**• Stored-Procedure**  
**Định nghĩa:** Một Stored procedure được định nghĩa gồm những thành phần chính sau:  
- Tên của stored procedure  
- Các tham số  
- Thân của stored procedure: bao gồm các lệnh của Transact-SQL dùng để thực thi procedure. Một stored procedure được tạo bằng lệnh Create Procedure, và có thể thay đổi bằng cách dùng lệnh Alter Procedure, và có thể xóa bằng cách dùng lệnh Drop Procedure trong lập lệnh của Transact – SQL.  
**Lệnh tạo Procedure**:

*Lưu ý*:  
o Trong SQL Server, có thể ghi tắt một số từ khóa mà tên có chiều dài hơn 4 ký tự. Ví dụ: có thể thay thế Create Procedure bằng Create Proc; Có thể tạo, chỉnh sửa Store bằng giao diện.  
o Tên hàm, tên biến trong SQL Server không phân biệt hoa thường.  
**Khai báo biến và gán giá trị cho biến**:

**Biên dịch và gọi thực thi một stored-procedure:**

**Ví dụ:**  
- *Stored-procedure tính tổng của 2 số nguyên:*

- *Stored procedure liệt kê những thông tin của đầu sách, thông tin tựa sách và số lượng sách hiện chưa được mượn của một* *đầu sách cụ thể (ISBN).*

**B. Trigger**  
**Giới thiệu:** Trigger là một trường hợp đặc biệt của store procedure, nó sẽ có hiệu lực khi chúng ta thay đổi dữ liệu trên một bảng dữ liệu cụ thể, hoặc các xử lý làm thay đổi dữ liệu của các lệnh: insert, update, delete. Trigger có thể chứa các lệnh truy vấn từ các bảng khác hoặc bao gồm những lệnh SQL phức tạp.Chúng ta sẽ xây dựng một ví dụ đơn giản đầu tiên với một bảng và ví dụ liên quan đến lưu trữ, lấy dữ liệu, cập nhật và xóa các mẫu tin.  
**Lệnh tạo Trigger**:

**Ví dụ:**

**C. Function**  
**Giới thiệu:** Trong SQL Server ta có thể viết hàm và lấy giá trị trả về. Các dạng hàm có thể viết như sau :  
- Hàm trả về giá trị vô hướng (scalar value) : varchar, int, ….  
- Hàm trả về giá trị là bảng tạm (inline table-valued) : table  
**Cú pháp:**

**Ví dụ:**  
- *Viết hàm tính tuổi của người có năm sinh là @ns:*

- *Viết hàm tạo bảng tạm từ một câu truy vấn :*

Kiểm tra thử hàm:

**D. Luyện tập**

**1 . Bài tập về Stored-Procedure**

**1.1. Store-Prodedure đơn giản**

1) Viết stored-procedure In ra dòng ‘Hello’

2) Viết stored-procedure In ra dòng ‘Xin chào’.

3) Viết stored-procedure In ra dòng ‘Xin chào’ + @ten với @ten là tham số đầu vào là tên của bạn.

4) Viết stored-procedure In ra dòng ‘Xin chào’ + @ten với @ten là tham số đầu vào là tên Tiếng Việt có dấu của bạn.

Gợi ý :

♦ sử dụng UniKey để gõ Tiếng Việt

♦ chuỗi unicode phải bắt đầu bởi N (vd: N’Tiếng Việt’)

♦ dùng hàm cast ( as ) để đổi thành kiểu tương ứng

5) Viết stored-procedure Nhập vào 2 số @s1,@s2. In ra tổng @s1+@s2.

6) Viết stored-procedure Nhập vào 2 số @s1,@s2. In ra câu ‘Tổng là : @tg’ với @tg=@s1+@s2.

7) Viết stored-procedure Nhập vào 2 số @s1,@s2. Xuất tổng @s1+@s2 ra tham số @tong. Cho thực thi và in giá trị của tham số này để kiểm tra.

