DienThoai
KH001	Cửa hàng Phú Lộc	Đà Nẵng	0511.3246135
KH002	Cửa hàng Hoàng Gia	Quảng Nam	0510.6333444
KH003	Nguyễn Lan Anh	Huế	0988.148248
KH004	Công ty TNHH An Phước	Đà Nẵng	0511.6987789
KH005	Huỳnh Ngọc Trung	Quảng Nam	0905.888555
KH006	Cửa hàng Trung Tín	Đà Nẵng	NULL

**LichSuGia**

MaHH	NgayHL	DonGia
BU	01-01-2011	300000
BU	01-01-2012	350000
CD	06-01-2011	650000
CD	01-01-2012	700000
DM	01-01-2011	1000000
DM	01-01-2012	1200000
MG	01-01-2011	4700000
MQ	06-01-2011	400000
TL	01-01-2011	5000000
TL	01-01-2012	5500000
TV	01-01-2012	7800000

**DonDatHang**

MaDat	NgayDat	MaKH	TinhTrang
DH01	02-02-2011	KH001	1
DH02	12-02-2011	KH003	1
DH03	22-01-2012	KH003	1
DH04	22-03-2012	KH002	0
DH05	14-04-2012	KH005	1
DH06	05-08-2012	KH003	1
DH07	25-11-2012	KH005	0

**PhieuGiaoHang**

MaGiao	NgayGiao	MaDat
GH01	02-02-2011	DH01
GH02	15-02-2011	DH02
GH03	23-01-2012	DH03
GH05	20-04-2012	DH05
GH06	05-08-2012	DH06

**ChiTietDatHang**

MaDat	MaHH	SLDat
DH01	BU	15
DH01	DM	10
DH01	TL	4
DH02	BU	20
DH02	TL	3
DH03	MG	8
DH04	TL	5
DH04	TV	5
DH05	BU	12
DH05	DM	15
DH05	MG	6
DH05	TL	5
DH06	BU	30
DH06	MG	7

**ChiTietGiaoHang**

MaGiao	MaHH	SLGiao	DonGiaGiao
GH01	BU	15	300000
GH01	DM	10	1000000
GH01	TL	4	5000000
GH02	BU	10	300000
GH03	MG	8	4700000
GH05	BU	12	350000
GH05	DM	15	1200000
GH05	MG	5	4700000
GH05	TL	5	5500000
GH06	BU	20	350000
GH06	MG	7	4700000

### Hướng dẫn:

Câu 1a. Tạo CSDL trước, tạo bảng sau:

```
-- Tạo CSDL
CREATE DATABASE QLDDH
GO -- Hết một lô

-- sử dụng CSDL đã tạo
USE QLDDH
GO

-- Tạo bảng HangHoa
CREATE TABLE HangHoa
(
    MaHH          char(5) ,
    TenHH          nvarchar(50) ,
    DVT            nvarchar(20) ,
    SLCon          smallint,
    DonGiaHH       int,
    Constraint pk_HH Primary Key (MaHH)
)
GO
--Tạo bảng DonDatHang
CREATE TABLE DonDatHang
(
    MaDat          char(10) ,
    NgayDat        smalldatetime,
    MaKH           char(5) ,
    TinhTrang      bit,
    Constraint pk_DDH Primary Key (MaDat)
)
GO
--Tạo bảng ChiTietDatHang
CREATE TABLE ChiTietDatHang
(
    MaDat          char(10) ,
    MaHH           char(5) ,
    SLDat          smallint,
    Constraint pk_CTDH Primary Key (MaDat, MaHH)
)
GO
--Tạo khóa ngoại giữa các bảng ChiTietDatHang với DonDatHang và HangHoa
ALTER TABLE ChiTietDatHang
ADD Constraint fk_CTDH_MaDat Foreign Key (MaDat) references DonDatHang (MaDat)
on update cascade on delete cascade,
Constraint fk_CTDH_MaHH Foreign Key (MaHH) references HangHoa (MaHH)
on update cascade on delete cascade

--Các bảng còn lại tạo tương tự
```

Câu 1b,c,d. Dùng lệnh ALTER TABLE để thêm các ràng buộc, đặt tên cụ thể cho các ràng buộc này.

