



PHẦN II: LẬP TRÌNH VỚI T – SQL

CHƯƠNG VI : THỦ TỤC - HÀM

Phù Khắc Anh



Thủ tục – Store Procedure

A decorative blue wavy line with a gradient, flowing from the top left towards the bottom left, partially obscuring the grid background.

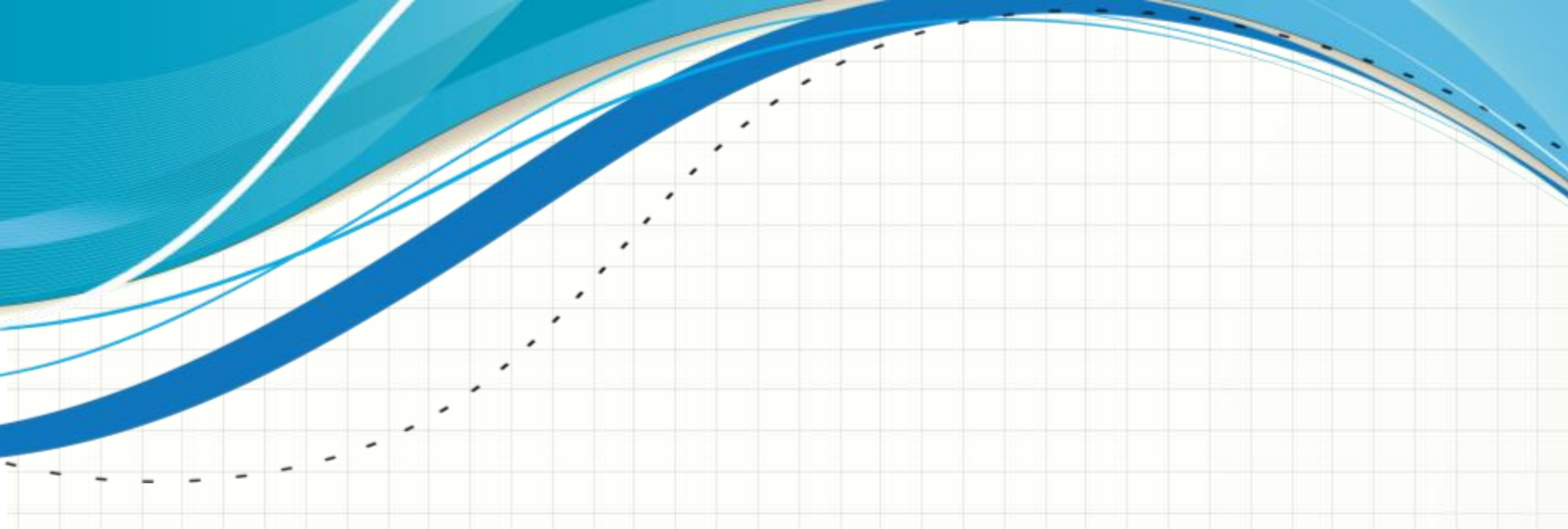
Hàm - Function



Bài tập trên lớp



Bài tập về nhà



I. THỦ TỤC – STORE PROCEDURE

I. Thủ tục – Store Procedure

- Được sử dụng như là một cách tự động hóa các tác vụ. Là một tập các lệnh T-sql, được thực thi như một đơn vị công việc.
- SP thi hành nhanh vì: SQL Server sau khi phân tích và biên dịch SP, nó sẽ giữ execution plans trong procedure cache. Các lần gọi sử dụng SP sau đó sẽ sử dụng lại execution plans đã có .
- Khi viết ứng dụng, gọi SP sẽ dễ bảo trì lệnh sql trong SP hơn.
- SP cho phép thực thi các lệnh SQL theo gói (batch) Và gửi trả tập kết quả cùng một lúc, làm giảm lưu thông trên mạng.

