

STORE PROCEDURE

(THỦ TỤC)

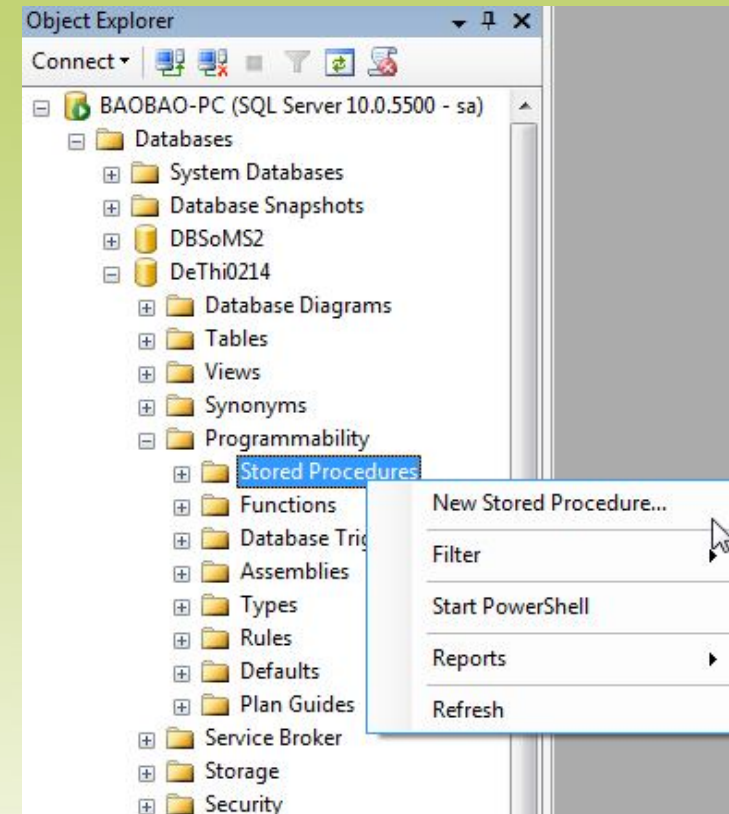
Khái niệm

❑ **Stored Procedure** là một đối tượng được xây dựng bởi những phát biểu T-SQL, và được lưu trữ trong CSDL với một tên phân biệt. Mỗi Stored Procedure có thể chứa nhiều câu lệnh SQL.

- ✓ Có thể **nhận vào các tham số**
- ✓ Có thể **trả về các giá trị** sau khi được tính toán trong thủ tục
- ✓ Được **gọi thực hiện từ các ngôn ngữ lập trình**
- ✓ Khi muốn thực thi các Stored Procedure đã lưu trữ ta **chỉ cần gọi tên** Stored Procedure (cung cấp tham số nếu có).

Tạo thủ tục

1. Chọn Database cần tạo thủ tục
2. Mở rộng mục Programmability
3. Right click vào thư mục Stored Procedures
4. Chọn New Stored Procedure



Tạo thủ tục

✓ Cú pháp:

Create Proc[cedure] <sp_name>

[<khai báo các tham số>]

[With Recompile|Encryption|Recompile,Encryption]

As

[Begin]

[Danh sách biến cục bộ]

<Các lệnh T-SQL>

[End]

Ví dụ tạo thủ tục

- ✓ Ví dụ: Cho biết thông tin sinh viên có mã số '001'
- *Lệnh SQL*: **Select** * **From** SinhVien **Where** MaSV = '001'
- *Bằng store procedure*

Create Proc spThongTinSV

As

Select * **From** SinhVien **Where** MaSV = '001'

Go

Exec <Store Procedure Name>

- ✓ Thực thi thủ tục vừa tạo: **Exec** spThongTinSV

Cho biết thông tin của sinh viên có mã số bất kỳ?

Khai báo tham số cho thủ tục

❑ Khai báo tham số: *@par_name DataType* dùng để truyền dữ liệu vào thủ tục.

✓ Cho biết thông tin sinh viên có mã số bất kỳ

– *Store procedure được viết:*

Create Proc spThongTinSV

@masosv Char(10)

As

Select * From SinhVien Where MaSV = *@masosv*

Go

Xem thông tin SV mã số '007'

✓ Thực thi thủ tục vừa tạo:

– Xem thông tin sinh viên mã số '007': *Exec* spThongTinSV '007'

– Xem thông tin sinh viên mã số 'abc': *Exec* spThongTinSV 'abc'

Xem thông tin SV mã số 'abc'

Ví dụ khai báo tham số cho thủ tục

- ✓ Ví dụ: Viết thủ tục cho biết danh sách sinh viên của một Khóa học và Ngành học bất kỳ do người dùng cung cấp.