8) Viết stored-procedure Nhập vào 2 số @s1,@s2. Xuất tổng @s1+@s2 ra tham số @tong. Cho thực thi và in giá trị của tham số này để kiểm tra dưới dạng ‘Tổng là : @tg’ với @tg =@s1+@s2.

9) Viết stored-procedure Nhập vào 2 số @s1,@s2. In ra max của chúng.

10) Viết stored-procedure Nhập vào 2 số @s1,@s2. In ra câu ‘Số lớn nhất của @s1 và @s2 là @max’ với @s1,@s2,max là các giá trị tương ứng.

11) Viết stored-procedure Nhập vào 2 số @s1,@s2. Xuất min và max của chúng ra tham số @max, @min. Cho thực thi và in giá trị của các tham số này để kiểm tra.

**1.2. Store-Prodedure có vòng lặp**

12) Viết stored-procedure Nhập vào số nguyên @n. In ra các số từ 1 đến @n.

13) Viết stored-procedure Nhập vào số nguyên @n. In ra tổng các số chẵn từ 1 đến @n

14) Viết stored-procedure Nhập vào số nguyên @n. In ra tổng, và số lượng các số chẵn từ 1 đến @n

15) Viết stored-procedure Nhập vào 2 số. In ra ước chung lớn nhất của chúng theo gợi ý dưới đây.

b1. Không mất tính tổng quát giả sử a <= A

b2. Nếu A chia hết cho a thì : (a,A) = a ngược lại : (a,A) = (A%a,a) hoặc (a,A) = (a,A-a)

b3. Lặp lại b1,b2 cho đến khi điều kiện trong b2 được thỏa

16) Viết stored-procedure Nhập vào 2 số nguyên @s1,@s2. Xuất ước chung lớn nhất của @s1 và @s2 ra tham số @gcd. Cho thực thi và in bằng lệnh select giá trị của tham số này để kiểm tra dưới dạng ‘Kết quả : ucln(@s1,@s2) = @gcd’ trong đó thay thế @s1,@s2,@gcd bởi các giá trị tương ứng.

**1.3. Store-Prodedure có đệ quy**

17) Viết stored-procedure Cài đặt có dùng đệ quy, thuật toán Euler tìm ước chung lớn nhất (a,b).

(a, b) = (b, a%b)

(15, 12) = (12, 3) = (3, 0)

18) Viết stored-procedure Nhập vào 2 số. In ra ước chung lớn nhất của chúng. Bắt buộc viết bằng đệ quy.

19) Viết stored-procedure Nhập vào số nguyên @n <= 5. In ra tất cả các số nhị phân có @n bit.

Ví dụ : @n=2 thì kết quả in được là 00 01 10 11

**1.4 . Store-Prodedure nâng cao**

20) Xây dựng thủ tục in ra bảng cửu chương n với tên spud\_InBangCC gồm 1 tham số vào là: n. Với n là số nguyên và n thuộc khoảng [2,10]

21) Xây dựng thủ tục giải phương trình bậc nhất (ax + b = 0) với tên pud\_GiaiPTBac1 gồm 2 tham số vào là: a và b

**2. Bài tập về ràng buộc toàn vẹn - trigger**

Cho CSDL Quản lý câu lạc bộ như sau (Khóa chính in đậm)

Bảng đội bóng - DOIBONG

**Field Name**

**Field Type**

**Field Size**

**Decription**

**MaDoi**

varchar

2

Mã đội bóng

TenDoi

Varchar

100

Tên đội

NamTL

Number

Integer

Năm thành lập

Bảng cầu thủ - CAUTHU

**Field Name**

**Field Type**

**Field Size**

**Decription**

**MaCauThu**

Varchar

2

Mã Cầu thủ

TenCauThu

Varchar

50

Tên Cầu thủ

Phai

Bit

1 :Nam / 0: Nữ

NgaySinh

DateTime

NoiSinh

Varchar

50

Bảng thi đấu – THIDAU

**Field Name**

**Field Type**

**Field Size**

**Decription**

**MaDoi**

Varchar

2

NgayThiDau

DateTime

HieuSo

Int

lấy trị tuyệt đối của (số bàn thắng -số bàn thua)

