

# CÁC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

---

## STORED PROCEDURE

Giảng viên: ThS. Nguyễn Thị Uyên Nhi  
Email: [uyennhisgu@gmail.com](mailto:uyennhisgu@gmail.com)

# 1. Transact-SQL nâng cao

- ❖ Transact-SQL là ngôn ngữ SQL mở rộng dựa trên SQL chuẩn của ISO (International Organization for Standardization) và ANSI (American National Standards Institute) được sử dụng với P-SQL (Procedural-SQL).

# 1.1. Khai biến

## a) Biến cục bộ:

- ✓ Là một đối tượng có thể chứa giá trị thuộc một kiểu dữ liệu nhất định, tên biến bắt đầu bằng một ký tự @.
- ✓ Biến cục bộ có giá trị trong một *query batch* hoặc trong một thủ tục thường trú (stored procedure) hoặc hàm (function).
- ✓ Khai báo biến cục bộ bằng lệnh declare: cung cấp tên biến và kiểu dữ liệu:

**Declare** *tên\_biến Kiểu\_dữ\_liệu*

*Ví dụ:*

Declare @MaSinhVien *char*(10)

Declare @HoTen *nvarchar*(30)

Declare @Sum *float*, @Count *int*

## 1.1. Khai biến

- ✓ Để gán giá trị cho một biến cục bộ dùng lệnh *set*. Giá trị gán cho biến phải phù hợp với kiểu dữ liệu của biến.

*Set tênbiến = giátrị*

*Set tênbiến = tênbiến*

*Set tênbiến = biểuthức*

*Set tênbiến = kết\_quả\_truy\_vấn*

- ✓ Ví dụ:

*Set @MaLop = 'TH2001'*

*Set @SoSV = (select count (\*) from SinhVien)*

*Set @MaLop = 'TH'+Year(@NgayTuyenSinh)*

*Đưa kết quả truy vấn vào biến:*

*SV(MaSV: int; HoTen: nvarchar(30), Tuoi int)*

*Select @Var1 = HoTen, @Var1 = Tuoi*

*from SV where MaSV = 1*

# 1.1. Khai biến

## b) Biến toàn cục:

- ✓ Là các biến hệ thống do SQL Server cung cấp, tên biến bắt đầu bằng 2 ký tự @@
- ✓ SQL tự cập nhật giá trị cho các biến này, người sử dụng không thể gán giá trị trực tiếp.
- ✓ Một số biến hệ thống thường dùng
  - ❖ **@@error**: thông báo mã lỗi, nếu @@error = 0: thao tác thực hiện thành công.
  - ❖ **@@rowcount**: cho biết số dòng bị ảnh hưởng bởi lệnh cuối (insert, update, delete).
  - ❖ **@@trancount**: cho biết số giao dịch đang hoạt động trên kết nối hiện tại.
  - ❖ **@@ fetch\_status**: cho biết thao tác lấy dữ liệu từ cursor có thành công không.

## 1.2. Cấu trúc điều khiển

### a) Lệnh If...else

✓ Chức năng: xét điều kiện để quyết định những lệnh T-SQL nào sẽ được thực hiện

✓ Cú pháp:

*If biểu\_thức\_điều\_kiện*

*Lệnh| Khối\_lệnh*

*[Else Lệnh| Khối\_lệnh]*

Trong đó: Khối lệnh là một hoặc nhiều lệnh nằm trong cặp từ khóa **begin...end**

## 1.2. Cấu trúc điều khiển

✓ Ví dụ: xét 2 lược đồ quan hệ (LĐQH)

HOCPHAN(MAHP, TENHP, SISO)

DANGKY(MASV, MAHP)

Viết lệnh để thêm một đăng ký mới cho sinh viên có mã số 001 vào học phần HP01 (giả sử học phần này đã tồn tại trong bảng HocPhan) và phải bảo đảm SISO HOCPHAN luôn  $< 50$ .

