



ThaoDTD @dothidiemthao

Theo dõi

★ 487 👤 11 ✎ 10

Đã đăng vào thg 1 12, 2019 4:00 CH - 5 phút đọc

👁 37.8K 💬 0 📖 1

Function người dùng tự định nghĩa trong SQL

...



Bài đăng này đã không được cập nhật trong 6 năm



Đôi lúc chúng ta muốn tự tạo một hàm riêng cho hệ thống để dễ mở rộng và tái sử dụng, trong SQL cũng hỗ trợ chúng ta thực hiện công việc này. Bài viết này mình xin chia sẻ một số kiến thức về Function (Hàm) trong MS SQL Server.

Function là gì?

Định nghĩa:

Là một đối tượng trong cơ sở dữ liệu (CSDL) sử dụng trong các câu lệnh SQL, được biên dịch sẵn và lưu trong CSDL nhằm mục đích thực hiện xử lý nào đó như tính toán phức tạp và trả về kết quả là giá trị nào đó.

Đặc điểm:

- Luôn trả về giá trị
- Gồm 2 loại: Function hệ thống và Function do người dùng tự định nghĩa
- Function người dùng tự định nghĩa gồm 2 loại:
 - Scalar-valued: Trả về giá trị vô hướng của các kiểu dữ liệu T-SQL
 - Table-valued: Trả về bảng, là kết quả của một hoặc nhiều lệnh

Cách định nghĩa Function

1. Tạo Function trả về giá trị loại Scalar-valued

```
CREATE FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], ..., [...]])
RETURNS <kiểu dữ liệu>
[WITH ENCRYPTION]
[AS]
BEGIN
    [Thân của hàm]
    RETURN <Biểu thức giá trị đơn>
END
```



Trong đó:

- Tên function: Tên của hàm chúng ta sẽ tạo

- Tên tham số: Là các tham số Input cho hàm. Khai báo bao gồm tên của tham số (trước tên tham số sử dụng tiền tố @), kiểu dữ liệu của tham số, chúng ta có thể chỉ định giá trị mặc định cho tham số. Có thể chỉ định nhiều tham số đầu vào
- RETURNS: từ khóa này chỉ định kiểu dữ liệu hàm sẽ trả về. Kiểu dữ liệu phải được chỉ định kiểu độ dài dữ liệu. Ví dụ: varchar(100)
- WITH ENCRYPTION: Từ khóa chỉ định code của hàm sẽ được mã hóa trong bảng syscomments.
- AS: Từ khóa cho biết code của hàm bắt đầu.
- BEGIN: Đi cùng với END để tạo thành bao khối bao các câu lệnh trong thân hàm.
- RETURN: Từ khóa này sẽ gửi giá trị tới thủ tục gọi hàm. **Một số lưu ý:**
- Tên function phải là duy nhất trong 1 CSDL. Function được tạo/định nghĩa trong CSDL nào thì chỉ sử dụng trong CSDL đó. Khác với Function có sẵn của SQL được truy cập ở bất cứ đâu.
- Danh sách tham số tối đa 1024 tham số.

2. Tạo Function trả về giá trị loại Table-valued

Function Table-valued có 2 loại:

- Hàm giá trị bảng đơn giản: Trả về bảng, là kết quả của một câu lệnh SELECT đơn
- Hàm giá trị bảng đa câu lệnh: Trả về bảng, là kết quả của nhiều câu lệnh

a) Hàm giá trị bảng đơn giản

```
CREATE FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], ..., [...]])
RETURNS TABLE
```

```
RETURN <Câu lệnh SQL>
END
```

Lưu ý Hàm giá trị bảng đơn còn được gọi là hàm giá trị bảng nội tuyến. Có thể được dùng trong câu lệnh truy vấn thay thế cho tên bảng hoặc tên View.