KetQua

Bit

1 : Thắng/0 : Thua

Bảng phạt - PENELTY

**Field Name**

**Field Type**

**Field Size**

**Decription**

**MaPhat**

varchar

2

Mã phạt

MaCT

Varchar

2

Mã cầu thủ

TienPhat

Numeric

LoaiThe

varchar

1

D: đỏ, V: vàng

NgayPhat

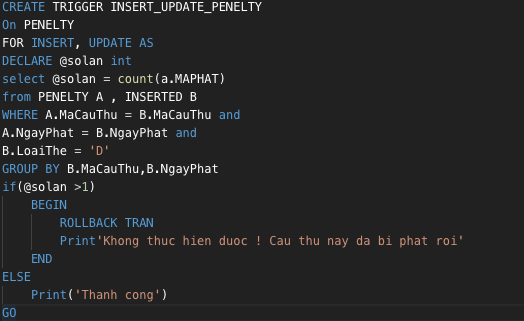
DateTime

Ngày phạt

**Cài đặt các ràng buộc trên các quan hệ (bảng)**

1) Loại thẻ phạt chỉ có thể là ‘D’ hoặc ‘V’;

2) Nếu loại thẻ là D (đỏ) thì số lần phạt cho mỗi cầu thủ trong một ngày chỉ tối đa bằng 1.



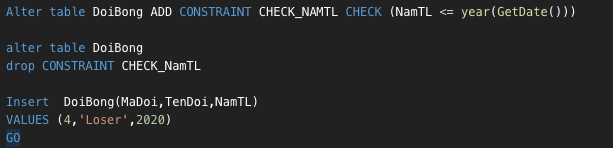
3) Tuổi của cầu thủ phải từ 17 đến 35. (Hướng dẫn: hàm Getdate(): lấy ngày hiện tại)



4) Mỗi lần bị phạt, ngày phạt cầu thủ phải lớn hơn ngày sinh của cầu thủ đó

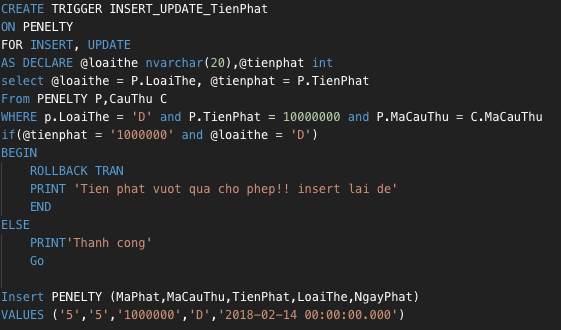


5) Năm thành lập đội bóng luôn nhỏ hơn hoặc bằng năm hiện tại. (Hướng dẫn: hàm lấy ngày hiện tại: Getdate( ) )



6) Năm gia nhập vào đội bóng của mỗi cầu thủ phải lớn hơn hoặc bằng năm thành lập của đội bóng đó.

7) Nếu LoaiThe là đỏ (‘D’) thì tiền phạt là 1000000.



8) Tuổi của cầu thủ (NgaySinh) khi tham gia vào đội bóng (NgayVaoCLB) phải lớn hơn hoặc bằng 18.

**3. Bài tập về function kết hợp Stored**

**Cho lược đồ quan hệ QuanLyBanHang:**

VatTu(MaVatTu, TenVatTu, DonViTinh, PhanTram)

NhaCungCap(MaNhaCungCap, TenNhaCungCap, DiaChi, DienThoai)

DonDH(sodh, ngaydh, manhacc)

ctdondh(sodh, mavtu, sldat)

pnhap(sopn, ngaynhap, sodh)

ctpnhap(sopn, mavtu, dgnhap)

pxuat(sopx, ngayxuat, tenkh)

ctpxuat(sopx, mavtu, slxuat, dgxuat)

tonkho(namthang, mavtu, sldau, slnhap, tongslxuat, slcuoi)

**Viết các function / stored thực hiện các công việc sau:**

• Xây dựng thủ tục tính số lượng hàng còn phải nhập với tên spud\_TinhSLConNhap gồm 2 tham sốvào là:Số đặt hàng và Mã vật tư.