## 1.2. Cấu trúc điều khiển

```
DECLARE @SISO int
SELECT @SISO = SISO FROM HOCPHAN
WHERE MAHP= 'HP01'
IF @SISO < 50
    BEGIN
        INSERT INTO DANGKY(MASV, MAHP) VALUES('001',
        'HP01')
        PRINT N'DĂNG KÝ THÀNH CÔNG'
    END
ELSE
    PRINT N'HỌC PHẦN ĐÃ ĐỦ SV'
```



## 1.2. Cấu trúc điều khiển

### b) Lệnh While

- ✓ Chức năng: thực hiện lặp lại một đoạn lệnh T-SQL khi điều kiện còn đúng.
- ✓ Cú pháp:  
While *biểu\_thức\_điều\_kiện*  
    *Lệnh\ Khối lệnh*
- ✓ Có thể sử dụng *Break* và *Continue* trong khối lệnh của While:
  - ❖ Break: thoát khỏi vòng while hiện hành.
  - ❖ Continue : trở lại đầu vòng while, bỏ qua các lệnh sau đó.

## 1.2. Cấu trúc điều khiển

✓ Ví dụ:

Xét lược đồ quan hệ:

SINHVIEN(MASV: INT, HOTEN: NVARCHAR(30))

*Viết lệnh xác định một mã sinh viên mới theo qui định: mã sinh viên tăng dần, nếu có chỗ trống thì mã mới xác định sẽ chèn vào chỗ trống đó.*

Chẳng hạn, nếu trong bảng sinhvien đã có các mã sinh viên 1, 2, 3, 7 □ mã sinh viên mới là 4.

## 1.2. Cấu trúc điều khiển

```
DECLARE @STT INT
```

```
SET @STT = 1
```

```
WHILE EXISTS(SELECT *
```

```
FROM SV
```

```
WHERE MASV = @STT)
```

```
SET @STT = @STT+1
```

```
INSERT INTO SV(MASV, HOTEN) VALUES(@STT, 'NGUYEN  
VAN A')
```

## 1.2. Cấu trúc điều khiển

### c) Lệnh Case

- ✓ Chức năng: kiểm tra một dãy các điều kiện và trả về kết quả phù hợp với điều kiện đúng. Lệnh CASE được sử dụng như một hàm trong câu SELECT.
- ✓ Cú pháp: Có hai dạng:

- Dạng 1 (simple case):

```
Case Biểu_thức_đầu_vào  
  When Giá_trị then kết_quả  
  [...n]  
  [ Else kết_quả_khác ]  
End
```

- Dạng 2 (searched case):

```
Case  
  When biểu_thức_điều_kiện then kết_quả  
  [...n]  
  [Else kết_quả_khác]  
End
```

## 1.2. Cấu trúc điều khiển

✓ **Ví dụ:** xét lược đồ quan hệ:

NHAN VIEN(MANV, HOTEN, NGAYSINH, CAPBAC, PHAI)

- a) *Cho biết những nhân viên đến tuổi nghỉ hưu biết rằng tuổi về hưu của nam là 60, của nữ là 55.*
- b) *Cho biết mã NV, họ tên và loại nhân viên (cấp bậc  $\leq 3$ : bình thường, cấp bậc = null: chưa xếp loại, còn lại: cấp cao).*

## 2. STORED PROCEDURE

✓ Các tùy chọn lập trình thủ tục trong Transact-SQL

Kiểu	Nhóm câu Lệnh (Batch)	Lưu trữ	Thực thi	Tham số
<b>Mã kịch bản (Script)</b>	Gồm nhiều nhóm câu lệnh	Trong file trên ổ đĩa	Từ công cụ client như Management Studio hoặc SQLCMD	Không
<b>Stored Procedure</b>	Duy nhất	Trong đối tượng của CSDL	Bởi ứng dụng hoặc trong mã kịch bản SQL	Có
<b>Hàm người dùng định nghĩa</b>	Duy nhất	Trong đối tượng của CSDL	Bởi ứng dụng hoặc trong mã kịch bản SQL	Có
<b>Trigger</b>	Duy nhất	Trong đối tượng của CSDL	Tự động bởi server khi một truy vấn hành động cụ thể xảy ra	Không