b) Hàm giá trị bảng đa câu lệnh

```
CREATE FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], ..., [...]])
RETURNS @<tên biến trả về> TABLE (<tên cột 1> <kiểu dữ liệu> [tùy chọn thuộc tính], ..., <tên cột n> <kiểu dữ liệu> [AS])
BEGIN
    <Câu lệnh SQL>
RETURN
END
```

Thay đổi, xóa, xem nội dung Function

1. Thay đổi Function

Để thay đổi các hàm đã khai báo ta sử dụng câu lệnh ALTER FUNCTION. Cú pháp tương tự như tạo mới Function, chỉ thay từ khóa CREATE bằng từ khóa ALTER

```
ALTER FUNCTION <Tên function>
([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], ..., [...]])
RETURNS <kiểu dữ liệu> | TABLE
[WITH ENCRYPTION]
[AS]
BEGIN
    [Thân của hàm]
    RETURN <Biểu thức giá trị đơn> | Câu lệnh SQL
END
```

2. Xóa Function

Để xóa hàm ta dùng câu lệnh DROP FUNCTION.

```
DROP FUNCTION [schema_name.] <Tên function>
```

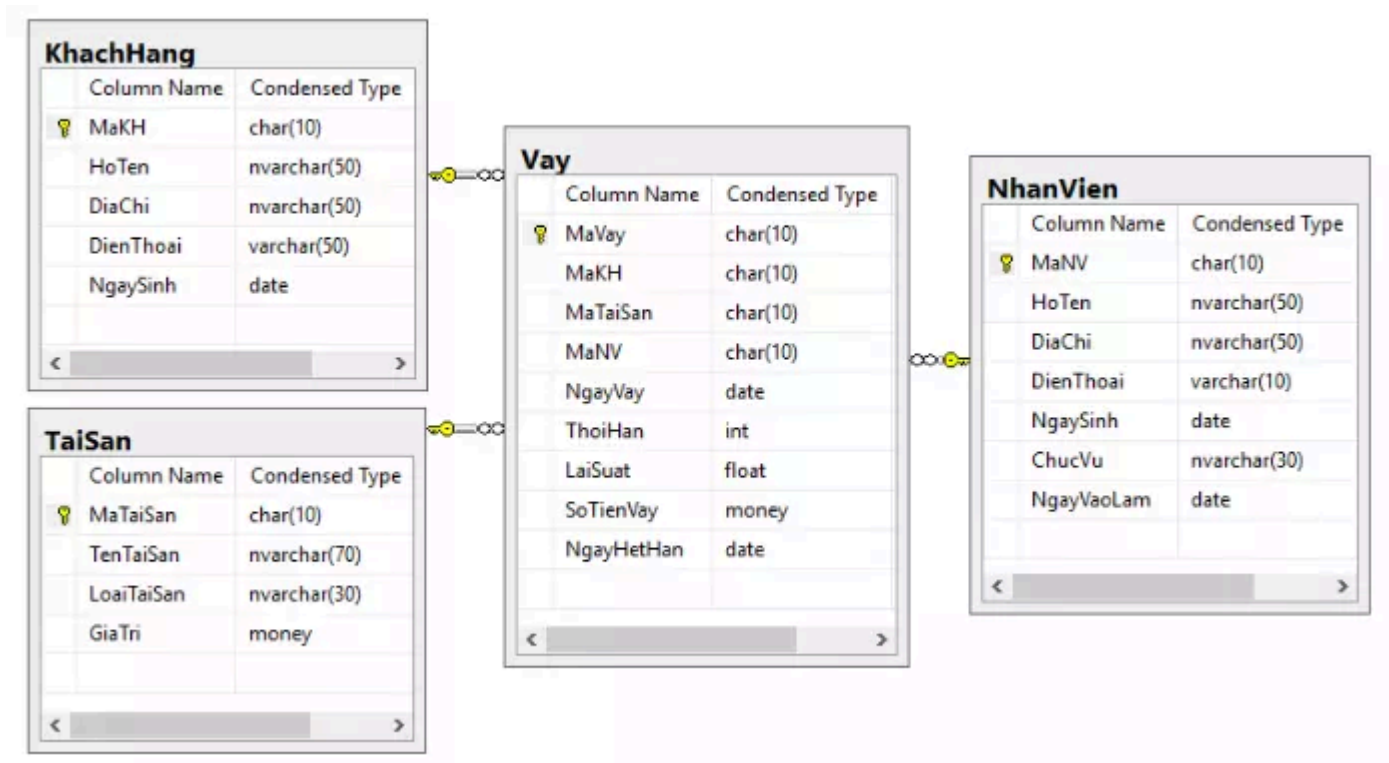
3. Xem nội dung Function

Để xem nội dung function ta sử dụng Store Procedure (Thủ tục) có sẵn của SQL là sp_helptext (Transact-SQL)

```
EXEC sp_helptext 'FunctionName'
```

Ví dụ cụ thể

Cho bài toán quản lý Vay có thể chấp tài sản đơn giản thể hiện qua sơ đồ mức vật lý như sau:



Ví dụ 1. Tạo function cho biết số lượng khách hàng theo địa chỉ bất kỳ nhận vào từ tham số với điều kiện là khách hàng có tổng số tiền vay từ trước đến nay từ 200 triệu trở lên.

Cách 1: Trả về giá trị vô hướng

```
CREATE FUNCTION count_customer_with_address (@address varchar(200))
RETURNS int
AS
BEGIN
    DECLARE @count int = 0
    SELECT @count = count(*) FROM (
        SELECT Vay.MaKH
        FROM Vay, KhachHang
        WHERE Vay.MaKH=KhachHang.MaKH AND KhachHang.DiaChi= @address
        GROUP BY Vay.MaKH
        HAVING SUM(Vay.SoTienVay)>=200
    ) AS Temp
    RETURN @count
END
```

Gọi thực thi function:

```