Câu 1e. Dùng lệnh DROP TABLE để xóa bảng

Câu 1f,g. Dùng lệnh ALTER TABLE để xóa cột và xóa khóa ngoại.

Câu 1h. Dùng câu lệnh INSERT INTO để nhập dữ liệu

```
/* Lưu ý: dữ liệu kiểu chuỗi và ngày tháng phải đặt trong cặp dấu nháy đơn,
chuỗi có dấu phải có ký tự N đứng trước chuỗi, dữ liệu kiểu số thì không có
dấu cần dấu nháy đơn*/
INSERT INTO HangHoa
VALUES ('BU', N'Bàn ủi Philip', N'Cái', 60, 350000)

/* Trong SQL Server, ngày được định dạng khi nhập liệu là tháng/ngày/năm,
nếu muốn ngày định dạng là ngày/tháng/năm thì phải thực thi lệnh sau trước
khi thực hiện lệnh INSERT INTO*/
SET DATEFORMAT dmy
```

## 2. QUERY (TRUY VẤN)

- Cho biết chi tiết giao hàng của đơn đặt hàng DH01, hiển thị: tên hàng hóa, số lượng giao và đơn giá giao.
- Cho biết thông tin những đơn đặt hàng không được giao, hiển thị: mã đặt, ngày đặt, tên khách hàng.
- Cho biết hàng hóa nào có đơn giá hiện hành cao nhất, hiển thị: tên hàng hóa, đơn giá hiện hành.
- Cho biết số lần đặt hàng của từng khách hàng, những khách hàng không đặt hàng thì phải hiển thị số lần đặt hàng bằng 0. Hiển thị: Mã khách hàng, tên khách hàng, số lần đặt
- Cho biết tổng tiền của từng phiếu giao hàng trong năm 2012, hiển thị: mã giao, ngày giao, tổng tiền, với tổng tiền= SUM(SL Giao\*DonGiaGiao)
- Cho biết khách hàng nào có 2 lần đặt hàng trở lên, hiển thị: mã khách hàng, tên khách hàng, số lần đặt.
- Cho biết mặt hàng nào đã được giao với tổng số lượng giao nhiều nhất, hiển thị: mã hàng, tên hàng hóa, tổng số lượng đã giao.
- Tăng số lượng còn của mặt hàng có mã bắt đầu bằng ký tự 'M' lên 10.
- Copy dữ liệu bảng HangHoa sang một bảng HangHoa\_copy, sau đó xóa những mặt hàng chưa được đặt trong bảng HangHoa. Chèn lại vào bảng HangHoa những dòng bị xóa từ bảng HangHoa\_copy.
- Cập nhật số điện thoại cho khách hàng có mã KH006.
- Sửa giá trị TinhTrang=NULL cho tất cả các đơn đặt hàng
- Thêm cột ThanhTien cho bảng ChiTietGiaoHang, sau đó cập nhật giá trị cho cột này với ThanhTien = SLGiao\*DonGiaGiao

### Hướng dẫn:

/\*Câu 2a: dùng câu lệnh select gồm các mệnh đề SELECT, FROM, WHERE, trong đó mệnh đề FROM gồm 3 bảng: PhieuGiaoHang, ChiTietGiaoHang, HangHoa\*/

```
SELECT TenHH,SLGiao,DonGiaGiao
FROM (PhieuGiaoHang pg inner join ChiTietGiaoHang ctg
      on pg.MaGiao = ctg.MaGiao) inner join HangHoa hh
      on ctg.MaHH = hh.MaHH
WHERE MaDat='DH01'
```