## 2.1. KHÁI NIỆM

- ✓ **Stored Procedure** là một tập các câu lệnh **T-SQL** thực hiện một nhiệm vụ cụ thể, được đặt tên và lưu trữ trong CSDL dưới dạng đã biên dịch.
- ✓ **Stored procedure** cung cấp một phương pháp hữu ích cho việc thực thi lặp lại cùng một nhiệm vụ
  - Giúp tái sử dụng code
  - Khi thực thi lại một nhiệm vụ, sử dụng lời gọi Stored Procedure thay vì viết và thực thi lại cùng một tập hợp các câu lệnh.
- ✓ Cách sử dụng các biến, cấu trúc điều khiển trong Stored Procedure tương tự như mã kịch bản

## 2.1. KHÁI NIỆM

### Ý NGHĨA:

- ✓ Tính tái sử dụng, tính uyển chuyển nhờ hệ thống tham số.
- ✓ Khi biên dịch SP, các lệnh được tối ưu hóa sao cho thực thi hiệu quả nhất □ được lưu bền vững. Khi gọi thực thi thủ tục không cần biên dịch và tối ưu hóa lại □ tiết kiệm thời gian và tài nguyên hơn khối lệnh tương đương trong thân thủ tục.
- ✓ Trong ứng dụng triển khai theo môi trường client/server, client gửi lời gọi SP lên server thì chiếm đường truyền ít hơn rất nhiều lần so với việc gửi khối lệnh tương đương trong thân thủ tục -> Giảm khối lượng thông tin trao đổi khi ứng dụng gửi yêu cầu thực hiện công việc về cho server do đó tránh nghẽn đường truyền, giảm trì trệ.



## 2.1. KHÁI NIỆM

### Ý NGHĨA:

- ✓ Đóng gói chỉ các thao tác cho phép trên CSDL vào các SP và quy định truy xuất dữ liệu phải thông qua SP. Ngoài ra còn có thể phân quyền trên SP -> Hỗ trợ tốt hơn cho việc đảm bảo an toàn (security) cho CSDL.
- ✓ SP giúp cho việc kết xuất báo biểu bằng Crystal Report trở nên đơn giản và hiệu quả hơn rất nhiều so với việc kết xuất dữ liệu trực tiếp từ các bảng và khung nhìn.

## 2.1. KHÁI NIỆM

### Một số Quy tắc khi tạo SP :

- ✓ Một thủ tục không thể tạo/xóa 1 đối tượng rồi lại tham chiếu đến đối tượng đó.
- ✓ Các thủ tục có thể tham chiếu đến các bảng tạm thời.
- ✓ Các bảng tạm (Temporary table) có thể được tạo ra bên trong thủ tục và được tự động xóa khi thủ tục kết thúc.
- ✓ Có thể tham chiếu đến các đối tượng từ các CSDL khác và server từ xa.
- ✓ Cho phép các thủ tục đệ quy (recursive) – thủ tục có thể gọi chính nó.
- ✓ Nhiều nhất là 2100 Parameters trong 1 SP.
- ✓ Các thủ tục có thể được gọi lồng nhau tối đa tới 32 mức.
- ✓ Kích cỡ cực đại của 1 thủ tục là 128 MB, và còn tùy thuộc vào bộ nhớ.

## 2.2. TẠO STORE PROCEDURE

```
CREATE PROCEDURE procedure_name  
@parameter1 data_type[output] /*các tham số*/,  
@parameter2 data_type[output]  
AS  
BEGIN  
[khai báo các biến cho xử lý]  
{Các câu lệnh TRANSACT-SQL}  
END  
GO
```

### Lưu ý:

- Trong đó các câu lệnh T-SQL có thể nằm trong cặp BEGIN ... END hoặc không.
- Tên tham số đặt theo qui tắc như tên biến cục bộ.
- Giá trị trả về của SP dùng một (hay một số) tham số *output*. Phần [output] là phần có thể có hoặc không để xác định loại tham số.
- Tên thủ tục chứa tối đa 128 kí tự

## 2.2. TẠO STORE PROCEDURE

Ví dụ 1:

```
CREATE PROCEDURE XinChao
    @hoTen nvarchar(50)
AS
BEGIN
    print N'Xin chào ' + @hoTen
END
GO
```

## 2.2. TẠO STORE PROCEDURE

Ví dụ 2: Xây dựng SP cho biết danh sách sinh viên của một lớp có mã cho trước

```
CREATE PROC DS_LOP @MALOP VARCHAR(10)
AS
    SELECT SV.MASV, SV.HOVATEN, SV.NGAYSINH
    FROM SINHVIEN SV
    WHERE SV.LOP = @MALOP
GO
```