Create Proc spDSSV_KhoaNganh

@khoahoc Char(10),

@nganhhoc Char(7)

As

Select * From SinhVien

Where KhoaHoc = @khoahoc And MaNganh = @nganhhoc

Go

- ✓ Thực thi store:

Exec spDSSV_KhoaNganh '01','TINHOC'

Exec spDSSV_KhoaNganh '03','KETOAN'

BAOBAO-PC.QLDaoTao - dbo.SinhVien* SQLQuery1.sql -		
Column Name	Data Type	Allow Nulls
MaSV	char(10)	<input type="checkbox"/>
HoSV	nvarchar(30)	<input checked="" type="checkbox"/>
TenSV	nvarchar(20)	<input type="checkbox"/>
PhaiNu	bit	<input type="checkbox"/>
NgaySinh	varchar(15)	<input type="checkbox"/>
MaTinh	char(2)	<input type="checkbox"/>
MaNganh	char(7)	<input type="checkbox"/>
KhoaHoc	char(10)	<input type="checkbox"/>

DS SV khóa 01 ngành tin học

DS SV khóa 03 ngành kế toán

Khai báo tham số với giá trị mặc định cho thủ tục

- ❑ Khai báo tham số với giá trị mặc định: *@par_name DataType = Value*
Khi thực thi thủ tục nếu không truyền giá trị cho tham số thì tham số sẽ nhận giá trị mặc định.

Ví dụ: Viết thủ tục cho biết danh sách sinh viên của một Khóa học và Ngành học bất kỳ do người dùng cung cấp, nếu người dùng không cung cấp thì xem danh sách sinh viên khóa '12', ngành kỹ thuật phần mềm ('KTPM')

Create Proc spDSSV_KhoaNganh

@khoahoc Char(10) = '12',

@nganhhoc Char(7) = 'KTPM'

As

Select * From SinhVien

Where KhoaHoc = @khoahoc And MaNganh = @nganhhoc

Go

✓ Thực thi store:

Exec spDSSV_KhoaNganh (giá trị mặc định của tham số được sử dụng)

Exec spDSSV_KhoaNganh @khoahoc = Default, @nganhhoc = 'HTTT'

BAOBAO-PC.QLDaoTao - dbo.SinhVien* SQLQuery1.sql -			
	Column Name	Data Type	Allow Nulls
	MaSV	char(10)	<input type="checkbox"/>
	HoSV	nvarchar(30)	<input checked="" type="checkbox"/>
	TenSV	nvarchar(20)	<input type="checkbox"/>
	PhaiNu	bit	<input type="checkbox"/>
	NgaySinh	varchar(15)	<input type="checkbox"/>
	MaTinh	char(2)	<input type="checkbox"/>
	MaNganh	char(7)	<input type="checkbox"/>
▶	KhoaHoc	char(10)	<input type="checkbox"/>

Cách truyền tham số khi gọi thực thi thủ tục

❑ Cách truyền tham số cho thủ tục:

Create Proc spViDu

@a int,

@b char(7),

@c float

As

.....

Go

✓ Thực thi store:

Exec spViDu 2014, '12SE111', 9.5

TRUYỀN THEO VỊ TRÍ THAM SỐ

(phải đúng theo thứ tự đã khai báo ở thủ tục)

- @a = 2014
- @b=12Se111
- @c=9.5

TRUYỀN THEO TÊN CỦA THAM SỐ

(không quan trọng về thứ tự)

Exec spViDu @c=9.5, @a=2014, @b='12SE111'

Cách khai báo tham số có thể nhận về giá trị sau khi thực thi thủ tục

❑ Khai báo tham số ra **output**: *@par_name DataType Output* vừa có thể truyền dữ liệu vào, vừa có thể lấy giá trị ra từ thủ tục.

✓ Ví dụ: Tạo thủ tục trả về Doanh thu bán hàng của một năm được truyền vào.

Create Proc spDTNam

@nam Smallint,

@doanhthu Float Output

AS

Select @doanhthu=Sum(SL*DonGia) From HoaDon Inner Join CTHD

On HoaDon.MaHD=CTHD.MaHD Where Year(NgayLapHD) = @nam

Go

✓ Thực thi thủ tục:

Khai báo biến

Declare @dt Float
Exec spDTNam 2005, @dt Output
Print @dt

Declare @dt Float
Exec spDTNam @doanhthu = @dt Output, @nam =2005
Print @dt

Thực thi thủ tục với tham số OUTPUT

```
Declare @dt Float  
Exec spDTNam 2005, @dt Output  
Print @dt
```

```
Declare @dt Float  
Exec spDTNam @doanhthu = @dt Output, @nam =2005  
Print @dt
```

❑ Để nhận giá trị của tham số **Output**:

- Đầu tiên, phải khai báo một biến (cùng kiểu dữ liệu với tham số **Output**)
- Sau đó gọi thực thi thủ tục.