• Xây dựng hàm tên Fn\_Giam\_DGX(@PhanTramGiam, @MucGiam) trả về danhsách các chi tiết xuất của các vật tư có số lượng xuất >= @MucGiam và cột đơn giá xuất đã giảm theo @PhanTramGiam. Côngthức:

Đơn giá giảm = Đơn giá - Đơn giá \* @PhanTramGiam/100

• Xây dựng hàm tên Fn\_LietKeDDH\_NCC(@Manhacc) trả về danh sách các đơn đặt hàng của mã nhà cung cấp truyền vào .

• Xây dựng thủ tục tính số lượng đặt hàng đã nhập với tên spud\_TinhSLNhap gồm 2 tham số vào là: Số đặt hàng và Mã vật tư.

• Xây dựng thủ tục tính số lượng đặt hàng với tên spud\_DONDH\_TinhSLDat gồm 2 tham số vào là: Số đặt hàng và Mã vật tư, 1 tham số ra là: Số lượng đặt hàng của vật tư có mã bằng với mã vật tư truyền vào và sốđặt hàng bằng với số đặt hàng truyền vào.

• Xây dựng thủ tục liệt kê các cột dữ liệu trong bảng TONKHO có thể hiện thêm cột TENVTU trong bảngVATTU với tên spud\_TONKHO\_BcaoTonkho gồm có 1 tham số vào là: Năm tháng muốn lọc dữ liệu.

• Xây dựng thủ tục tên spud\_HienThi\_DDH gồm 1 tham sốvào là: Năm tháng (dạng yyyymm). Thủ tục dung để hiển thị danh sách các thông tin trong bảng DONDH có lọc theo năm tháng truyền vào và có thêm cột ghi chú biết rằng nếu đã có nhập hàng thì in ra “Có x phiếu nhập hàng”, ngược lại nếu chưa có nhập hàng thì in ra “Chưa có nhập hàng”.

• Xây dựng thủ tục tính số lượng đặt hàng với tên spud\_TinhSLDat gồm 2 tham số vào là: Số đặt hàng và Mã vật tư.

• Xây dựng thủ tục tên spud\_TinhTienNhap\_200901 để tính tổng thành tiền nhập của các phiếu nhập trongtháng 1/2009.

• Xây dựng hàm tên Fn\_LietKePN(@Sodh) trả về danh sách các phiếu nhập của số đặt hàng truyền vào.

11) Xây dựng hàm tên Fn\_CacSoNT(@n) trả về chuỗi các số nguyên tố nằm trong khoảng từ 2 đến n.

Ví dụ:--In ra các sốnguyên tố từ 2 đến 10

DECLARE @Kq VARCHAR(1000)SET @Kq = dbo.Fn\_CacSoNT(10)PRINT @Kq --In ra 2, 3, 5, 7

12) Xây dựng hàm tên Fn\_SoNT(@n) trả về 1 nếu @n là sốnguyên tố, ngược lại thì trả về 0.

13) Xây dựng hàm tên Fn\_TongDaySo(@n) trả về tổng của các số nguyên từ

1 đến n

14) Xây dựng hàm tên Fn\_TongHaiSo(@So1, @So2) trả về tổng của hai số nguyên.

Ví dụ:--In ra tổng của hai số nguyên 10 và 15

DECLARE @Kq INTSET @Kq = dbo.Fn\_TongHaiSo(10,15)

PRINT @Kq --In ra 25

/Users/nam/Desktop/SQL\_Server