PRINT dbo.count_customer_with_address('Ha Noi')
```

Cách 2: Trả về giá trị table bằng câu lệnh đơn

```
CREATE FUNCTION count_customer_with_address2 (@address varchar(200))
RETURNS TABLE
AS
RETURN SELECT count(*) AS 'Total customers' FROM (
    SELECT Vay.MaKH
    FROM Vay, KhachHang
    WHERE Vay.MaKH=KhachHang.MaKH AND KhachHang.DiaChi= @address
    GROUP BY Vay.MaKH
    HAVING SUM(Vay.SoTienVay)>=200
) AS Temp
```

Gọi thực thi function:

```
SELECT * FROM dbo.count_customer_with_address2 ('Da Nang')
```

Cách 3: Trả về giá trị table bằng đa câu lệnh

```
CREATE FUNCTION count_customer_with_address3 (@address varchar(200))
RETURNS @new_table TABLE (DiaChi varchar(200), SoLuong int)
AS
BEGIN
    DECLARE @count int = 0
    SELECT @count = count(*) FROM (
        SELECT Vay.MaKH
        FROM Vay, KhachHang
        WHERE Vay.MaKH=KhachHang.MaKH AND KhachHang.DiaChi= @address
        GROUP BY Vay.MaKH
        HAVING SUM(Vay.SoTienVay)>=200
    ) AS Temp
    INSERT INTO @new_table VALUES (@address, @count)
```

Ví dụ 2. Sửa đổi function đã tạo ở trên bổ sung điều kiện khách hàng có tuổi từ 30 trở lên

```
ALTER FUNCTION count_customer_with_address (@address varchar(200))
RETURNS int
AS
BEGIN
    DECLARE @count int = 0
    SELECT @count = count(*) FROM (
        SELECT Vay.MaKH
        FROM Vay JOIN KhachHang ON Vay.MaKH=KhachHang.MaKH
        WHERE KhachHang.DiaChi= @address AND YEAR(GETDATE()) - YEAR(NgaySinh) >= 30
        GROUP BY Vay.MaKH
        HAVING SUM(Vay.SoTienVay)>=200
    ) AS Temp
    RETURN @count
END
```

Ví dụ 3. Xóa bỏ function đã tạo

```
DROP FUNCTION count_customer_with_address
```

Kết luận

Trên đây là những gì mình tổng hợp và tự đưa ra ví dụ minh họa cho từng phần. Hi vọng sẽ giúp ích cho các bạn trong quá trình sử dụng function trong câu lệnh sql.

