



# BÀI 4: HÀM

📅 Ngày học @October 16, 2023

## 4.1 Hàm là gì?

- Tương tự như thủ tục, hàm là đối tượng của CSDL và khi có lời gọi thì các lệnh bên trong hàm sẽ được thực thi và trả kết quả về cho lời gọi hàm.
- Giữa hàm và thủ tục có sự khác biệt về cách sử dụng:
- Thủ tục được xem như 1 câu lệnh khi gọi và có thể sử dụng một cách độc lập. Kết quả của thủ tục được trả về cho Client.  
VD: sp\_who là một thủ tục của SQL Server => để thực hiện lời gọi, ta có thể viết lệnh:  
sp\_who  
hoặc: execute sp\_who
- Hàm khi sử dụng phải nằm bên trong một câu lệnh khác tại vị trí phù hợp với kiểu trả về của hàm. Kết quả của hàm không trả về cho Client mà trả về cho câu lệnh thực hiện lời gọi hàm.  
(Hàm không thể sử dụng một cách độc lập như thủ tục)  
VD: Hàm getdate() của SQL Server cho biết thời gian hiện tại của hệ thống. Để sử dụng hàm này, chúng ta không thể viết như sau:  
getdate()  
Có thể viết như sau:

```
SELECT getdate()  
hoặc:  
DECLARE @d date  
SET @d = getdate();  
PRINT @d;
```

Cũng tương tự như thủ tục, trong SQL Server có các hàm do hệ thống định nghĩa sẵn và người sử dụng có thể tự định nghĩa hàm. Trong SQL Server, hàm được chia thành 2 loại:

- Hàm scalar (scalar function):** Kết quả trả về của hàm là 1 giá trị thuộc vào các kiểu dữ liệu chuẩn (số, ngày, chuỗi,...)
- Hàm có dữ liệu trả về dạng bảng (table-valued function):** kết quả trả về của hàm là tập các dòng và các cột dưới dạng bảng, chia thành 2 loại:
  - Inline table-values function**
  - Multi-statement table-valued function**

## 4.2 Scalar function

- Hàm dạng này trả về 1 giá trị
- Có thể sử dụng tại những vị trí mà một biểu thức là được cho phép, phù hợp với kiểu dữ liệu trả về của hàm

```
Hàm YEAR(d) trong SQL Server là hàm dạng Scalar:  
SELECT YEAR('2023/10/16')
```

Để tạo hàm, sử dụng lệnh CREATE FUNCTION theo cú pháp:

```

💡 IF EXISTS (SELECT * FROM sys.objects WHERE name = 'Tên_hàm')
    DROP FUNCTION Tên_hàm;
GO
CREATE FUNCTION Tên_hàm (Danh_sách_tham_số)
RETURNS Kiểu_dữ_liệu_trả_về_của_hàm
AS
BEGIN
    DECLARE @biến_trả_về kiểu_data;
    SELECT @biến_trả_về = (thường sẽ là hàm gộp)
        FROM .....

    Phần thân của hàm
    RETURN @biến_trả_về;
END
GO

```

Trong đó:

- Cho dù có hay không có tham số, cặp dấu ( ) bắt buộc phải có sau tên hàm.
- Các tham số của hàm khai báo theo cú pháp:

💡 @Tên\_tham\_số Kiểu\_dữ\_liệu

và phân cách nhau bởi dấu phẩy.

- Với hàm, không thể sử dụng tham số kiểu bảng như thủ tục
- Phần thân của thủ tục: là nơi lập trình để xử lý và trả về kết quả về cho hàm, cần lưu ý:
  - Sử dụng lệnh:

💡 RETURN giá\_trị

để trả dữ liệu về cho hàm.

💡 **Không được sử dụng các lệnh có trả dữ liệu về cho Client (VD: SELECT) bên trong phần thân của hàm.**

- Tên hàm phải duy nhất trong CSDL. vì vậy các hàm do người định nghĩa nên đặt tên với tiền tố để phân biệt và tránh bị trùng tên (VD: fn\_, func\_, uf\_,...)

💡 Gọi hàm:  
SELECT dbo.Tên\_hàm(biến\_truyền)

VD: Viết hàm fn\_GetWeekDayName có chức năng trả về tên của thứ (bằng tiếng Việt) tương ứng với

```

IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetWeekDayName')
    DROP FUNCTION fn_GetWeekDayName;
GO

CREATE FUNCTION fn_GetWeekDayName
(
    @d date
)

```

```

RETURNS nvarchar(50)
AS
BEGIN
    DECLARE @weekDayName nvarchar(50);
    SET @weekDayName = CASE DATEPART(WEEKDAY, @d)
        WHEN 2 THEN N'Thứ hai'
        WHEN 3 THEN N'Thứ ba'
        WHEN 4 THEN N'Thứ tư'
        WHEN 5 THEN N'Thứ năm'
        WHEN 6 THEN N'Thứ sáu'
        WHEN 7 THEN N'Thứ bảy'
        ELSE N'Chủ nhật' -- 1
    END
    RETURN @weekDayName;
END
GO

-- Test
PRINT dbo.fn_GetWeekDayName('2023/10/16')
SELECT dbo.fn_GetWeekDayName(getdate())

```



Để xác định tên lược đồ CSDL (VD: dbo):