I. Thủ tục – Store Procedure

- SP chỉ chứa các lệnh sql, bao gồm cả những lệnh gọi thi hành thủ tục SP khác.
- SP có thể nhận tham số vào và trả giá trị thông qua tham số.
- Có thể trả về mã tình trạng cho biết tình trạng của việc thi hành SP.

I. Thủ tục – Store Procedure

- * Thủ tục trong SQL Server được lưu trữ tại server khi nó được tạo ra.
- * Trước khi thủ tục được tạo, SQL Server sẽ kiểm tra tính đúng đắn của các cú pháp lệnh. Tên của thủ tục được lưu trong bảng hệ thống **SysObjects** và nội dung được lưu trong bảng hệ thống **SysCommanes**.

I. Thủ tục – Store Procedure

Phân loại:

*** System sp: được lưu trữ trong CSDL master và được đặt tên với tiếp đầu ngữ là sp. Chúng đóng vai trò khác nhau của các tác vụ được cung cấp trong SQL Server.**

Ví dụ: Sp_help, Sp_helpConstraint,

*** Local sp: được lưu trữ trong các CSDL của người dùng, nó thực thi các tác vụ (Task) trong CSDL chứa nó. Một Local sp có thể được người sử dụng tạo hoặc từ các sp hệ thống.**

*** Temporary sp: giống như là một local sp, nhưng nó chỉ hiện hữu cho đến khi kết nối tạo ra nó bị đóng. Nó được nằm trong CSDL TempDB. Có 3 loại temporary sp: local (private), Global, sp tạo trực tiếp trong TempDB.**

I. Thủ tục – Store Procedure

Phân loại:

*** Extended sp:** Là một thủ tục được tạo từ các ngôn ngữ lập trình khác (không phải SQL Server) và nó được triển khai tính năng của một thủ tục trong SQL server. Các thủ tục này có tên với tiếp đầu ngữ là xp.

*** Remote sp:** là một thủ tục được gọi thực thi từ một server từ xa.

I. Thủ tục – Store Procedure

1. Tạo thủ tục :

Cách 1: Dùng SQL Server Manager

R-Click tại Store procedure trong CSDL, chọn New Store procedure

Cách 2: Bằng lệnh Create procedure

**CREATE PROC [EDURE] *procedure_name* [;
number]**

[{ @*parameter data_type* }]

AS

sql_statement [...n]

I. Thủ tục – Store Procedure

Các tham số cách nhau bởi dấu phẩy. Khai báo của mỗi một tham số tối thiểu phải bao gồm hai phần:

Danh_sách_tham_số

- tên tham số được bắt đầu bởi dấu @.
- kiểu dữ liệu của tham số

RECOMPILE

Nếu tùy chọn WITH RECOMPILE được chỉ định, thủ tục sẽ được dịch lại mỗi khi được gọi.

ENCRYPTION

Thủ tục sẽ được mã hoá nếu tùy chọn WITH ENCRYPTION được chỉ định.

Các_câu_lệnh

Tập hợp các câu lệnh sử dụng trong nội dung thủ tục. Các câu lệnh này có thể đặt trong cặp từ khoá BEGIN...END hoặc có thể không.

I. Thủ tục – Store Procedure

Ví dụ : Giả sử ta cần thực hiện một chuỗi các thao tác như sau trên cơ sở dữ liệu

- 1. Bổ sung thêm môn học **Cơ sở dữ liệu** có mã *TI-005* và số đơn vị học trình là 5 vào bảng MONHOC
- 2. Lên danh sách nhập điểm thi môn *Cơ sở dữ liệu* cho các sinh viên học lớp có mã *C24102* (tức là bổ sung thêm vào bảng DIEMTHI các bản ghi với cột MAMONHOC nhận giá trị *TI-005*, cột MASV nhận giá trị lần lượt là mã các sinh viên học lớp có mã *C24105* và các cột điểm là NULL).