## 2.3. THỰC THI SP

### Cú pháp gọi thực hiện thủ tục SP

- Để gọi thực hiện một SP ta sử dụng cú pháp sau:

EXEC|EXECUTE <Tên SP> [<Danh sách các tham số>]

- ✓ Khi gọi thực hiện SP, cần truyền đủ tham số với kiểu dữ liệu phù hợp và thứ tự chính xác như khai báo trong định nghĩa SP.
- ✓ Có thể truyền giá trị cho tham số đầu vào (input) là một hằng hoặc một biến đã gán giá trị, không truyền được một biểu thức.
- ✓ Để nhận được giá trị kết quả (thông qua tham số đầu ra), cần truyền vào một biến và có từ khóa output.
- ✓ VÍ DỤ: EXEC XinChao N'Hằng'

## 2.3. THỰC THI SP

- ✓ Cú pháp của câu lệnh xóa SP: DROP PROC  
DROP {PROC|PROCEDURE} procedure\_name
- ✓ Cú pháp của câu lệnh sửa SP: ALTER PROC  
ALTER PROCEDURE procedure\_name  
@parameter1 data\_type [output] /\*các tham số\*/,  
@parameter2 data\_type [output]  
AS  
BEGIN  
[khai báo các biến cho xử lý]  
{Các câu lệnh transact-sql}  
END  
GO

## 2.4. THAM SỐ TRONG SP

- Tham số đầu vào: Đây là loại tham số mặc định, cho phép truyền các giá trị vào trong stored procedure để hỗ trợ xử lý.

VÍ DỤ:

```
CREATE PROC SUM
```

```
    @So1 int,
```

```
    @So2 int
```

```
AS
```

```
BEGIN
```

```
    DECLARE @KQ int
```

```
    SET @Kq = @So1 + @So2
```

```
    PRINT @KQ
```

```
END
```

```
GO
```

```
EXEC SUM 5,7
```



## 2.4. THAM SỐ TRONG SP

- Tham số đầu ra: Tham số dùng để nhận kết quả trả về từ Stored Procedure. Sử dụng từ khóa OUTPUT (hoặc viết tắt là OUT) để xác định tham số.

VÍ DỤ:

```
CREATE PROC TRU
    @So1 int,
    @So2 int,
    @KQ int OUTPUT
AS
BEGIN
    SET @KQ = @So1 - @So2
END
GO
DECLARE @test int
EXEC Tru 1, 2, @test output
PRINT @test
```

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

### a) Trả về giá trị từ lệnh RETURN

Lệnh RETURN được sử dụng để trả về giá trị từ Stored Procedure mà không cần sử dụng tham số đầu ra. Giá trị trả về này có một số đặc điểm:

- ✓ Giá trị trả về chỉ có thể là số nguyên. Nếu trả về các loại giá trị khác thì lúc thực thi Stored Procedure sẽ báo lỗi (ngoại trừ 1 số kiểu dữ liệu được tự động chuyển đổi sang kiểu số nguyên như: float, double,...).
- ✓ Giá trị trả về mặc định là 0.
- ✓ Có thể nhận giá trị trả về này bằng 1 biến.
- ✓ Sau khi gọi RETURN, Stored Procedure sẽ trả về giá trị và kết thúc xử lý.

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

```
CREATE PROC Test
    @Lenh int
AS
BEGIN
    if (@Lenh = 1) return 1
    if (@Lenh = 2) begin
        declare @float float set @float = 2.6
        return @float
    end
    if (@Lenh = 3) begin
        declare @char varchar(50)
        set @char = 'hello'
        return @char
    end
END
GO
```

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

```
DECLARE @test float  
EXEC @test = Test 3  
PRINT @test
```

- Nếu giá trị truyền vào là 1: stored procedure trả về giá trị “1”.
- Nếu giá trị truyền vào là 2: stored procedure trả về giá trị “2”.
- Nếu giá trị truyền vào là 3: stored procedure báo lỗi không thể chuyển chuỗi ‘hello’ thành số nguyên.
- Nếu truyền các giá trị khác: stored procedure trả về giá trị “0”.