Sửa, xóa thủ tục

❑ Đổi tên thủ tục

`sp_Rename <OldName>, <NewName>`

❑ Thay đổi nội dung thủ tục

`ALTER Proc[cedure] <sp_name>`

`[<khai báo các tham số>]`

`[With Recompile|Encryption|Recompile,Encryption]`

`As`

`[Begin]`

`[Danh sách biến cục bộ]`

`<Các lệnh T-SQL>`

`[End]`

❑ Xóa thủ tục

`DROP PROC <spName>`

Xem thông tin thủ tục

- ❑ Sys.Procedures

Select * From Sys.Procedures

- ❑ Sys. Parameters: chứa các tham số của thủ tục

Select * From Sys.Parameters Where Object_ID=Object_ID ('spDTNam')

- ❑ Xem nội dung thủ tục: SP_HELPTEXT 'spName'

- ❑ Kiểm tra thủ tục có tồn tại

If Exists (Select * From Sys.Procedures Where Name = 'spTonKho')

Hoặc

IF Object_ID('spDTNam') IS NOT NULL

Biến – Các phát biểu điều khiển

1. Biến cục bộ
2. Phát biểu **Return** [Biểu thức số nguyên]
3. Phát biểu **If .. Else**
4. Phát biểu **While**
5. Hàm **Case ... When**
6. Phát biểu : **Exec**(@string_var | [N]'string_expression')

Biến cục bộ

❑ Biến cục bộ

- Khai báo: **DECLARE** @varname *data_type* [...*n*]
 - Tất cả các biến cục bộ phải bắt đầu bằng ký hiệu @ trước tên của nó.
 - Biến khai báo trong một stored procedure hoặc một gói lệnh chỉ khả dụng bên trong vùng đã khai báo nó.
 - Sau khi khai báo, biến có giá trị **NULL**, trong T-SQL không có khái niệm biến toàn cục
 - Gán giá trị cho biến:
 - Dùng lệnh **SET** @varname = <Expression> (Ví dụ: **Set** @a = 10)
 - Dùng lệnh **SELECT** @varname = <Expression> [**FROM** ...]
(Nếu Select trả về nhiều dòng, thì biến được gán giá trị từ dòng cuối cùng trả về).
- Ví dụ: **Select** @dgm = (Select **Max**(DG) **From** SanPham) hoặc
Select @dgm = **Max**(DG) **From** SanPham

Ví dụ về biến cục bộ

- ❑ Ví dụ tạo thủ tục liệt kê danh sách mặt hàng có đơn giá cao nhất, thấp nhất.

```
Create Proc spMHMaxMin
```

```
AS
```

```
Begin
```

```
    Declare @dgmax float, @dgmin float
```

```
    SELECT @dgmax = Max(DonGia), @dgmin = Min(DonGia) FROM MatHang
```

```
    SELECT * FROM MatHang WHERE DonGia=@dgmax Or DonGia=@dgmin
```

```
End
```


Phát biểu Return

- ❑ Phát biểu **Return** dùng kết thúc thủ tục và trả về nơi gọi thủ tục **một số nguyên** nào đó, mặc định là 0

- ❑ Ví dụ: **Create Proc** spPhepCong

@a int,

@b int

As

Declare @kq int

Set @kq = @a+@b

Return @kq

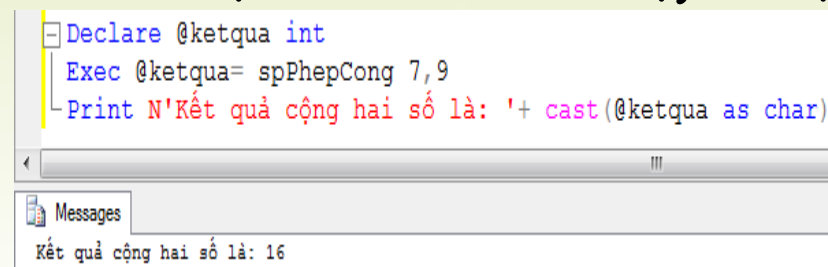
Go

- ❑ **Đề nhận giá trị Return:** đầu tiên phải khai báo một biến và sau đó chạy thủ tục.