I. Thủ tục – Store Procedure

Ví dụ : Giả sử ta cần thực hiện một chuỗi các thao tác như sau trên cơ sở dữ liệu

INSERT INTO MONHOC

VALUES('TI-005','Cơ sở dữ liệu',5)

INSERT INTO DIEMTHI(MAMONHOC,MASV)

SELECT 'TI-005', MASV

FROM SINHVIEN

WHERE MALOP='C24102'

I. Thủ tục – Store Procedure

Ta có thể định nghĩa một thủ tục lưu trữ với các tham số vào là *@mamonhoc*, *@tenmonhoc*, *@sodvht* và *@malop* như sau:

```
CREATE PROC sp_LenDanhSachDiem(  
    @mamonhoc    NVARCHAR(10),  
    @tenmonhoc    NVARCHAR(50),  
    @sodvht       SMALLINT,  
    @malop        NVARCHAR(10))  
  
AS  
  
BEGIN  
    INSERT INTO monhoc  
    VALUES(@mamonhoc,@tenmonhoc,@sodvht)  
    INSERT INTO diemthi(mamonhoc,masv)  
        SELECT @mamonhoc,masv  
        FROM sinhvien  
        WHERE malop=@malop  
  
END
```

*sp_LenDanhSachDiem 'TI-005','Cơ sở
dữ liệu',5,'C24102'*

I. Thủ tục – Store Procedure

- ❖ Không viết chung lệnh tạo sp với các lệnh khác trong cùng một batch
- ❖ SP có thể tham chiếu tới các object trong một DB như table, user-defined function, view, sp, temporary table.

Ví dụ:

I. Thủ tục – Store Procedure

I. Thủ tục – Store Procedure

I. Thủ tục – Store Procedure

- *Xem nội dung thủ tục :*
sp_helptext 'Procedure_name'
- *Xem thông tin về người tạo :*
sp_help 'Procedure_name'
- *Xem các đối tượng mà các lệnh trong sp tham chiếu đến :*
sp_depends Procedure_name
- *Liệt kê tất cả các sp trong database :*
sp_stored_procedures

I. Thủ tục – Store Procedure

2. Thực thi thủ tục :

- ❖ Phải đặt lệnh thi hành sp ở dòng đầu của một query, hay dòng đầu của một batch.

Ví dụ: sp_help Orders

 Select * from Customers

- ❖ Bắt đầu dòng lệnh gọi sp bằng EXECUTE hay EXEC

Ví dụ:

USE Northwind

GO

Select * from Customers

EXEC sp_help Orders

I. Thủ tục – Store Procedure

- ❖ Nếu gọi sp trong db khác, phải viết tên đầy đủ
db_name.owner.sp_name
- ❖ Mã hoá để ngăn user đọc nội dung mã sp, sử dụng
WITH ENCRYPTION trong lệnh tạo sp

Cú pháp:

```
CREATE PROCEDURE procedure_name  
WITH ENCRYPTION  
AS  
    sql_statement [...n]  
GO
```

I. Thủ tục – Store Procedure

Lưu ý :

I. Thủ tục – Store Procedure

```
sp_LenDanhSachDiem 'TI-005','Cơ sở dữ liệu',5,'C24102'
```

Lời gọi thủ tục ở ví dụ trên có thể viết như sau:

```
sp_LenDanhSachDiem @malop='C24102',  
                    @tenmonhoc='Cơ sở dữ liệu',  
                    @mamonhoc='TI-005',  
                    @sodvht=5
```

I. Thủ tục – Store Procedure

3. Sửa đổi thủ tục

- ❖ Phải là owner, thành viên của sysadmin server role, thành viên của db_owner hay db_ddladmin database role thì mới có quyền sửa nội dung sp.
- ❖ Không thể gán quyền khi sửa đổi sp

Cú pháp

ALTER PROCEDURE procedure_name

[WITH option]

AS

sql_statement [...n]

GO

I. Thủ tục – Store Procedure

Ví dụ :

I. Thủ tục – Store Procedure

Ví dụ :

I. Thủ tục – Store Procedure

4. Xóa thủ tục

- ❖ Phải là owner, thành viên của sysadmin server role, thành viên của db_owner hay db_ddladmin database role thì mới có quyền xóa sp.

