**Bài thực hành 7: HÀM VÀ THỦ TỤC**

**I/ LÝ THUYẾT:**

**1/ Thủ tục lưu trữ:**

**+ Định nghĩa:** Thủ tục lưu trữ (Procedure): là một đối tượng trong CSDL gồm tập nhiều lệnh SQL được nhóm lại thành một nhóm và các lệnh này sẽ được thực hiện khi thủ tục lưu trữ được thực thi.

Thủ tục lưu trữ có thể có các thành phần:

* Cấu trúc điều khiển (IF, WHILE, FOR)
* Biến để lưu các giá trị tính toán, các giá trị truy xuất từ CSDL
* Các câu lệnh SQL được kết hợp thành khối lệnh trong thủ tục, một thủ tục có thể có tham số truyền vào hay giá trị trả về giống như ngôn ngữ lập trình thông thường.

**+ Ưu điểm của thủ tục:**

* Đơn giản hóa thao tác dữ liệu do tính module hóa thao tác
* Việc thực thi nhanh hơn so với thực hiện rời rạc các lệnh SQL thông thường
* Giảm lưu thông trên mạng: Thực hiện một yêu cầu bằng một câu lệnh đơn giản thay vì sử dụng nhiều dòng SQL
* Tăng bảo mật CSDL do việc cấp phát quyền trên thủ tục lưu thay vì tác động trực tiếp đến cơ sở dữ liệu
* Tập trung tại Server nên dễ quản lý
* Được biên dịch một lần và sử dụng lại kết quả trong các lần tiếp theo

**+ Cú pháp tạo thủ tục:**

**CREATE PROC** *name-proc*[( *parameters* )]

*[with* **RECOMPILE| ENCRYPTION | RECOMPILE, ENCRYPTION** ]

**AS**

**Begin**

*các câu lệnh của thủ tục*

**End**

Trong đó:

* **Name-proc**: tên thủ tục cần tạo ra, tuân thủ nguyên tắc định danh, không quá 128 ký tự
* **Parameters**: các tham số truyền vào để thủ tục thực hiện
  + Các tham số cách nhau bởi dấu phẩy
  + Cú pháp: ***@***tên-tham-số ***kiểu-tham-số***
  + VD: @mamonhoc nvarchar(10)
* **RECOMPILE:** cho phép dịch lại mỗi khi được gọi
* **ENCRYPTION**: cho phép mã hóa thủ tục => không xem được nội dung của thủ tục nữa

**Ví dụ 1:** Tạo thủ tục xem thông tin của sinh viên: mã sinh viên, tên sinh viên, ngày sinh, tên môn học và điểm

create proc sp\_Sv

as

select a.Masv, Tensv, Ngaysinh, Tenmh, Diem

from sinhvien a, ketqua b, monhoc c

where (a.masv=b.masv) and (b.mamh=c.mamh)

**Ví dụ 2:** Tạo thủ tục cho biết thông tin của các sinh viên thuộc lớp bất kỳ

create proc Sp\_SV2 (@Lop nvarchar(6))

as

Select \*

from sinhvien

where Lop=@Lop

**Ví dụ 3:** Tạo thủ tục chèn dữ liệu vào hai bảng sinhvien và Monhoc

create proc Sp\_Insert(@Ten nvarchar(30), @NS date, @GT nvarchar(3), @que nvarchar(30),@MH nvarchar(30), @dvht int)

as

Insert into sinhvien(TenSV,Ngaysinh,GT,Que)

values(@Ten,@NS,@GT,@que)

Insert into monhoc(tenMH,DVHT)

values(@Mh,@dvht)

**+ Thực thi thủ tục:**

* Thực thi lời gọi thủ tục có dạng

*Tên\_thủ\_tục [danh\_sách\_đối\_số]*

*Chú ý:* Số lượng đối số và thứ tự phải phù hợp với số lượng và thứ tự của tham số khi định nghĩa

Khi không quan tâm đến thứ tự của các đối thì có thể dùng lời gọi sau:

*Tên\_thủ\_tục @tên\_tham\_số = giá\_trị*

* Lời gọi thủ tục được thực hiện bên trong một thủ tục khác, bên trong trigger hay với các câu lệnh SQL khác ta thực thi như sau:

