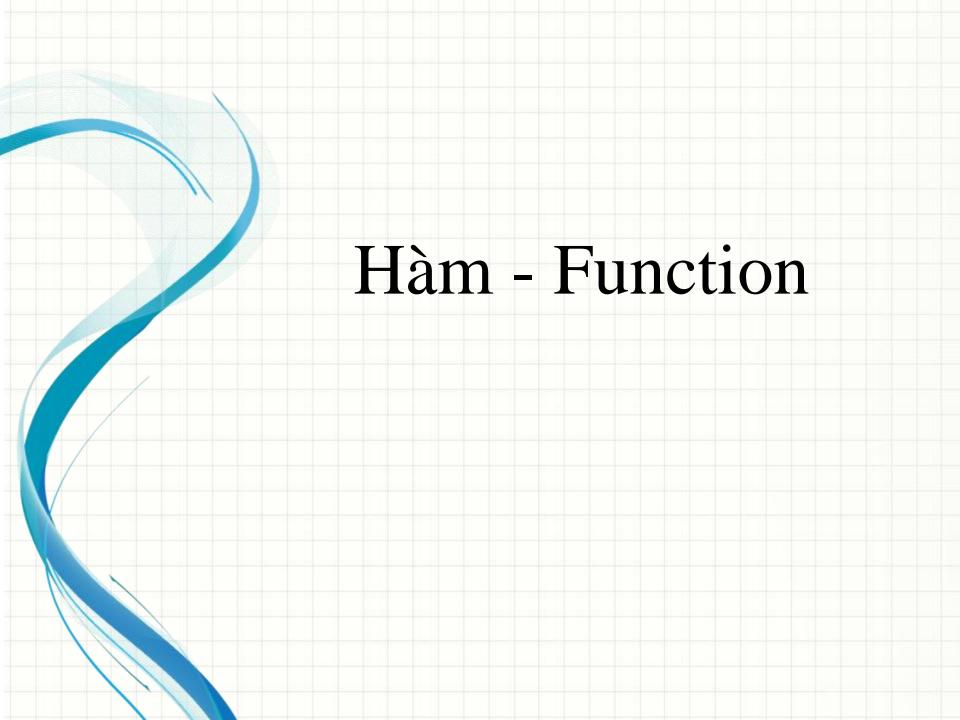
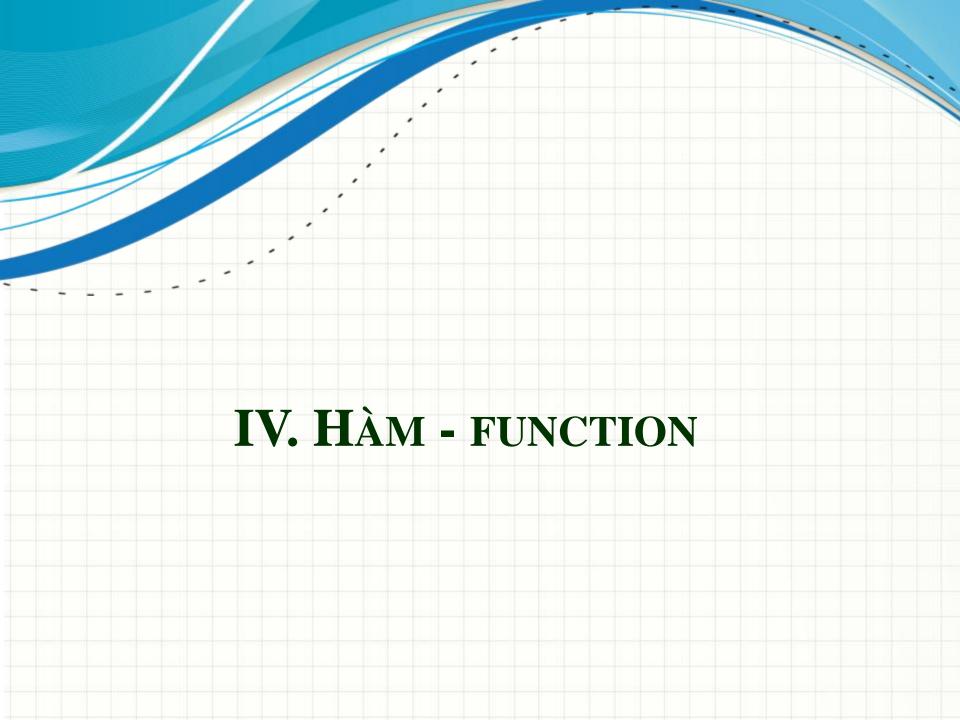