Declare @ketqua int

EXEC @ketqua = spPhepCong 7, 9

Select @ketqua



The screenshot shows a SQL Server Enterprise Manager interface. The top window is a query editor with the following T-SQL code:

```
Declare @ketqua int
Exec @ketqua= spPhepCong 7,9
Print N'Kết quả cộng hai số là: ' + cast(@ketqua as char)
```

The bottom window is the 'Messages' pane, which displays the output of the query:

```
Kết quả cộng hai số là: 16
```

Hàm IF

```
IF <Boolean Expression>
BEGIN
    <Các lệnh>
END
ELSE
BEGIN
    <Các lệnh>
END
```

Ví dụ:

```
Create Proc spTonKho
@thang int,
@nam int
As
    If @thang<1 Or @thang>12
        Print 'Tháng sai'
    Else
        Begin
            Select * From TonKho
            Where Thang = @thang
            And Nam = @nam
        End
Go
```

Hàm While

```
WHILE <Boolean Expression>  
BEGIN  
    <Các lệnh>  
    [CONTINUE][BREAK]  
END
```

WHILE được dùng nhiều trong kiểu dữ liệu **CURSOR**. Thông thường **WHILE** thường sử dụng để duyệt từ mẫu tin đầu tiên đến mẫu tin cuối cùng hoặc ngược lại của câu lệnh Select.

✓ Ví dụ: $S = 1 + 3 + \dots + (2N-1)$

```
Create Proc spTong
```

```
@n int
```

```
AS
```

```
Declare @i int, @s int
```

```
Set @i = 1
```

```
Set @s = 0
```

```
WHILE @i <= 2*@n - 1
```

```
Begin
```

```
Set @s = @s + @i
```

```
Set @i = @i + 2
```

```
End
```

```
Return @s
```

Hàm Case ... When

- ❑ Cấu trúc **Case...When** kiểm tra điều kiện để chọn lựa giá trị kết quả

Dạng 1

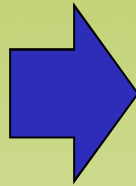
```
CASE <tên cột>/<biểu thức>  
  WHEN <giá trị 1> THEN <biểu thức 1>  
  WHEN <giá trị 2> THEN <biểu thức 2>  
  ...  
  [ELSE <biểu thức n>]  
END
```

Dạng 2

```
CASE  
  WHEN <bt luận lý 1> THEN <biểu thức 1>  
  WHEN <bt luận lý 2> THEN <biểu thức 2>  
  ...  
  [ELSE <biểu thức n>]  
END
```

Hàm Case ... When

MANV	HO	TEN	CHUCVU
01	Nguyễn Bá	Long	GD
02	Trần Minh	Tuấn	NV
03	Võ Tấn	Viên	PGD
04	Nguyễn Thị	Vỹ	NV
05	Lâm Minh	Nhật	NV
06	Bá Xuân	Cường	TP

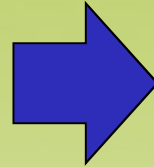


MANV	HO	TEN	CHUCVU	PCCV
01	Nguyễn Bá	Long	GD	500
02	Trần Minh	Tuấn	NV	0
03	Võ Tấn	Viên	PGD	300
04	Nguyễn Thị	Vỹ	NV	0
05	Lâm Minh	Nhật	NV	0
06	Bá Xuân	Cường	TP	200

```
SELECT MANV, HO, TEN, CHUCVU,  
       CASE CHUCVU  
         WHEN 'GD' THEN 500  
         WHEN 'PGD' THEN 300  
         WHEN 'TP' THEN 200  
         ELSE 0  
       END AS PCCV  
FROM NHANVIEN
```

Hàm Case ... When

MAHV	HO	TEN	DIEM
01	Nguyễn Bá	Long	7.00
02	Trần Minh	Tuấn	4.00
03	Võ Tấn	Viên	9.00
04	Nguyễn Thị	Vỹ	6.50
05	Lâm Minh	Nhật	5.00
06	Bá Xuân	Cường	2.00



MAHV	HO	TEN	KETQUA
01	Nguyễn Bá	Long	ĐẠT
02	Trần Minh	Tuấn	KỖ
03	Võ Tấn	Viên	ĐẠT
04	Nguyễn Thị	Vỹ	ĐẠT
05	Lâm Minh	Nhật	ĐẠT
06	Bá Xuân	Cường	KỖ

```
SELECT MAHV, HO, TEN,  
       CASE  
         WHEN DIEM >= 5 THEN 'ĐẠT'  
         ELSE 'KỖ'  
       END AS KETQUA  
FROM HOCVIEN
```

Phát biểu Exec(@string_var | [N]'string_expression')

- ❑ Exec(@string_var | [N]'string_expression'): Dùng thực hiện phát biểu T-SQL lưu trong biến chuỗi hoặc được tạo bởi một biểu thức chuỗi.

✓ Ví dụ:

EXEC ('Use HoaDon; Select MaHD, MaNV, NgayLHD FROM HoaDon')