/\*Câu 2b,2c: dùng câu SELECT lồng trong mệnh đề WHERE

Câu 2d, e: dùng câu SELECT có thêm mệnh đề GROUP BY

Câu 2f: dùng câu SELECT có thêm mệnh đề GROUP BY và HAVING

Câu 2g: dùng câu SELECT lồng với toán tử >=All trong mệnh đề HAVING

Câu 2h,j, k: dùng câu lệnh UPDATE

Câu 2i: dùng câu lệnh INSERT INTO + DELETE kết hợp với câu SELECT lồng trong mệnh đề WHERE

Câu 2k: dùng câu lệnh UPDATE

Câu 2l: dùng câu lệnh ALTER TABLE để thêm cột, sau đó dùng câu lệnh UPDATE để cập nhật giá trị.

\*/

### 3. VIEW ( KHUNG NHÌN)

- Tạo view thống kê doanh số giao hàng của từng mặt hàng trong 6 tháng đầu năm 2012
- Tạo view cho biết mặt hàng nào có tổng số lượng được đặt lớn nhất trong năm 2012
- Tạo view cho biết danh sách khách hàng ở Đà Nẵng có sử dụng WITH CHECK OPTION, sau đó chèn 2 khách hàng vào view này, một khách hàng có địa chỉ Đà Nẵng và một khách hàng có địa chỉ ở Quảng Nam, có nhận xét gì trong 2 trường hợp này?

**Hướng dẫn:**

```
//Câu 3a: Tạo view
CREATE VIEW vw_DoanhSoGiaoHang_6thang
as
    SELECT hh.MaHH,TenHH,SUM(SLGiao*DonGiaGiao) as TongTien
    FROM (PhieuGiaoHang pg inner join ChiTietGiaoHang ctg
         on pg.MaGiao = ctg.MaGiao) inner join HangHoa hh
         on ctg.MaHH = hh.MaHH
    WHERE Month(NgayGiao) between 1 and 6
    and YEAR(NgayGiao)=2012
    GROUP BY hh.MaHH,TenHH

-- Xem view vừa tạo
SELECT * FROM vw_DoanhSoBanHang_6thang
```

### 4. CURSOR (CON TRỎ)

- Thêm cột TongTien vào phiếu giao hàng, sau đó dùng con trỏ cập nhập giá trị cho cột TongTien, với TongTien=SUM(SLGiao\*DonGiaGiao) hay nói cách khác TongTien = SUM(ThanhTien)
- Thêm mới cột THUONG\_2012 vào bảng NHANVIEN để lưu giữ số tiền khách hàng được thưởng trong năm 2012. Dùng con trỏ để cập nhập giá trị cho cột này như sau:
  - Thưởng 3 triệu đối với khách mua hàng trên 50 triệu trong năm 2012
  - Thưởng 2 triệu đối với khách hàng mua hàng trên 35 triệu trong năm 2012 và có mua mặt hàng Máy giặt
  - Thưởng 1 triệu đối với những khách hàng có mua hàng trong cả 2 năm 2012 và 2011.
  - Tiền thưởng = 0 cho các trường hợp còn lại.
 Lưu ý là mỗi khách hàng chỉ nhận một mức tiền thưởng cao nhất.
- Vào ngày 1/1/2013, cần tăng giá của tất cả các mặt hàng lên 10% so với đơn giá hiện hành. Song song với việc tăng giá tất cả các mặt hàng là việc chèn 1 dòng dữ liệu vào LichSuGia ứng với mỗi hàng hóa, có nghĩa là có bao nhiêu hàng hóa sẽ có bấy nhiêu dòng dữ liệu được chèn vào bảng LichSuGia với các giá trị tương ứng. Dùng con trỏ để thực hiện công việc này.

