

User Defined Functions

Sau khi học này, sinh viên có thể:

- ✓ **Hiểu được 3 loại user-defined functions.**
- ✓ **Tạo, sử dụng và thay đổi được các user-defined functions.**

Nội dung

- ✓ User-defined Function là gì?
- ✓ **Định nghĩa** User-defined Functions
- ✓ **Một số ví dụ tạo** User-defined Functions

User Defined Functions là gì (1)

- ✓ **User-defined function là một chương trình con, có thể nhận hoặc không nhận vào tham số, có thể trả giá trị về là một giá trị vô hướng hay một bảng. Các tham số có các kiểu dữ liệu trong bảng data type ngoại trừ các kiểu sau: timestamp, cursor, hoặc table.**
- ✓ **Không hỗ trợ output parameters.**

User Defined Functions là gì (2)

✓ User-defined functions có 3 loại:

✓ Scalar Functions:

Hàm trả về một giá trị vô hướng

✓ Multi-Statement Table-valued Functions:

✓ là hàm trả về một table được xây dựng từ một hoặc nhiều Transact-SQL statements

✓ Giống stored procedure nhưng có thể được tham chiếu trong mệnh đề FROM của lệnh SELECT, xem như là một view.

✓ In-Line Table-valued Functions

✓ Trả về một table là kết quả của một lệnh SELECT đơn.

✓ Giống view nhưng phức tạp hơn vì có truyền tham số

Tạo Scalar Functions

```
CREATE FUNCTION function_name  
([{@parameter_name [AS]  
  scalar_parameter_data_type [=default]}[  
  ,...n ]] )  
RETURNS scalar_return_data_type  
[ WITH < function_option > [ [,] ...n ] ]  
[ AS ]  
BEGIN  
  function_body  
  RETURN scalar_expression  
END
```

Tham số

- ✓ ***scalar_parameter_data_type***: là các kiểu dữ liệu vô hướng (trừ kiểu *cursor* và *table*). Kiểu *timestamp* và các *user-defined data types* không được hỗ trợ.
- ✓ ***scalar_return_data_type***: giá trị vô hướng được trả về ngoại trừ kiểu *text*, *ntext*, *image*, và *timestamp*.
- ✓ ***scalar_expression***: tính ra giá trị mà hàm sẽ trả về.
- ✓ ***function_body*** là một dãy các lệnh *Transact-SQL* dùng để tính toán ra một *scalar value*.

Ví dụ (1)

```
CREATE FUNCTION fn_NewValue  
(@myinput nvarchar(30))  
RETURNS nvarchar(30)  
BEGIN  
IF @myinput IS NULL  
SET @myinput = 'Not Applicable'  
RETURN @myinput  
END
```

Ví dụ (2)

```
CREATE FUNCTION fn_DateFormat  
(@indate datetime, @separator char(1))  
RETURNS Nchar(20)  
AS  
BEGIN  
RETURN  
CONVERT(Nvarchar(20), datepart(mm,@indate))  
+ @separator  
+ CONVERT(Nvarchar(20), datepart(dd, @indate))  
+ @separator  
+ CONVERT(Nvarchar(20), datepart(yy, @indate))  
END  
go
```

Gọi hàm `SELECT dbo.fn_DateFormat(GETDATE(), ':')`

Áp dụng

1. Viết hàm f1 để kiểm tra một số là chẵn hay lẻ
2. Viết hàm f2 để tính n giai thừa
3. Trong csdl QLVATTU, tính doanh thu của năm, với năm là tham số truyền vào
4. Lấy ra tên của một sản phẩm được bán với số lượng nhiều nhất trong năm, với năm là tham số truyền vào

Câu 4

Create function f4 (@nam int)

Returns nvarchar(30)

Begin

declare @tensp nvarchar(30)

**set @tensp=(select top 1 tenvt from vt v,
cthd c, hd h where c.mavt=v.mavt and
h.mahd=c.mahd and year(ngay)=@nam**

group by tenvt

order by sum(sl) desc)

return @tensp

End

7/29/2024

Tạo Inline Table-valued Functions

CREATE FUNCTION

[*owner_name.*]*function_name*
([{ @*parameter_name* [AS]

scalar_parameter_data_type
[=*default*]} [,...*n*]])

RETURNS TABLE

[WITH < *function_option* > [[,]
***...n*]]**

[AS]

RETURN [(] *select-stmt* [)]

Tham số

- ✓ **TABLE:** chỉ ra rằng giá trị của hàm được trả về là một table
- ✓ **TABLE** trả về được định nghĩa thông qua lệnh **SELECT** đơn.
- ✓ Không kết hợp với các biến trả về
- ✓ ***select-stmt:** là một lệnh **SELECT** đơn để định nghĩa giá trị trả về cho hàm.*

Ví dụ (3)