*execute tên\_thủ\_tục [danh\_sách\_các\_đối\_số]*

**Ví dụ:** Lời gọi thực thi các thủ tục ở trên

--Lời gọi không có đối số

sp\_Sv

--Lời gọi có đối số

Sp\_Sv2 'L01'

Sp\_Insert N'B','1996/3/3',N'Nữ',N'Phú Thọ',N'Toán Rời rạc',3

--Lời gọi gán giá trị cho tham số không theo thứ tự

Sp\_Insert @Mh=N'Tin học văn phòng',@dvht=4,@Ten=N'C',@GT=N'Nam',@NS='1995/2/2',@que=N'Nghệ An'

**+ Sử dụng biến trong thủ tục:** Sử dụng các biến nhằm lưu giá trị tính toán được hoặc truy xuất được từ cơ sở dữ liệu. Khai báo biến bằng từ khóa DECLARE:

DECLARE @Tên\_biến Kiểu\_dữ\_liệu

**Ví dụ 1:** Kiểm tra 2 sinhvien có cùng năm sinh hay không

create procedure kiemtra\_thongtinsv

@masv1 int,@masv2 int

AS

declare @namsinh1 int, @namsinh2 int

select @namsinh1 = year(Ngaysinh) from Sinhvien where MaSV = @masv1

select @namsinh2 = year(Ngaysinh) from Sinhvien where MaSV = @masv2

if @namsinh1 <>@namsinh2

print N'hai bạn sinh viên mã'+ str(@masv1)+' và '+str(@masv2)+ N' không cùng ngày sinh'

else

print N'hai bạn sinh viên mã'+ str(@masv1)+' và '+str(@masv2)+ N' cùng ngày sinh'

kiemtra\_thongtinsv 1, 2

**Ví dụ 2:** Viết thủ tục tìm điểm cao nhất của môn CSDL

???

**+ Sửa thủ tục:** Khi một thủ tục được tạo ra, ta có thể tiến hành định nghĩa lại thủ tục đó bằng câu lệnh có cú pháp sau:

**ALTER PROCEDURE** tên\_thủ\_tục

[(danh sách tham số)]

**[WITH RECOMPILE|ENCRYPTION| RECOMPILE,ENCRIPTION]**

**As**

Các\_câu\_lệnh\_của\_thủ\_tục

**+ Xóa thủ tục**: Để xóa một thủ tục đã có ta sử dụng câu lệnh có cú pháp sau:

**DROP PROCEDURE** Tên\_thủ\_tục

**2. Hàm:**

+ **Khái niệm:** Hàm là đối tượng trong cơ sở dữ liệu, Hàm trả về một giá trị thông qua tên hàm. Có thể sử dụng hàm như là một thành phần của biểu thức.

**+ Cú pháp:**

**CREATE FUNCTION** tên\_hàm ([danh\_sách\_tham\_số])

**RETURNS** (kiểu\_trả\_về\_của\_hàm)

**AS**

**BEGIN**

các\_câu\_lệnh\_của\_hàm

return <giá trị trả về>

**END**

**Ví dụ :** Định nghĩa hàm tính ngày trong tuần của một giá trị kiểu ngày

\* Sử dụng hàm DatePart(N,date): date là ngày truyền vào để lấy các phần tương ứng, n là 1 hoặc 2 chữ cái viết tắt đại diện cho một phần của tham số date. N có thể có các giá trị sau:

* yy: năm của date
* qq, q: quý của date
* mm, m: tháng của date
* dy, y: ngày date là ngày thứ mấy của năm
* dd,d: Lấy phần ngày của date
* wk, w: date thuộc tuần thứ mấy trong năm
* dw: là ngày thứ mấy trong tuần, chú ý rằng 1 chính là ngày Chủ nhật.

Create Function Thu(@ngay Datetime)

returns Nvarchar(10)

as

Begin

declare @st Nvarchar(10)

Select @st=case Datepart(dw,@ngay)

when 1 then N'Chủ Nhật'

when 2 then N'Thứ Hai'

when 3 then N'Thứ Ba'

when 4 then N'Thứ Tư'

when 5 then N'Thứ Năm'

when 6 then N'Thứ Sáu'

when 7 then N'Thứ Bảy'

end

return (@st) --Trả về giá trị cho hàm

end

**+ Sử dụng hàm:** Hàm có thể được sử dụng trong thân một hàm khác, trong thủ tục hoặc trong câu lệnh select theo cú pháp sau:

**Dbo.<tên hàm>(danh sách tham số thực sự)**

**Ví dụ:** với hàm Thu ở trên, ta có thể sử dụng trong câu lệnh select để lấy ra thứ của ngày sinh như sau:

select Masv, tensv, Gt, dbo.thu(ngaysinh) as 'Ngày trong tuần', Lop

from sinhvien

+ Hàm với giá trị trả về là kiểu bảng: Hàm cũng có thể trả về giá trị là bảng nhằm tăng thêm tính linh hoạt của khung nhìn.