Thủ tục – Store Procedure





Bài tập về nhà



IV. Hàm - Function

Hàm thực sự tương tự như Stored procedure của SQL Server, nội dung bao gồm các phát biểu T-SQL kết hợp tạo thành hàm, có thể gọi thực thi các hàm như là một đơn vị độc lập.

Hàm được dùng trong:

- Danh sách chọn của một câu lệnh Select để cho ra một giá trị.
- ☐ Một điều kiện tìm kiếm của mệnh đề Where trong các câu lệnh T-SQL

IV. Hàm - Function

Hàm thực sự tương tự như Stored procedure của SQL Server, nội dung bao gồm các phát biểu T-SQL kết hợp tạo thành hàm, có thể gọi thực thi các hàm như là một đơn vị độc lập.

Hàm được dùng trong:

- Danh sách chọn của một câu lệnh Select để cho ra một giá trị.
- ☐ Một điều kiện tìm kiếm của mệnh đề Where trong các câu lệnh T-SQL

5.2.2. Các hàm cơ bản

HÀM HỆ THỐNG

HÀM CHUYỂN ĐỔI KIỂU DỮ LIỆU

CÁC HÀM BIỂU DIỄN NGÀY GIỜ a. Hàm chuyển đổi kiểu dữ liệu

Công dụng: chuyển dữ liệu từ số, ngày sang chuỗi

Hàm Cast: chuyển một kiểu dữ liệu sang kiểu bất kỳ.

Cú pháp: Cast(Biểu_Thức as Kiểu_Dữ Liệu)

Ví du: Cast (@tong as varchar(10))

a. Hàm chuyển đổi kiểu dữ liệu (tt)

Hàm CONVERT: chuyển từ ngày sang chuỗi

Cú pháp:

Convert(Kiểu_dữ_Liệu,Biểu_Thức[,Định_Dạng])

Ví du: Convert(char(10), Getdate(), 105)

a. Hàm chuyển đổi kiểu dữ liệu (tt)

Một số kiểu định dạng:

Định dạng năm(YY)	Định dạng năm(YYYY)	Hiển thị dữ liệu
1	101	Mm/dd/yyyy
3	103	Dd/mm/yyyy
5	105	Dd-mm-yyyy
12	112	Yyyymmdd

a. Hàm chuyển đổi kiểu dữ liệu(tt)

Hàm Str: chuyển số thành chuỗi
Cú pháp: STR(số thực, Số ký tự[,số lẻ])
Ví dụ Str(12.345,5,2)->12.35
Chú ý: khi nối chuỗi với số, ngày ta phải chuyển các giá trị này sang chuỗi, sau đó sử dụng dấu + để nối

b. Các hàm về ngày giờ Bảng mô tả các định dạng trong các hàm thời gian

Giá trị	Định dạng	
Năm	уу, уууу	
Quý	qq, q	
Tháng	mm, m	
Ngày trong năm	dy, y	
Ngày trong tuần	dw	
Ngày trong tháng	dd, d	
Tuần	wk, ww	
Giờ	hh	
phút	mi, n	
giây	SS, S	

b. Các hàm về ngày giờ

Hàm DateAdd: dùng cộng một số vào giá trị ngày và trả về một giá trị ngày

Cú pháp: DateAdd(Định dạng,Số,Ngày)->Ngày mới Ví dụ:

declare @ngaymoi datetime
set @ngaymoi=dateadd(mm,5,getdate())
print convert(char(10),@ngaymoi,105)