USE pubs

GO

CREATE FUNCTION SalesByStore

(@storeid varchar(30))

RETURNS TABLE

AS

**RETURN (SELECT title, qty FROM
sales s, titles t WHERE s.stor_id =
@storeid and t.title_id = s.title_id)**

Áp dụng (QLVT)

- ✓ Tạo hàm để trả về danh sách các khách hàng đã mua hàng trong tháng, năm truyền vào
- ✓ Tạo hàm để trả về tên vật tư và số lượng bán được của vật tư đó trong năm truyền vào

Tạo Multi-statement Table-valued Functions

```
CREATE FUNCTION [ owner_name. ]  
  function_name  
  ([{@parameter_name [AS]  
  scalar_parameter_data_type [=default]}[  
  ,...n ] ] )  
  RETURNS @return_variable TABLE  
    < table_type_definition >  
  [ WITH < function_option > [ [,] ...n ] ]  
  [ AS ]  
  BEGIN  
    function_body  
  RETURN  
END
```

Tham số

- ✓ *@return_variable* là một **TABLE variable**, được dùng để lưu trữ và tích lũy các dòng sẽ cho hàm.
 - ✓ *function_body* là dãy các lệnh Transact-SQL để trả về một table cho hàm.
- < **table_type_definition** > :: =
({ *column_definition* |
table_constraint } [,...*n*])

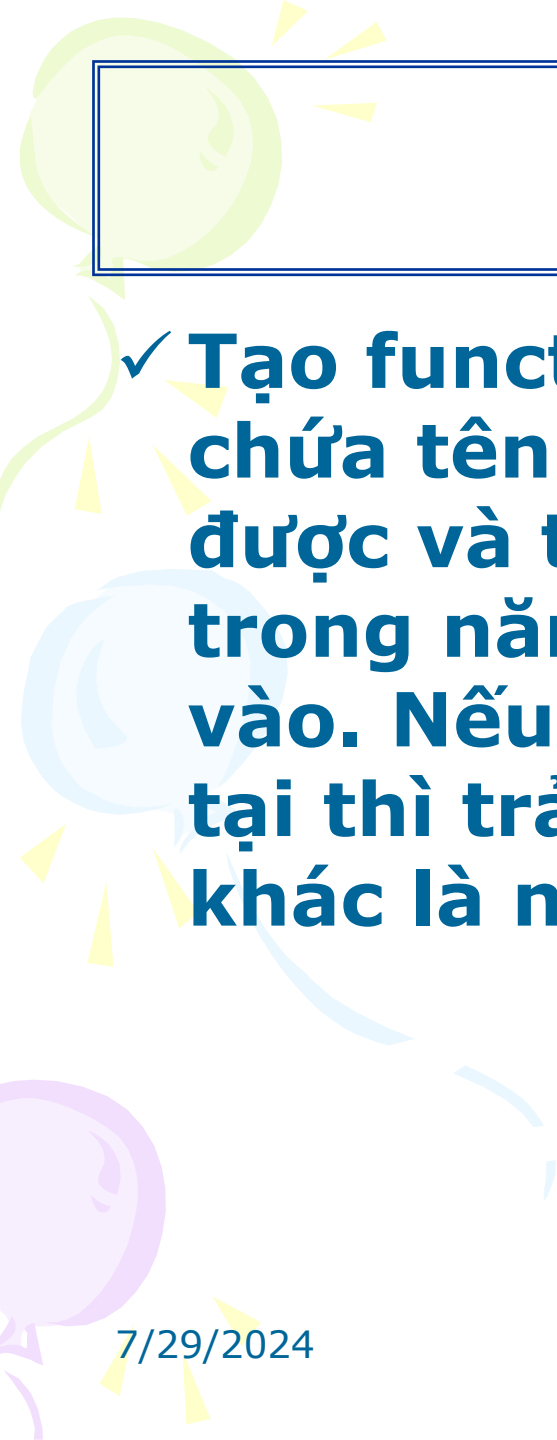
Ví dụ (4)

```
USE Northwind
GO
CREATE FUNCTION fn_Employees
(@length nvarchar(9))
RETURNS @fn_Employees TABLE
(EmployeeID int PRIMARY KEY NOT NULL,
[Employee Name] Nvarchar(61) NOT NULL)
AS
BEGIN
IF @length = 'ShortName'
INSERT @fn_Employees SELECT EmployeeID, LastName
FROM Employees
ELSE IF @length = 'LongName'
INSERT @fn_Employees SELECT EmployeeID,
(FirstName + ' ' + LastName) FROM Employees
RETURN
END
```

Áp dụng CSDL QLVT

- ✓ **Tạo function f7 để trả về một table chứa tên các khách hàng, số lượng hoa đơn đã mua và tổng trị giá của các hoa đơn đó trong năm, với năm là tham số truyền vào. Nếu năm là null thì tính tất cả các năm**