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

### **b) Trả về dữ liệu từ lệnh SELECT**

Mỗi lệnh SELECT đặt trong Stored Procedure sẽ trả về 1 bảng.

Ví dụ:

```
CREATE PROC TestSelect
```

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM SINHVIEN
```

```
    SELECT * FROM LOP
```

```
END
```

```
GO
```

```
EXEC TestSelect
```

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

### THÊM SINH VIÊN VÀO CSDL

CREATE PROC ThemSinhVien

    @mssv  varchar(10),  
    @hoTen nvarchar(100),  
    @namSinh int,  
    @danToc nvarchar(20),  
    @maLop varchar(10)

AS

BEGIN

    IF(EXISTS(SELECT \* FROM SinhVien s WHERE s.ma = @mssv))

        BEGIN

            PRINT N'Mã số sinh viên ' + @mssv + N' đã tồn tại'

            RETURN -1

    END

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

```
IF(NOT EXISTS(SELECT * FROM Lop L WHERE L.ma = @maLop))
    BEGIN
        PRINT N'Mã số lớp ' + @maLop + N' chưa tồn tại'
        RETURN -1
    END
    INSERT INTO SinhVien(ma, hoTen, namSinh, danToc, maLop)
    VALUES(@mssv, @hoTen, @namSinh, @danToc, @maLop)
    RETURN 0 /* procedure tự trả về 0 nếu không RETURN */
END
GO
```

```
DECLARE @KQ INT
EXEC @KQ = ThemSinhVien '0212005', N'Nguyễn Văn A', 1987, 'Kinh',
'TH2002/01'
PRINT @KQ
```

## 2.5. TRẢ VỀ GIÁ TRỊ TRONG SP

### TRẢ VỀ DANH SÁCH SINH VIÊN TRONG LỚP