- Function SQL
- Hàm SQL

All rights reserved



Bài viết liên quan

[JavaScript function](#)

[Vu Tong](#)
14 phút đọc
👁 6.4K 📌 8 💬 1 ⬆ 19

[Những câu lệnh SQL](#)

[Bùi Thị Ngọc](#)
9 phút đọc
👁 2.7K 📌 4 💬 0 ⬆ 3

[SQL testing \(Phần 1 - Ứng dụng của SQL trong testing\)](#)

[Phạm Thị Hong Ha](#)
5 phút đọc
👁 7.5K 📌 3 💬 0 ⬆ 0

[SQL Injection](#)

[Do Thi Minh Hoa](#)
19 phút đọc
👁 7.6K 📌 3 💬 0 ⬆ 0

[JavaScript function](#)

[Vu Tong](#)
14 phút đọc
👁 6.4K 📌 8 💬 1 ⬆ 19

[Những câu lệnh SQL](#)

[Bùi Thị Ngọc](#)
9 phút đọc
👁 2.7K 📌 4 💬 0 ⬆ 3

[SQL testing \(Phần 1 - Ứng dụng của SQL trong testing\)](#)

[Pham Thi Hong Ha](#)
5 phút đọc
👁 7.5K 📌 3 💬 0 ⬆ 0

[SQL Injection](#)

[Do Thi Minh Hoa](#)
19 phút đọc
👁 7.6K 📌 3 💬 0 ⬆ 3

Bài viết khác từ ThaoDTD

[MySQL Stored Procedures](#)

[ThaoDTD](#)
5 phút đọc
👁 11.1K 📌 2 💬 0 ⬆ 5

[i18n Rails](#)

[ThaoDTD](#)
9 phút đọc
👁 1.1K 📌 1 💬 1 ⬆ 4

[Transaction SQL \(Phần 1\).](#)

[ThaoDTD](#)
9 phút đọc
👁 22.0K 📌 8 💬 0 ⬆ 5

[SQL Azure](#)

[ThaoDTD](#)
8 phút đọc
👁 2.7K 📌 1 💬 0 ⬆ 1

[MySQL Stored Procedures](#)

[ThaoDTD](#)
5 phút đọc
👁 11.1K 📌 2 💬 0 ⬆ 5

[i18n Rails](#)

[ThaoDTD](#)
9 phút đọc
👁 1.1K 📌 1 💬 1 ⬆ 4

[Transaction SQL \(Phần 1\).](#)

[ThaoDTD](#)
9 phút đọc
👁 22.0K 📌 8 💬 0 ⬆ 5

[SQL Azure](#)

[ThaoDTD](#)
8 phút đọc
👁 2.7K 📌 1 💬 0 ⬆ 1

Bình luận

💬 Đăng nhập để bình luận

TÀI NGUYÊN

[Bài viết](#)

[Tổ chức](#)

[Câu hỏi](#)

[Videos](#)

[Thảo luận](#)

[Công cụ](#)

[Trạng thái hệ thống](#)

[Tags](#)

[Tác giả](#)

[Đề xuất hệ thống](#)

[Machine Learning](#)

DỊCH VỤ

 [Viblo](#)

 [Viblo Learning](#)

 [Viblo Code](#)

 [Viblo Partner](#)

 [Viblo CTF](#)

 [Viblo Battle](#)

 [Viblo CV](#)

 [Viblo Interview](#)

ỨNG DỤNG DI ĐỘNG



LIÊN KẾT