→ 11-01-2007

Hàm DateDiff: trả về khoảng cách của hai ngày

Cú pháp: Datediff (định dạng,ngày_1,ngày_2)

Ví dụ:

Số nguyên

print Datediff(mm, getdate(),@ngaymoi)

→ 5

Hàm DateName: trả về chuỗi thời gian

Cú pháp: Datename(Định dạng, Ngày)

Ví dụ:

Chuỗi

print datename(dw,getdate())

→ Friday

Hàm DatePart: trả về một giá trị trong của ngày

Cú pháp: Datepart(định dạng,ngày) → số nguyên

Ví du: print datepart(qq,getdate()) → 3

Hàm Getdate() → Trả về ngày hiện hành

Hàm Day(), Month(), Year()

→ trả về ngày,tháng,năm

Datepart	Abbreviation	Values
Hour	hh	0-23
Minute	Mi	0-59
Second	Ss	0-59
Millisecond	Ms	0-999
Day of Year	Dy	1-366
Day	Dd	1-31
Week	wk	1-53
Weekday	dw	1-7
Month	mm	1-12
Quarter	qq	1-4
Year	уу	1753-9999

Function	Example
GETDATE()	SELECT GETDATE()
DATEADD(datepart, numbe r, date)	SELECT DATEADD(mm,4,'01/01/99') - returns 05/01/99 in the current date format
DATEDIFF(datepart,date1, date2)	SELECT DATEDIFF(mm,'01/01/99','05/0 1/99') - returns 4
DATENAME(datepart,date)	SELECT DATENAME(dw,'01/01/2000') - returns Saturday
DATEPART(datepart,date)	SELECT DATEPART(day,'01/15/2000') - returns 15

Hàm Round: hàm làm tròn số

Cú pháp: Round(số, số chử số thập phân)

Vi du: round(12.3456,2)->12.35

Left (chuỗi nguồn, số ký tự) → Lấy các ký tự bên trái

Right (chuỗi nguồn, số ký tự) → Lấy các ký tự bên phải

Substring (chuỗi, ví trí, số ký tự) → Lấy chuỗi con

Hàm Exists (Câu truy vấn Select) = True nếu câu truy

vấn có giá trị trả về

Function	Example			
ABS(num_expr)	SELECT ABS(-43) return 43			
CEILINC(num ovnr)	SELECT CEILING(43.5)			
CEILING(num_expr)	returns 44			
	SELECT FLOOR(43.5) returns			
FLOOR(num_expr)	43			
DOMED (numerous)	SELECT POWER(5,2) returns			
POWER(num_expr,y)	25			
ROUND(num_expr,leng	SELECT ROUND(43.543,1)			
th)	returns 43.500			

Function	Return Values
DB_ID([`database_name'])	Database identification number
DB_NAME([database_id])	Database name
HOST_ID()	Identification number of the workstation
HOST_NAME()	Workstation name
ISNULL(expr,value)	Null expressions replaced with a specified value
OBJECT_ID('obj_name')	Database object identification number
OBJECT_NAME(object_id)	Database object name
USER_ID(['user_name'])	User's database identification number
USER_NAME([user_id])	User's database username

P_Id	ProductName	UnitPrice	UnitsInStock	UnitsOnOrder
1	Jarlsberg	16.2	67	15
2	Mascarpone	23	22	
3	Gorgonzola	45	19	20