```
CREATE PROC XuatDanhSachSinhVien
    @maLop varchar(10)
AS
BEGIN
    IF(NOT EXISTS(SELECT * FROM Lop L WHERE L.ma = @maLop))
    BEGIN
        PRINT N'Mã số lớp ' + @maLop + N' chưa tồn tại'
        RETURN -1
    END
    SELECT * FROM Lop l where l.ma = @maLop
    /*procedure luôn trả về 0 nếu không RETURN*/
END
GO
```



## Bài tập:

1. Xây dựng SP tính toán giá trị cho đơn hàng có mã cho trước với quan hệ DONHANG(MA, SOLUONG, DONGIA, THUESUAT, CHIETKHAU, THANHTIEN)
2. Viết thủ tục thêm một đăng ký của sinh viên vào một học phần (tổng quát ví dụ trong phần IF... ELSE)

## Bài tập:

3. Xây dựng SP tính điểm trung bình và xếp loại cho sinh viên thuộc lớp cho trước. Giả sử có các quan hệ như sau:

**SINHVIEN** (MASV, HOTEN, DTB, XEPLOAI, LOP)

**MONHOC** (MAMH, TENMH)

**KETQUA** (MAMH, MASV, LANTHI, DIEM)

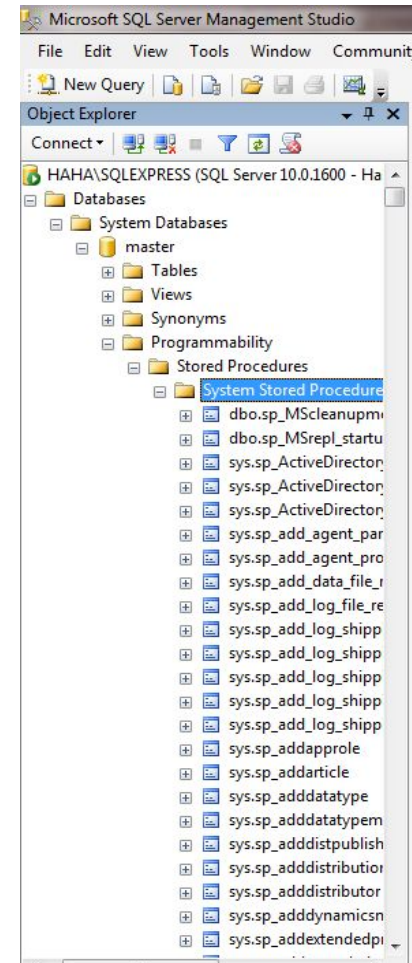
trong đó:

- Điểm thi chỉ tính lần thi sau cùng.
- Xếp loại: Xuất sắc [9, 10], Giỏi [8, 8.9], Khá [7, 7.9], Trung bình [5.0, 6.9], Yếu [0,4.9].
- Kết quả xuất dạng tham số output, không ghi xuống CSDL.

## 2.3. System Stored Procedure

### ■ System Stored Procedure

- Là những Stored Procedure được cung cấp sẵn khi cài đặt SQL Server
- Rất hữu ích khi thực hiện các hành động quản trị và xem thông tin các đối tượng trong SQL Server



## 2.3. System Stored Procedure

Tham khảo System Stored Procedure tại:

<http://msdn.microsoft.com/en-us/library/ms187961.aspx>

Trên trang này các System Stored Procedure được phân theo nhóm

Hai nhóm quan trọng chứa các System Stored Procedure hay sử dụng

Database Engine Stored Procedures: chứa nhiều System SP để xem thông tin các đối tượng trong SQL Server

Security Stored Procedures: chứa các System SP dùng cho mục đích quản trị bảo mật hệ thống

# Giới thiệu một số System Stored Procedure hữu ích

Thủ tục	Mô tả
<b>sp_Help [&lt;tên đối tượng CSDL&gt;]</b>	Trả về thông tin đối tượng CSDL. (bảng, view, stored procedure...) Trả về tổng hợp tất cả đối tượng trong cơ sở dữ liệu hiện thời, nếu không có tham số.
<b>sp_HelpText &lt;tên&gt;</b>	Trả về văn bản của stored procedure, hàm người dùng định nghĩa, trigger, hay view không mã hóa.
<b>sp_HelpDb [&lt;tên CSDL&gt;]</b>	Trả về thông tin CSDL, hoặc toàn bộ cơ sở dữ liệu, nếu không chỉ định tham số.
<b>sp_Helpfile</b>	Xem tên database vật lý và thuộc tính của các tập tin liên quan đến database hiện tại. Dùng stored procedure này để xác định tên vật lý của database cần detach hoặc attach.

# Giới thiệu một số System Stored Procedure hữu ích

Thủ tục	Mô tả
<b>sp_Who</b> [ <b>&lt;ID đăng nhập&gt;</b> ]	Trả về thông tin người đang đăng nhập và các tiến trình đang chạy. Trả về thông tin của toàn bộ người dùng đang hoạt động, nếu không chỉ định tham số.
<b>sp_Columns</b> <tên>	Trả về thông tin cột được định nghĩa trong bảng hoặc view xác định.
<b>sp_tables</b>	Xem danh sách các đối tượng có thể truy vấn trong database hiện tại. Tất cả các đối tượng trong mệnh đề FROM.
<b>sp_columns</b> <tên>	Xem thông tin các cột trong 1 table hoặc view
<b>sp_depends</b> <tên>	Xem danh sách các Stored Procedure, View phụ thuộc vào (tham chiếu đến ) bảng hoặc view trong CSDL



# Giới thiệu một số System Stored Procedure hữu ích

Thủ tục	Mô tả
<b>sp_helpsrvrole</b> ' <b>&lt;Tên Server Role&gt;</b> '	-Trả về danh sách tất cả Server Role nếu không được truyền tham số  - Trả về thông tin mô tả cho Server Role được chỉ định trong câu lệnh
<b>sp_password</b> ' <b>&lt;Password cũ&gt;</b> ', ' <b>&lt;password mới&gt;</b> ', ' <b>&lt;Login ID&gt;</b> '	Thay đổi password cho một Login ID
<b>sp_helplogins</b>	Cung cấp thông tin về LoginID cùng với thông tin các Database User liên kết với Login ID này
<b>sp_addlogin</b>	Thêm một login ID mới
<b>Sp_adduser</b>	Thêm một Database User cho một Login ID

# Thiết lập phím tắt cho các Stored Procedure hệ thống

- Thiết lập phím tắt cho các Stored Procedure hệ thống  
Sử dụng menu Tools > Option