**Create Function** <tên\_hàm>([Danh\_sách\_tham\_số])

**Returns Table**

**As**

**Return** <Câu\_lệnh\_select>

**Chú ý:** Trong thân hàm chỉ có một câu lệnh select

**Ví dụ 1:** Hàm hiển thị thông tin về sinh viên theo lớp, thông tin gồm: Mã sinh viên, Tên sinh viên, ngày sinh, giới tính, điểm

create function diem\_lop(@lop nvarchar(10))

returns Table

as

return

(select a.Masv, Tensv, Ngaysinh, GT, Tenmh, Diem

from (sinhvien a join ketqua b on a.masv=b.masv) join monhoc c on b.mamh=c.mamh

where lop=@lop

)

--Thực thi lời gọi hàm

select \* from dbo.diem\_lop('L01')

Nếu có nhiều lệnh, ta dùng cú pháp sau

**Create Function** <tên\_hàm>[(danh\_sách\_tham\_số)]

**Returns** @Biến Table (định\_nghĩa\_bảng)

**As**

**Begin**

<các lệnh trong thân hàm>

**return**

**end**

**Ví dụ 2**: Viết hàm tạo một bảng thống kê số lượng sinh viên theo từng lớp tham gia thi môn học được nhập vào từ bàn phím. Bảng có hai cột: cột tên lớp và số lượng sinh viên.

Create Function tao\_bang\_TK (@MMH int)

Returns @Bang\_TK Table (Lop nvarchar(10), SL int)—Bảng có hai cột Lop và SL

As

Begin

Insert into @Bang\_TK

select Lop, count(a.Masv) as SL

from (sinhvien a join ketqua b on a.masv=b.masv) join monhoc c on b.mamh=c.mamh

where b.mamh=@mmh

group by Lop

return

end

--Thực thi lời gọi hàm

select \* from dbo.Tao\_bang\_TK(1)

**+ Sửa và xóa hàm**

* Sửa hàm:

ALTER FUNCTION tên\_hàm ([danh\_sách\_tham\_số])

RETURNS (kiểu\_trả\_về\_của\_hàm)

AS

BEGIN

các\_câu\_lệnh\_của\_hàm

END

* Xóa hàm:

DROP FUNCTION tên\_hàm

**II/Bài tập:**

**Bài 1**: Cho CSDL Sinh viên, thực hiện các yêu cầu sau:

1. Viết một thủ tục đưa ra các sinh viên có năm sinh bằng với năm sinh được nhập vào (lấy năm sinh bằng hàm datepart(yyyy,ngaysinh))
2. So sánh 2 sinh viên có mã được nhập vào xem sinh viên nào được sinh trước.
3. Viết một hàm đưa ra tháng sinh. Áp dụng để đưa ra tháng sinh các bạn sinh viên đã thi môn có mã là 1.

**Bài 2**: Cho CSDL gồm các bảng sau:

**NHANVIEN**(MaNV, Hoten, DiaChi, SDT, NgaySinh, GT, HSL)

**HANG**(MaHang, TenHang, NhaSX, TGianBaoHanh)

**KHACHHANG**(MaKH, TenKH, CMT, DiaChi, SoDienThoai, Email)

**HOADONXUAT**(MaHD, MaKH, NgayLapHD, MaNV, PhuongThucTT)

**CT**\_**HOADON**(MaHD, MaHang, SoLuongMua, DonGia)

Sinh viên tự thiết kế kiểu dữ liêu, tạo liên kết cho các bảng và nhập vào mỗi bảng 5 bản ghi.

Viết các hàm và thủ tục để:

* Tính tổng tiền đã mua hàng của một khách hàng nào đó theo mã KH
* Cho biết tổng số tiền hàng đã mua của một hóa đơn nào đó
* Cho biết tổng số tiền hàng đã bán của một tháng nào đó.
* Cho biết họ tên của nhân viên có tuổi cao nhất