**Hướng dẫn:**

Câu 4a. Thêm cột TongTien vào phiếu giao hàng, sau đó dùng con trỏ cập nhập giá trị cho cột TongTien này.

```
-- Thêm cột tổng tiền
ALTER TABLE PhieuGiaoHang
ADD TongTien money
GO

--Khai báo biến cursor, các biến cục bộ
DECLARE @magiao char(4), @tongtien int
DECLARE cur_PG CURSOR
FORWARD_ONLY
FOR
    SELECT MaGiao FROM PhieuGiaoHang
```

```
-- Mở cursor
OPEN cur_PG
-- Đọc dữ liệu và cập nhật giá trị
FETCH NEXT FROM cur_PG INTO @magiao
WHILE @@FETCH_STATUS = 0
BEGIN
    SELECT @tongtien = SUM(SLGiao*DonGiaGiao)
    FROM ChiTietGiaoHang
    WHERE MaGiao = @magiao

    PRINT 'Đang cập nhật phiếu giao:' + @magiao
    UPDATE PhieuGiaoHang
    SET TongTien = @tongtien
    WHERE MaGiao=@magiao -- Hoặc là: Where Current OF cur_PG

    FETCH NEXT FROM cur_PG INTO @magiao
END
-- Đóng và hủy cursor
CLOSE cur_PG
DEALLOCATE cur_PG
```

## 5. STORE PROCEDURE+TRANSACTION (Thủ tục nội tại + giao dịch)

- Tạo thủ tục truyền vào mã đơn đặt hàng (@maddh) và mã hàng hóa (@mahh), xuất ra số lượng hàng hóa @mahh được đặt trong đơn đặt hàng @maddh.
- Tạo thủ tục truyền vào mã phiếu giao hàng, xuất ra tổng tiền của phiếu giao hàng đó.
- Tạo thủ tục truyền vào mã khách hàng, hiển thị các đơn đặt hàng của khách hàng đó, gồm các thông tin: Mã đặt, ngày đặt, mã giao, ngày giao.
- Tạo thủ tục truyền vào ngày1 và ngày2, đếm xem có bao nhiêu phiếu giao hàng được giao trong khoảng thời gian từ ngày1 đến ngày2
- Viết lại câu 4a, 4b, 4c bằng cách dùng thủ tục.
- Tạo thủ tục thêm mới một hàng hóa với tham số đầu vào là: mã hàng, tên hàng, đơn vị tính, số lượng, đơn giá. Yêu cầu:
  - Kiểm tra khóa chính, nếu vi phạm thì báo lỗi và chấm dứt thủ tục.
  - Kiểm tra tên hàng phải là duy nhất (có nghĩa tên hàng nếu khác null phải khác với tất cả các tên hàng đã tồn tại trong bảng HangHoa), nếu không duy nhất thì báo lỗi và chấm dứt thủ tục.
  - Kiểm tra số lượng nếu khác null thì phải  $\geq 0$ , ngược lại thì báo lỗi và chấm dứt thủ tục.
  - Kiểm tra đơn giá nếu khác null thì phải  $\geq 0$ , ngược lại thì báo lỗi và chấm dứt thủ tục.
  - Nếu các điều kiện trên thỏa thì cho thêm mới hàng hóa.
- Tạo thủ tục thêm mới một ChiTietGiaoHang với các tham số đầu vào là: mã giao, mã hàng hóa, số lượng giao. Yêu cầu:
  - Kiểm tra hàng hóa này có được đặt không, có nghĩa mã hàng hóa truyền vào có tồn tại trong ChiTietDatHang của đơn đặt hàng tương ứng với phiếu giao hàng này không? Nếu không thì báo lỗi và chấm dứt procedure.
  - Kiểm tra số lượng giao có nhỏ hơn số lượng đặt ứng với hàng hóa này không? Nếu không thì báo lỗi và chấm dứt procedure.
  - Kiểm tra số lượng giao có nhỏ hơn số lượng còn của hàng hóa này không? Nếu không thì báo lỗi và chấm dứt procedure.
  - Nếu thỏa 3 điều kiện trên thì cho thêm mới vào chi tiết giao hàng, với đơn giá giao được lấy từ đơn giá hiện hành của hàng hóa này. Sau khi thêm mới phải cập nhật lại cột số lượng còn của HangHoa: SLCon= SLCon-SLGiao. Cần phải lưu ý với 2 hành động thêm

mới chi tiết giao hàng và cập nhật lại số lượng còn, nếu một trong hai hành động thất bại thì cả hai cùng thất bại. Cần phải sử dụng giao dịch (transaction) để giải quyết vấn đề này.

