



Hàm do người dùng định nghĩa User-defined Functions

Hàm Do Người Dùng Định Nghĩa

- Hàm của người dùng là các hàm T-SQL có thể nhận các tham số đầu vào và trả về hoặc 1 giá trị đơn (**single value**) hoặc 1 bộ giá trị (**result set**) dạng bảng.
- Hàm người dùng có thể có từ 0 đến 1024 tham số ngõ vào.
- Hàm người dùng không có tham số ngõ ra.

Hàm 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

Hàm trả về 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)

Hàm tạo bảng

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

```
Create Function Ten_Ham[(các tham số)]
```

```
Returns @Ten_Bang_trả_về Table
```

```
(Tên_cột Kiểu_dữ_liệu[,...]
```

```
)as
```

```
Begin
```

```
<các lệnh T-SQL>
```

```
Insert into @ten_bang_tra_ve ..
```

```
Return
```

```
end
```

Ví dụ

- Bạn cần tạo 1 bảng City chứa chi tiết thành phố và tiểu bang của nhân viên trong công ty. Trong khi chèn các hàng vào bảng, rất khó nhớ tên tiểu bang của tất cả các thành phố. Bạn cần làm đơn giản quá trình chèn sao cho tiểu bang của 1 thành phố nào đó được đưa vào bảng một cách tự động.

Ví dụ

```
CREATE FUNCTION dispState (@city varchar(30))
RETURNS varchar(30)
AS
BEGIN
--Tạo 1 biến lưu trữ giá trị được trả về
DECLARE @StateName varchar(30)
SELECT @StateName=case @city
    when 'Menlo Park' then 'CA'
    when 'Oakland' then 'CA'
    when 'Berkeley' then 'CA'
    when 'Nashville' then 'TN'
    when 'Rockville' then 'MD'
    when 'Walnut Creek' then 'CA'
else 'Unknown'
end
return @StateName
END
```

Tạo bảng có sử dụng hàm

CREATE TABLE City
(ccity varchar(30),
cState AS (dbo.dispState(ccity)))

Thực hiện các lệnh INSERT sau để chèn các hàng được
cho vào bảng City.

INSERT City(ccity)
values ('Menlo Park')

INSERT City(ccity)
values ('Oakland')

INSERT City(ccity)
values ('Berkeley')

INSERT City(ccity)
values ('Nashville')

View có tham số

Parameterized views

- Khảo sát view sau:

```
CREATE VIEW vw_PublisherInUSA AS  
SELECT pub_id, pub_name FROM pubs.dbo.Publishers  
WHERE Country = 'USA'
```

- View không hỗ trợ các tham số trong biểu thức điều kiện của mệnh đề WHERE. Hàm lệnh đơn có thể được dùng để hỗ trợ cho các tham số này.

```
CREATE FUNCTION fn_Publishers  
(@CountryPar nvarchar(30) )  
RETURNS table  
AS  
RETURN (  
SELECT pub_id, pub_name  
FROM pubs.dbo.Publishers WHERE country = @CountryPar )
```

- ➔ Hàm fn_Publishers chấp nhận 1 tham số người dùng là tên nước để hàm trả về chi tiết các nhà xuất bản của nước đó.

Các thuận lợi của hàm nhiều lệnh với giá trị trả về dạng bảng

- Hàm người dùng trả về 1 bảng có thể được xem như 1 cách khác tạo view vô cùng hiệu quả.
 - Hàm người dùng trả về bảng có thể dùng ở bất kỳ nơi nào mà bảng hay view được phép sử dụng trong lệnh truy vấn SQL.

Ví dụ: `SELECT * FROM fn_Publishers('USA')`

- Trong khi lệnh tạo View bị giới hạn trong 1 lệnh SELECT, hàm người dùng có thể chứa nhiều lệnh nên linh động hơn lệnh tạo view nhiều.
- Thuận lợi của hàm nhiều lệnh so với thủ tục là hàm có thể được tham chiếu đến trong mệnh đề FROM của lệnh SELECT trong khi thủ tục thì không thể tham chiếu được.

Ví dụ hàm nhiều lệnh trả về dạng bảng

```
CREATE FUNCTION LargeSales(@lParm int )  
RETURNS @LargeSalesTab TABLE  
(  
    StorID char(4),  
    title_id char(6),  
)  
AS  
BEGIN  
    INSERT into @LargeSalesTab  
        SELECT s.stor_id, t.title_id  
        FROM Sales AS s INNER JOIN titles AS t  
            ON s.title_id= t.title_id  
        WHERE s.qty > @lParm  
    RETURN  
END
```

Gọi hàm – Function Invocation

- Hàm trả về giá trị đơn có thể được gọi bằng tên hàm có 2 thành phần

`[database_name.]owner_name.`

`function_name ([argument_expr][,...])`

Ví dụ: gọi hàm tính tổng cho hóa đơn 000001

`SELECT dbo.tinhhtong('000001')`

- Hàm trả về giá trị bảng có thể được gọi bằng tên 1 thành phần.

`[database_name.][owner_name.]function_name
([argument_expr][,...])`

Ví dụ: gọi hàm trả về bảng những hóa đơn bán sách số lượng lớn hơn 30

`SELECT * FROM LargeSales(30)`