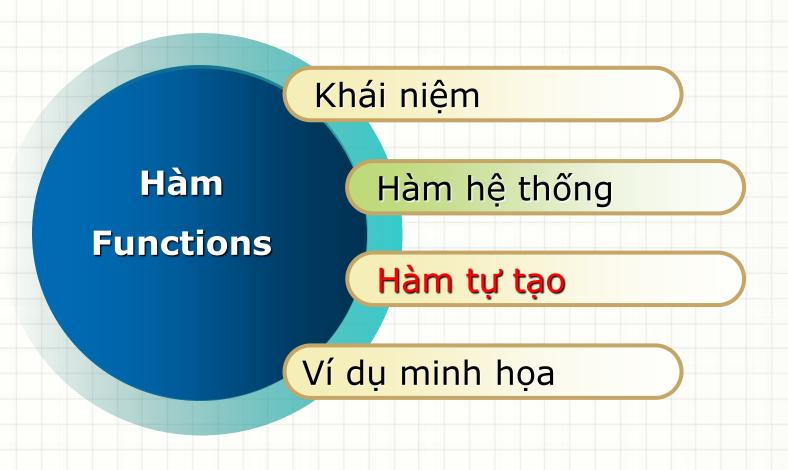
SELECT

ProductName,UnitPrice*(UnitsInStock+UnitsOnOrder)
FROM Products

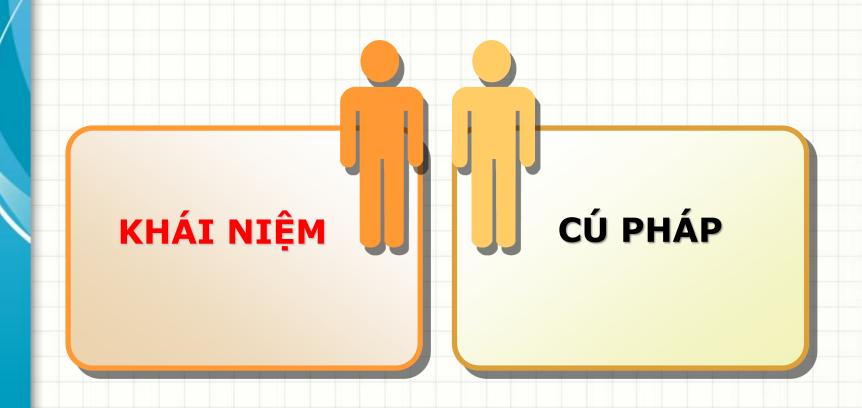
Trong ví dụ trên, nếu có bất kỳ giá trị "UnitsOnOrder" nào NULL thì kết quả trả về là NULL. Khi đó, ta thay:

SELECT ProductName, UnitPrice*(UnitsInStock+ ISNULL(UnitsOnOrder, 0)) FROM Products

5.2. Hàm – Functions



5.2.3. Hàm do người dùng định nghĩa



5.2.3.1. Khái niệm

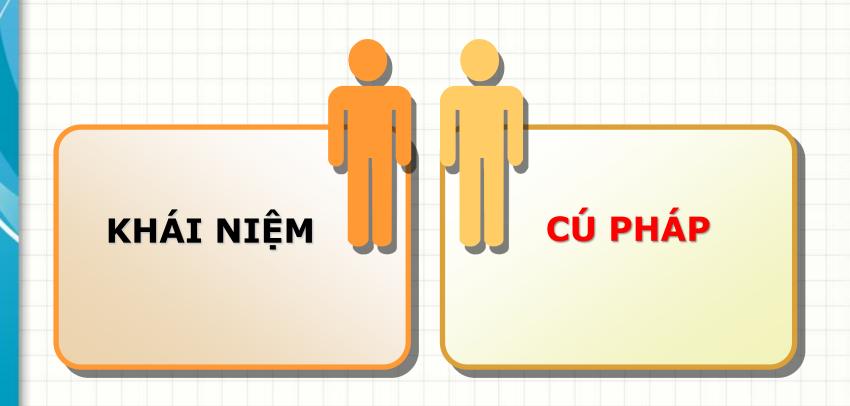
Khái Niệm:

- Do người dùng tự định nghĩa để thực hiện một số chức năng riêng biệt nào đó.
- Mang đầy đủ tính chất của hàm: có thể có tham số vào, xử lý và trả về kết quả.

Các loại hàm:

- Hàm xác định (deter-ministic): luôn trả về 1 giá
 trị khi nhận các giá trị truyền vào như nhau.
- Hàm không xác định (non-deterministic): cho giá
 trị khác nhau tùy thời gọi như hàm Getdate().