### Hướng dẫn:

5b. Tạo thủ tục truyền vào mã phiếu giao hàng, xuất ra tổng tiền của phiếu giao hàng đó

=> Đầu vào: mã phiếu giao, đầu ra: tổng tiền

```
--Tạo thủ tục cho câu 4b
CREATE PROC sp_TongTien_PhieuGiao
    @mapg char(10), @tongtien money OUTPUT
AS
BEGIN
    --Kiểm tra @mapg tồn tại chưa, nếu chưa tồn tại return 0
    IF NOT EXISTS(
        SELECT *
        FROM ChiTietGiaoHang
        WHERE MaGiao=@mapg )
        RETURN 0
    --Nếu @mapg truyền vào tồn tại thì return 1
    SELECT @tongtien= SUM(SLGiao * DonGiaGiao)
    FROM dbo.ChiTietGiaoHang
    WHERE MaGiao = @mapg
    RETURN 1
END

-- Gọi thủ tục vừa tạo
DECLARE @tt money
DECLARE @kq tinyint
EXEC @kq = sp_TongTien_PhieuGiao 'GH0001', @tt OUTPUT
IF @kq=0
    PRINT N'Mã giao hàng không tồn tại'
ELSE
    PRINT N'Tổng tiền là: ' + cast(@tt as nvarchar(20))
```

## 6. FUNCTION (HÀM DO NGƯỜI DÙNG ĐỊNH NGHĨA)

- Viết lại câu 5a bằng cách dùng Function
- Viết lại câu 5b bằng cách dùng Function
- Viết lại câu 5c bằng 2 hàm: hàm trả về Inline Table và hàm trả về Multi-statement Table
- Viết lại câu 5d bằng cách dùng Function

### Hướng dẫn:

6b. Tạo thủ tục truyền vào mã phiếu giao hàng, xuất ra tổng tiền của phiếu giao hàng đó

=> Đầu vào: mã phiếu giao, đầu ra: tổng tiền

```
--Tạo hàm cho câu 5b
CREATE FUNCTION udf_TongTien_PhieuGiao(@mapg char(10))
RETURNS money
AS
BEGIN
    DECLARE @tongtien money
    --Kiểm tra @mapg tồn tại chưa, nếu chưa tồn tại return -1
    IF NOT EXISTS(
        SELECT *
        FROM ChiTietGiaoHang
        WHERE MaGiao=@mapg )
        RETURN -1
    --Nếu @mapg truyền vào tồn tại thì return tổng tiền
    SELECT @tongtien= SUM(SLGiao * DonGiaGiao)
    FROM dbo.ChiTietGiaoHang
    WHERE MaGiao = @mapg
    RETURN @tongtien
END
```



```
-- Gọi hàm vừa tạo
DECLARE @tt money
SET @tt = dbo.udf_TongTien_PhieuGiao('GH01')
IF @tt=-1
    PRINT N'Mã giao hàng không tồn tại'
ELSE
    PRINT N'Tổng tiền là: ' + cast(@tt as nvarchar(20))
```