Cú pháp:

```
DROP PROC owner.stored_procedure_name
```

I. Thủ tục – Store Procedure

5. Tham số trong thủ tục

Input parameter

Dùng để truyền giá trị vào trong sp.

Cú pháp:

```
CREATE PROCEDURE procedure_name  
[@parameter_name data_type] [=default_value]  
[WITH option]  
AS  
sql_statement [...n]
```

- ❖ **Lưu ý:** Nên khai báo giá trị default cho biến, để khi user không nhập giá trị thì sp vẫn chạy thành công

I. Thủ tục – Store Procedure

Ví dụ 1: CREATE PROC dbo.MovieByRating
@rating varchar(5) = NULL
AS
SELECT rating , title
FROM movie
WHERE rating = @rating
ORDER BY title

Thi hành sp có input parameter: có 2 cách dùng biến tham chiếu và chỉ định giá trị theo vị trí

VD:

EXEC MoviebyRating @rating = 'G'

hoặc

EXEC MovieByRating 'G'

I. Thủ tục – Store Procedure

Ví dụ 2: Kiểm tra giá trị của biến

```
CREATE PROC sp_name
@parameter data_type =value
AS
IF @parameter IS NULL
    BEGIN
        PRINT 'Message Line 1'
        PRINT 'Message Line 2'
    RETURN
END
SELECT statement
GO
```

I. Thủ tục – Store Procedure

Ví dụ 3 :

I. Thủ tục – Store Procedure

Ví dụ 4:

I. Thủ tục – Store Procedure

Ví dụ 5 :

I. Thủ tục – Store Procedure

Ví dụ 6: Xét thủ tục sau:

```
CREATE PROCEDURE sp_Conghaiso(@a INT,@b INT, @c INT)  
AS
```

```
    SELECT @c=@a+@b
```

Sau khi đã tạo thủ tục, ta thực thi một tập các câu lệnh sau:

```
    DECLARE @tong INT
```

```
    SELECT @tong=0
```

```
    EXECUTE sp_Conghaiso 100,200,@tong
```

```
    SELECT @tong
```

→ 0

I. Thủ tục – Store Procedure

- ❖ **Output parameter**: Dùng để trả về giá trị. Thông thường giá trị trả về của một sp được sử dụng trong một sp khác.

Cú pháp:

```
CREATE PROCEDURE procedure_name  
[@parameter_name data_type]  
[=default_value] OUTPUT/OUT  
[WITH option]  
AS  
sql_statement [...n]
```

I. Thủ tục – Store Procedure

Ví dụ 1:

```
CREATE PROC count_row  
(@movie_count int OUTPUT)  
AS  
SELECT @movie_count = COUNT(*) FROM  
Movie  
GO
```

- ❖ Thực thi sp có tham số output: phải khai báo một biến để lưu giá trị trả về của tham số output.

```
DECLARE @num int  
EXEC count_row @num OUTPUT  
SELECT @num  
GO
```

I. Thủ tục – Store Procedure

Ví dụ 2: Xét thủ tục sau:

```
CREATE PROCEDURE sp_Conghaiso(  
@a INT,  
@b INT,  
@c INT OUTPUT)  
AS  
  
SELECT @c=@a+@b
```

Sau khi đã tạo thủ tục, ta thực thi một tập các câu lệnh sau:

```
DECLARE @tong INT
```

```
SELECT @tong=0
```

```
EXECUTE sp_Conghaiso 100,200,@tong OUT
```

```
SELECT @tong
```



300

I. Thủ tục – Store Procedure

Ví dụ 3

I. Thủ tục – Store Procedure

Ví dụ 4

I. Thủ tục – Store Procedure

Ví dụ 5

I. Thủ tục – Store Procedure

Tham số với giá trị mặc định

❖ Các tham số được khai báo trong sp có thể nhận các giá trị mặc định. Giá trị mặc định sẽ được gán cho tham số trong trường hợp không truyền đối số cho tham số khi có lời gọi đến sp.