- ✓ **Create function f7 (@nam int)**
- ✓ **Returns @THKH table**
- ✓ **(**
- ✓ **tenkh nvarchar(30), sluonghd int, tongtrigia bigint**
- ✓ **)**
- ✓ **Begin**
- ✓ **If @nam is not NULL**
- ✓ **insert into @THKH select tenkh, count(b.mahd),**
sum(sl*giaban)
- ✓ **from khachhang a, hoadon b, chitiethoadon c**
- ✓ **where a.makh=b.makh and b.MAHD=c.mahd and**
year(ngay) = @nam
- ✓ **group by tenkh**
- ✓ **Else**
- ✓ **... không có điều kiện năm**
- ✓ **return**
- ✓ **end**

- 
-
- ✓ **Tạo function f6 để trả về một table chứa tên vật tư, tổng số lượng bán được và tổng số lãi của vật tư đó trong năm, với năm là tham số truyền vào. Nếu năm truyền vào không tồn tại thì trả về tên vật tư là và các cột khác là null**

Các hạn chế của Functions

- ✓ ***Nondeterministic* functions** (là các hàm không xác định) có thể trả về các giá trị khác nhau với cùng một tập các giá trị input. Ví dụ hàm **GETDATE()**.
- ✓ **Các Built-in nondeterministic functions** không được phép gọi trong phần thân của **user-defined functions**.

Built-in nondeterministic functions

@@ERROR	FORMATMESSAGE	IDENTITY	USER_NAME
@@IDENTITY	GETANSINULL	NEWID	@@ERROR
@@ROWCOUNT	GETDATE	PERMISSIONS	@@IDENTITY
@@TRANCOUNT	GetUTCDate	SESSION_USER	@@ROWCOUNT
APP_NAME	HOST_ID	STATS_DATE	@@TRANCOUNT
CURRENT_TIMESTAMP	HOST_NAME	SYSTEM_USER	
CURRENT_USER	IDENT_INCR	TEXTPTR	
DATENAME	IDENT_SEED	TEXTVALID	

Quyền tạo function

- ✓ Muốn tạo được user defined function, user phải có quyền **CREATE FUNCTION**
- ✓ Quyền **CREATE FUNCTION** là default của sysadmin **role**, db_owner và db_ddladmin **roles**.
- ✓ Các sysadmin và db_owner có thể gán quyền **CREATE FUNCTION** cho các user khác bằng lệnh **GRANT**.

Ví dụ

1. Viết hàm để tính tổng doanh thu của một tháng trong năm.
2. Lấy ra một mã hoá đơn có trị giá lớn nhất.
3. Viết hàm để trả về một bảng các hoá đơn có cùng ngày mua với ngày mua của khách hàng có mã khách hàng là tham số truyền vào.
4. Viết hàm cho biết tên và địa chỉ các khách hàng đã mua mặt hàng có mãVT là tham số truyền vào?

Giải câu 3

Create function KH_HD(@makh nvarchar(30))

Returns table

As

**Return (select mahd, ngay, makh from hoadon
where ngay in (select ngay from hoadon where
makh=@makh) and makh<>@makh)**

Select * from dbo.KH_HD('KH01')



Create function KH_DC(@mavt nvarchar(5))

Returns table

As

**Return (select distinct tenkh,diachi from khachhang a,
hoadon b, chitiethoadon c where a.makh=b.makh and
b.mahd= c.mahd and mavt=@mavt)**

Select * from dbo.KH_DC('VT01')

Cơ sở dữ liệu QLDIEM

1. Tạo hàm tính điểm trung bình ứng với mã sinh viên và học kỳ được truyền vào.
2. Tạo hàm tính xếp loại ứng với mã sinh viên và học kỳ được truyền vào.
3. Tạo view V1 chứa các thông tin MASV, HOTEN, DIEMTB, XL của học kỳ 1 với DIEMTB = trung bình điểm của các môn học (lấy điểm lần 2 nếu có điểm, ngược lại, lấy điểm lần 1), XL là DAT hoặc HONG.



✓ **Tạo f1 trả về doanh thu theo năm với năm là tham số truyền vào**

**Create function f1 (@nam int)
returns int**

As

Begin

declare @dt int

select @dt=sum(sl*giaban)

from chitiethoadon cthd, hoadon hd

where (cthd.madh=hd.mahd)

and year(ngay)=@nam

return @dt

end

Select dbo.f1(2000)

Bài tập

- ✓ **Viết hàm để trả về số lượng khách hàng ở địa chỉ là tham số truyền vào**
- ✓ **Viết hàm để lấy ra danh sách khách hàng ở địa chỉ là tham số truyền vào.**
- ✓ **Đếm số lượng khách hàng mua hàng trong năm với năm là tham số truyền vào. Nếu năm không truyền vào, thì đếm tất cả khách hàng.**