5.2.3. Hàm do người dùng định nghĩa



5.3.2.2. Cú pháp xây dựng hàm

CREATE FUNCTION Tên_Hàm

```
([danh_sách_tham_số])
```

RETURNS (Kiểu_trả_về_của_hàm)

AS

BEGIN

Các_câu_lệnh_của_hàm

END

5.3.2.2. Cú pháp xây dựng hàm (tt)



A. Trả về một giá trị

Hàm trả về một giá trị: giá trị trả về có kiểu dữ liệu là một trong các kiểu của SQL Server

Cú pháp:

```
Create Function Ten_Ham[(Các tham số)]
Returns Kiểu_dữ_liệu_trả_về as
Begin
```

<các xử lý>

Return ket_qua

End

5.3.2.2. Cú pháp xây dựng hàm (tt)



b. Trả về một bảng dữ liệu

Hàm trả về dữ liệu được lấy từ các bảng trong

CSDL: giống như view nhưng có tham số vào

Cú pháp:

Create Function Tên_Hàm[(các tham số vào)]
Returns Table as
Return(Câu lệnh select)

❖ Gọi hàm loại này giống như view

b. Trả về một bảng dữ liệu

- Kiểu trả về của hàm phải được chỉ định bởi mênh đề RETURNS TABLE.
- Trong phần thân của hàm chỉ có duy nhất một câu lệnh RETURN xác định giá trị trả về của hàm thông qua duy nhất một câu lệnh SELECT. Ngoài ra, không sử dụng bất kỳ câu lệnh nào khác trong phần thân của hàm.

5.3.2.2. Cú pháp xây dựng hàm (tt)



c. Hàm tạo bảng Hàm tạo bảng: tạo và trả về một bảng, trong hàm có thể chứa tất cả các lệnh T-SQL Cú pháp

```
Create Function Ten_Ham [(các tham số)]
Returns @Ten_Bang_tra_ve Table
(Tên_cột Kiểu_dữ_liệu[,...] ) as
Begin
      <các lênh T-SQL>
      Insert into @ten_bang_tra_ve ...
      Return
End
```

c. Hàm tạo bảng (tt)

❖ Trong trường hợp cần phải sử dụng đến nhiều câu lệnh trong phần thân của hàm, ta sử dụng cú pháp như sau để định nghĩa hàm:

CREATE FUNCTION Tên_Hàm([danh_sách_tham_số])

RETURNS @biến_bảng TABLE định_nghĩa_bảng

AS

BEGIN

Các_câu_lệnh_trong_thân_hàm RETURN

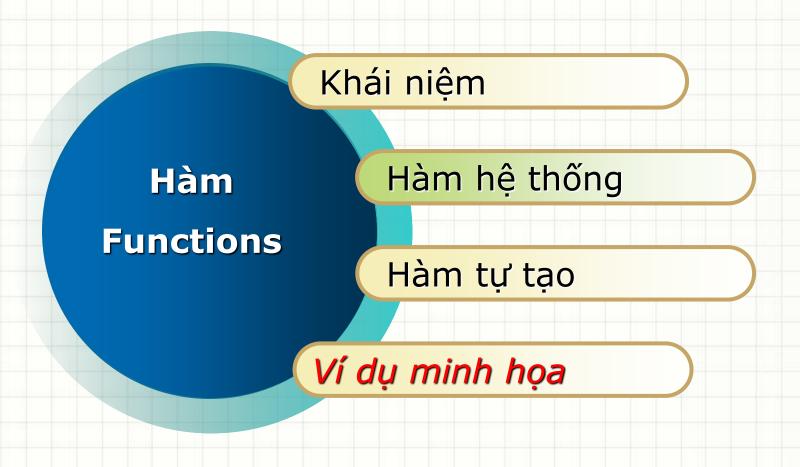
END

c. Hàm tạo bảng (tt)

Khi định nghĩa hàm dạng này cần lưu ý một số điểm sau:

Cấu trúc của bảng trả về bởi hàm được xác định dựa vào định nghĩa của bảng trong mệnh đề RETURNS. Biến @biến_bảng trong mệnh đề RETURNS có phạm vi sử dụng trong hàm và được sử dụng như là một tên bảng. ❖ Câu lệnh RETURN trong thân hàm không chỉ định giá trị trả về. Giá trị trả về của hàm chính là các dòng dữ liệu trong bảng có tên là @biếnbảng được định nghĩa trong mênh đề RETURNS