❖ Tham số với giá trị mặc định được khai báo theo cú pháp như sau:

@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định

I. Thủ tục – Store Procedure

```
CREATE PROC sp_TestDefault( @tenlop
NVARCHAR(30)=NULL,
                        @noisinh NVARCHAR(100)='Huế')
AS
    BEGIN
        IF @tenlop IS NULL
            SELECT hodem,ten
            FROM sinhvien INNER JOIN lop
                ON sinhvien.malop=lop.malop
            WHERE noisinh=@noisinh
        ELSE
            SELECT hodem,ten
            FROM sinhvien INNER JOIN lop
                ON sinhvien.malop=lop.malop
            WHERE noisinh=@noisinh AND
                tenlop=@tenlop
    END
```

I. Thủ tục – Store Procedure

Thủ tục *sp_TestDefault* với tham số **@tenlop** có giá trị mặc định là *NULL* và tham số **@noisinh** có giá trị mặc định là *Huế*.

- Cho biết họ tên của các sinh viên sinh tại *Huế*:

```
sp_testdefault
```

- Cho biết họ tên của các sinh viên lớp *Tin K24* sinh tại *Huế*:

```
sp_testdefault @tenlop='Tin K24'
```

- Cho biết họ tên của các sinh viên sinh tại *Nghệ An*:

```
sp_testDefault @noisinh=N'Nghệ An'
```

- Cho biết họ tên của các sinh viên lớp *Tin K26* sinh tại *Đà Nẵng*:

```
sp_testdefault @tenlop='Tin K26',@noisinh=N'Đà Nẵng'
```


I. Thủ tục – Store Procedure

VD :

I. Thủ tục – Store Procedure

VD :

I. Thủ tục – Store Procedure

**QUẢN TRỊ
LỖI**

**QUẢN TRỊ
THỦ TỤC**

I. Thủ tục – Store Procedure

Quản trị lỗi

- ❖ SQL Server cung cấp một số công cụ để quản trị lỗi:
 - *Lệnh RETURN*
 - *Thủ tục sp_addmessage*
 - *Lệnh RAISERROR*
 - *Function @@ERROR.*

I. Thủ tục – Store Procedure

Lệnh RETURN:

Được sử dụng để buộc sp kết thúc không điều kiện. Lệnh RETURN có thể kết thúc sp đồng thời trả về mã tình trạng (status code) phục vụ cho xử lý khác.

I. Thủ tục – Store Procedure

❖ *Ví dụ 1:*

```
CREATE PROC ListCustomers
@cus_id nchar(5) = NULL
AS
IF @cus_id IS NULL
BEGIN
    PRINT 'Please enter a valid customer ID
    number .'
    PRINT 'Use the format [a-z][a-z][a-z][a-
    z][a-z]'
    RETURN
END
SELECT CustomerID, CompanyName, Phone
FROM Customers
WHERE CustomerID = @cus_id
```


I. Thủ tục – Store Procedure

Ví dụ 2:

```
USE Northwind
```

```
GO
```

```
CREATE PROC ListCustomers
```

```
@cus_id nvarchar(5) = NULL
```

```
AS
```

```
    SELECT CustomerID, CompanyName, Phone
```

```
    FROM Customers
```

```
    WHERE CustomerID like @cus_id
```

```
    RETURN (@@rowcount)
```

```
GO
```

Thi hành sp và xem mã tình trạng

```
DECLARE @answer smallint
```

```
EXEC @answer = ListCustomers 'B%'
```

```
SELECT 'Tong so khách hàng ', @answer
```