## 7. TRIGGER (RÀNG BUỘC TOÀN VỆN)

- Cài đặt ràng buộc sau bằng 2 cách: constraint và trigger  
“Số lượng còn của hàng hóa phải >0”
- Cài đặt ràng buộc sau bằng 2 cách: constraint và trigger  
“Đơn vị tính của hàng hóa chỉ nhận một trong các giá trị: Cái, Thùng, Chiếc, Chai, Lon”
- Cài đặt ràng buộc: “Mỗi đơn đặt hàng chỉ có tối đa 1 phiếu giao hàng”
- Cài đặt ràng buộc: “Ngày giao hàng phải bằng hoặc sau ngày đặt hàng nhưng không được quá 30 ngày”
- Tạo trigger sau khi chèn 1 dòng mới vào bảng LichSuGia (gồm: mã hàng hóa, ngày hiệu lực mới, đơn giá mới), nếu ngày có hiệu lực mới lớn hơn tất cả các ngày hiệu lực trong lịch sử giá của hàng hóa tương ứng thì cập nhật lại DonGiaHH bằng đơn giá mới cho hàng hóa này, ngược lại thì rollback.
- Cài đặt ràng buộc: “Số lượng hàng hóa được giao không được lớn hơn số lượng hàng hóa được đặt tương ứng”

### Hướng dẫn:

Câu 7d.

```
CREATE TRIGGER trg_NgayGiao_NgayDat
ON PhieuGiaoHang
AFTER INSERT, UPDATE
AS
    DECLARE @madat char(10), @ngaygiao datetime, @ngaydat datetime
    --Trường hợp thêm mới
    IF NOT EXISTS (SELECT * FROM deleted)
    BEGIN
        SELECT @madat = MaDat, @ngaygiao = NgayGiao
        FROM inserted

        SELECT @ngaydat = NgayDat
        FROM DonDatHang
        WHERE MaDat=@madat

        IF @ngaygiao<@ngaydat
        BEGIN
            RAISERROR (N'Ngày giao phải sau ngày đặt',16,1)
            ROLLBACK
            RETURN
        END
        IF DATEDIFF(DD, @ngaydat, @ngaygiao)> 30
        BEGIN
            RAISERROR (N'Ngày giao - ngày đặt <= 30 ngày',16,1)
            ROLLBACK
            RETURN
        END
    END
    ELSE
    -- Trường hợp sửa
    BEGIN
        IF UPDATE(NgayGiao)
```

```

BEGIN
    SELECT @madat = MaDat, @ngaygiao = NgayGiao
    FROM inserted

    SELECT @ngaydat = NgayDat
    FROM DonDatHang
    WHERE MaDat=@madat

    IF @ngaygiao<@ngaydat
    BEGIN
        RAISERROR (N'Ngày giao phải sau ngày đặt',16,1)
        ROLLBACK
        RETURN
    END
    IF DATEDIFF(DD, @ngaydat, @ngaygiao)> 30
    BEGIN
        RAISERROR (N'Ngày giao - ngày đặt <= 30 ngày',16,1)
        ROLLBACK
        RETURN
    END
END
END
END

```

## 8. AN TOÀN + BẢO MẬT DỮ LIỆU

- Hãy Export bảng KháchHang ra file có tên là tblKhachHang.txt
  - Xóa hết dữ liệu trong bảng KháchHang, sau đó Import lại dữ liệu từ file tblKhachHang.txt
  - Thực hiện Export toàn bộ các bảng của CSDL sang Access
  - Sao lưu (Backup) toàn bộ CSDL thành file QLDDH.bak, sau đó xóa và sửa một vài dòng bất kỳ trong CSDL rồi thực hiện phục hồi (restore) lại CSDL từ file QLDDH.bak, có nhận xét gì?
  - Giả sử cơ sở dữ liệu về Quản lý nhập xuất tồn có các nhóm, người sử dụng như sau:
    - Các account: Admin, Director có quyền quản trị.
    - Các user: user1, user2, user3 có quyền xem tất cả các bảng nhưng không có quyền thêm, xóa, sửa bất kỳ bảng nào.
    - Các user: user4, user5, user6 có quyền xem tất cả các bảng và quyền thêm, xóa, sửa bảng DonDatHang+ChiTietDatHang và PhieuGiaoHang+ChiTietGiaoHang.
- Hãy tạo các nhóm, role, user để đảm bảo quyền trên.