5.2. Hàm – Functions



Ví du 1: Hàm trả về một giá trị create function thu (@ngay DATETIME) **RETURNS NVARCHAR(10)** AS **BEGIN** DECLARE @st NVARCHAR(10) SELECT @st=CASE DATEPART(DW,@ngay) WHEN 1 THEN 'Chu nhât' WHEN 2 THEN 'Thứ hai' WHEN 3 THEN 'Thứ ba' WHEN 4 THEN 'Thứ tư' WHEN 5 THEN 'Thứ năm' WHEN 6 THEN 'Thứ sáu' ELSE 'Thứ bảy' **END** RETURN (@st) /* Trị trả về của hàm */

END

Ví dụ 1: Hàm trả về một giá trị (tt)

Một hàm khi đã được định nghĩa có thể được sử dụng như các hàm do hệ quản trị cơ sở dữ liệu cung cấp (thông thường trước tên hàm ta phải chỉ định thêm tên của người sở hữu hàm)

Ví dụ Câu lệnh SELECT dưới đây sử dụng hàm đã được định nghĩa ở ví dụ trước:

SELECT masv,hodem,ten,dbo.Thu(ngaysinh),ngaysinh

FROM SinhVien

WHERE malop='C24102'

Ví dụ 2: Hàm trả về kiểu bảng

CREATE FUNCTION func_XemSV(@khoa SMALLINT)
RETURNS TABLE

AS

RETURN(SELECT masv,hodem,ten,ngaysinh
FROM sinhvien INNER JOIN lop
ON sinhvien.malop=lop.malop
WHERE khoa=@khoa)

Ví dụ 2: Hàm trả về kiểu bảng

❖ Với hàm được định nghĩa như trên, để biết danh sách các sinh viên khoá 25, ta sử dụng câu lênh như sau:

SELECT * FROM dbo.func_XemSV(25)

Danh sách sinh viên khoá 26

SELECT * FROM dbo.func_XemSV(26)

Đối với hàm nội tuyến, phần thân của hàm chỉ cho phép sự xuất hiện duy nhất của câu lệnh RETURN

Ví dụ 3: Hàm tạo bảng

```
CREATE FUNCTION Func_Tongsv(@khoa
SMALLINT)
RETURNS @bangthongke TABLE
(makhoa NVARCHAR(5),
tenkhoa NVARCHAR(50),
tongsosv INT
```

```
AS
   BEGIN
      IF @khoa=0
        INSERT INTO @bangthongke
        SELECT khoa.makhoa,tenkhoa,COUNT(masv)
         FROM (khoa INNER JOIN lop
               ON khoa.makhoa=lop.makhoa)
                INNER JOIN sinhvien
               on lop.malop=sinhvien.malop
        GROUP BY khoa.makhoa,tenkhoa
      ELSE
        INSERT INTO @bangthongke
        SELECT khoa.makhoa,tenkhoa,COUNT(masv)
        FROM (khoa INNER JOIN lop
               ON khoa.makhoa=lop.makhoa)
                INNER JOIN sinhvien
               ON lop.malop=sinhvien.malop
        WHERE khoa=@khoa
        GROUP BY khoa.makhoa,tenkhoa
      RETURN /*Trả kết quả về cho hàm*/
```

Với hàm được định nghĩa như trên, câu lệnh: SELECT * FROM dbo.func_TongSV(25)

→ Kết quả thống kê tổng số sinh viên khoá 25 của mỗi khoa. Câu lênh:

SELECT * FROM dbo.func_TongSV(0)

→ Cho ta biết tổng số sinh viên hiện có (tất cả các khoá) của mỗi khoa

```
create function func_DiemCaoNhat()
returns float
as
begin
     declare @diem float
           @diem = ( select max(diemtrungbinh)
                      from sinhvien)
     return (@diem)
end
Go
```

select * from SinhVien where DiemTrungBinh =
dbo.func_DiemCaoNhat()