I. Thủ tục – Store Procedure

Ví dụ 3:

I. Thủ tục – Store Procedure

❖ *Sử dụng thủ tục `sp_addmessage`*: Bạn có thể tạo các message thông báo lỗi (error msg) cho chương trình của mình bằng cách:

Định nghĩa trước các error msg và lưu vào bảng sysmessage

`sp_addmessage`

Trong các sp, gọi hiển thị các error msg

`RAISERROR`

I. Thủ tục – Store Procedure

❖ *Cú pháp 1:*

sp_addmessage

```
@msgnum = msg_id ,  
@severity = severity ,  
@msgtext = 'msg'  
[ , [ @lang = ] 'language' ]  
[ , [ @with_log = ] 'with_log' ]  
[ , [ @replace = ] 'replace' ]
```

❖ *Cú pháp 2:*

```
RAISERROR ( msg_id | msg_str , severity ,  
            state [ , argument [ ,...n ] ] )  
[ WITH option [ ,...n ] ]
```

I. Thủ tục – Store Procedure

*Ví dụ 1: Tạo một error message bằng thủ tục
sp_addmessage*

EXEC sp_addmessage

@msgnum = 50001,

@severity = 10,

@msgtext='Cannot delete customer.

Customer has orders .',

@withlog = 'true'

GO

I. Thủ tục – Store Procedure

Ví dụ 2: *Hiển thị error message trong một thủ tục xóa khách hàng*

```
CREATE PROC DeleteCust
```

```
@cust_num nvarchar(5) = null
```

```
AS
```

```
IF EXISTS (SELECT customerID FROM Orders WHERE  
customerID like @cust_num)
```

```
BEGIN
```

```
RAISERROR (50001, 10, 1)
```

```
RETURN
```

```
END
```

```
DELETE FROM Customers
```

```
WHERE customerID like @cust_num
```

```
GO
```


I. Thủ tục – Store Procedure

Ví dụ 3:

I. Thủ tục – Store Procedure

Sử dụng @@ERROR

Trả về mã lỗi của lệnh sql vừa thi hành.
@@ERROR sẽ trả về giá trị 0 nếu thành công.
Ngược lại, sẽ trả về một giá trị tương ứng với message lỗi trong bảng sysmessages.

❖ *Sử dụng @@ERROR để phát hiện lỗi*

Ví dụ: USE pubs

GO

UPDATE authors SET au_id = '172 32 1176'

WHERE au_id = "172-32-1176"

*IF @@ERROR = 547 print "A check
constraint violation occurred"*

I. Thủ tục – Store Procedure

Vd :

I. Thủ tục – Store Procedure

❖ *Sử dụng @@ERROR là điều kiện thoát khỏi sp*

The IF...ELSE statements in this example test @@ERROR after an INSERT statement in a stored procedure. The value of the @@ERROR variable determines the return code sent to the calling program, indicating success or failure of the procedure.

❖ *Sử dụng @@ERROR để kiểm tra kết quả thi hành lệnh thành công hay không*

This example depends on the successful operation of the INSERT and DELETE statements. Local variables are set to the value of @@ERROR after both statements and are used in a shared error-handling routine for the operation.

I. Thủ tục – Store Procedure

❖ *Sử dụng @@ERROR với @@ROWCOUNT*

This example uses @@ERROR with @@ROWCOUNT to validate the operation of an UPDATE statement. The value of @@ERROR is checked for any indication of an error, and @@ROWCOUNT is used to ensure that the update was successfully applied to a row in the table.

I. Thủ tục – Store Procedure

**QUẢN TRỊ
LỖI**

**QUẢN TRỊ
THỦ TỤC**

I. Thủ tục – Store Procedure

- Buộc SQL Server biên dịch lại sp mỗi lần chạy

CREATE PROC name

WITH RECOMPILE

AS sql_statement

hay

EXEC procedure_name

WITH RECOMPILE