## 9. TRUY XUẤT DỮ LIỆU ĐỒNG THỜI

- Cho ví dụ về một trường hợp deadlock, kiểm tra xem deadlock này có xảy ra trong SQL Server không?
- So sánh mức cô lập Read Uncommitted và Read Committed

- Trường hợp 1:

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED UPDATE KháchHang SET TenKH= 'ABC' WHERE MaKH= 'KH001' WAITFOR DELAY '00:00:05'  ROLLBACK TRAN         </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED SELECT * FROM KháchHang WHERE TenKH= 'ABC' COMMIT TRAN         </pre>

Nhận xét kết quả đọc dữ liệu của T2?

- Trường hợp 2:

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ COMMITTED UPDATE KháchHang SET TenKH= 'ABC' WHERE MaKH= 'KH001' WAITFOR DELAY '00:00:05'  ROLLBACK TRAN         </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ COMMITTED SELECT * FROM KháchHang WHERE TenKH= 'ABC' COMMIT TRAN         </pre>

Nhận xét kết quả đọc dữ liệu của T2?

c. So sánh mức cô lập Read Committed và Repeatable Read

- Trường hợp 1:

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ COMMITTED SELECT TenKH FROM KhachHang WHERE MaKH = 'KH001' WAITFOR DELAY '00:00:05'  SELECT TenKH FROM KhachHang WHERE MaKH = 'KH001' COMMIT TRAN </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL READ COMMITTED UPDATE KhachHang SET TenKH= 'ABC' WHERE MaKH= 'KH001' COMMIT TRAN </pre>

**Nhận xét kết quả 2 lần đọc dữ liệu của T1?**

- Trường hợp 2:

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL REPEATABLE READ SELECT TenKH FROM KhachHang WHERE MaKH = 'KH001' WAITFOR DELAY '00:00:05'  SELECT TenKH FROM KhachHang WHERE MaKH = 'KH001' COMMIT TRAN </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL REPEATABLE READ UPDATE KhachHang SET TenKH= 'ABC' WHERE MaKH= 'KH001' COMMIT TRAN </pre>

**Nhận xét kết quả 2 lần đọc dữ liệu của T1?**

d. So sánh mức cô lập Repeatable Read và Serializable

- Trường hợp 1:

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL REPEATABLE READ SELECT * FROM HangHoa WHERE SLCon = 100 WAITFOR DELAY '00:00:05'  SELECT * FROM HangHoa WHERE SLCon = 100 COMMIT TRAN         </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL REPEATABLE READ INSERT INTO HangHoa VALUES ('IP', 'Ipad', 'Cái', 100, 10000000) COMMIT TRAN         </pre>

**Nhận xét kết quả 2 lần đọc dữ liệu của T1?**

- Trường hợp 2

T1	T2
<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL SERIALIZABLE SELECT * FROM HangHoa WHERE SLCon = 100 WAITFOR DELAY '00:00:05'  SELECT * FROM HangHoa WHERE SLCon = 100 COMMIT TRAN         </pre>	<pre> BEGIN TRAN SET TRANSACTION ISOLATION LEVEL SERIALIZABLE INSERT INTO HangHoa VALUES ('IP', 'Ipad', 'Cái', 100, 10000000) COMMIT TRAN         </pre>

**Nhận xét kết quả 2 lần đọc dữ liệu của T1?**

**(\*) Bài tập về xử lý truy xuất đồng thời:**

Viết một thủ tục (store procedure) thêm một phiếu giao hàng có chèn lệnh *waitfor delay* '00:00:10' vào trước lệnh insert. Sau đó giả lập 2 giao dịch cùng thực hiện stored procedure này, lần lượt với các mức cô lập: read uncommitted, read committed, repeatable read, serializable. Nhận xét về các vấn đề xảy ra?

Trở lại mức cô lập mặc định, hãy đặt lock trực tiếp vào các lệnh sao cho có thể giải quyết các vấn đề trên một cách hiệu quả nhất.