Vào CSDL → **Programmability** → **Functions** → **Scalar-valued Functions**

VD: Viết hàm fn\_GetRevenueOfMonth có chức năng tính tổng doanh thu bán hàng của một tháng thuộc một năm nào đó

```

IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetRevenueOfMonth ')
    DROP FUNCTION fn_GetRevenueOfMonth ;
GO

CREATE FUNCTION fn_GetRevenueOfMonth
(
    @month int,
    @year int
)
RETURNS money
AS
BEGIN
    IF @month NOT BETWEEN 1 AND 12 RETURN 0;
    DECLARE @fromDate date = DATEFROMPARTS(@year, @month, 1);
    -- EOMONTH(ngày cần lấy ngày cuối tháng, 0: nếu muốn lấy ngày cuối tháng của ngày hiện tại; 1: nếu lấy ngày cuối tháng của tháng tiếp...
    DECLARE @toDate date = EOMONTH(@fromDate, 0);
    DECLARE @totalRevenue money;
    SELECT @totalRevenue = SUM(od.Quantity * od.SalePrice)
        FROM Orders as o JOIN OrderDetails as od
        ON o.OrderId = od.OrderId
        WHERE -- MONTH(o.OrderDate) = @month AND YEAR(o.OrderDate) = @year
              o.OrderDate BETWEEN @fromDate AND @toDate;
    RETURN @totalRevenue
END
GO

SELECT dbo.fn_GetRevenueOfMonth(2, 2013)

```

### 4.3 Hàm với giá trị trả về kiểu bảng (table-valued function)

- Kết quả trả về của hàm là 1 bảng.
- Hàm dạng này có thể sử dụng ở những vị trí mà 1 bảng là được cho phép (**sau FROM**).
- Hàm dạng này chia thành 2 loại:

Inline function: là loại hàm mà kết quả trả về của hàm chỉ cần giải quyết bởi duy nhất 1 câu lệnh SELECT (**Trong phần thân của hàm chỉ là một câu lệnh SELECT**).

Multi-statement function: là loại hàm mà bên trong phần thân của hàm có thể sử dụng kết hợp nhiều lệnh để lập trình.

#### 4.3.1 Inline function

Cú pháp:

```
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'Tên_hàm')
    DROP FUNCTION Tên_hàm;
GO
CREATE FUNCTION Tên_hàm (Danh_sách_tham_số)
RETURNS TABLE
AS
RETURN
(
    Câu_lệnh_SELECT_trả_dữ_liệu_về_cho_hàm
)
GO
```

VD: Viết hàm fn\_GetRevenueByDates có chức năng trả về bảng cho biết doanh thu bán hàng từng ngày trong khoảng thời gian từ ngày @fromDate cho đến ngày @toDate (Chỉ cần thống kê các ngày có dữ liệu)

```
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetRevenueByDates')
    DROP FUNCTION fn_GetRevenueByDates;
GO
CREATE FUNCTION fn_GetRevenueByDates
(
    @fromDate date,
    @toDate date
)
RETURNS TABLE
AS
RETURN
(
    SELECT o.OrderDate, SUM(od.Quantity * od.SalePrice) AS Revenue
    FROM Orders as o JOIN OrderDetails as od
    on o.OrderId = od.OrderId
    WHERE o.OrderDate between @fromDate and @toDate
    GROUP BY o.OrderDate
)
GO
--Test
SELECT *
FROM dbo.fn_GetRevenueByDates('2017/12/1', '2017/12/31')
```

#### 4.3.2 Multi-statement function

Cú pháp:



```

IF EXISTS (SELECT * FROM sys.objects WHERE name = 'Tên_hàm')
    DROP FUNCTION Tên_hàm;
GO
CREATE FUNCTION Tên_hàm (Danh_sách_tham_số)
RETURNS @Tên_biến TABLE
(
    Cấu_trúc_của_bảng_chứa_dữ_liệu_trả_về
)
AS
BEGIN
    Các_lệnh_lập_trình_trong_phần_thân_của_hàm

    RETURN;
END
GO

```

LƯU Ý:

- Phải định nghĩa cấu trúc của bảng chứa dữ liệu trả về
- Cuối cùng của phần thân hàm phải là lệnh RETURN và chỉ có **duy nhất 1 lệnh RETURN** trong thân hàm. (Dữ liệu được return chính là @Tên\_biến).
- Trong quá trình lập trình trong hàm, phải đưa được dữ liệu vào @Tên\_biến.
- Trong thân hàm không được sử dụng câu lệnh có kiểu trả về cho Client.

VD: Viết hàm fn\_GetRevenueByFullDates có chức năng trả về dữ liệu cho biết doanh thu bán hàng từng ngày trong khoảng thời gian từ ngày @fromDate đến ngày @toDate. Yêu cầu phải đủ tất cả các ngày (không dùng CTE)

```

IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetRevenueByFullDates')
    DROP FUNCTION fn_GetRevenueByFullDates;
GO
CREATE FUNCTION fn_GetRevenueByFullDates
(
    @fromDate date,
    @toDate date
)
RETURNS @tbl TABLE
(
    SummaryDate date primary key,
    Revenue money
)
AS
BEGIN
    INSERT INTO @tbl(SummaryDate, Revenue)
        SELECT o.OrderDate, SUM(od.Quantity * od.SalePrice) AS Revenue
        FROM Orders as o JOIN OrderDetails as od
        on o.OrderId = od.OrderId
        WHERE o.OrderDate between @fromDate and @toDate
        GROUP BY o.OrderDate

    DECLARE @d date = @fromDate;
    WHILE @d <= @toDate
    BEGIN
        -- Lệnh Exists chỉ để kiểm tra có hay không có data (số dòng trả về)
        IF NOT EXISTS (SELECT * FROM @tbl WHERE SummaryDate = @d)
            INSERT INTO @tbl(SummaryDate, Revenue) VALUES(@d, 0)
        SET @d = DATEADD(DAY, 1, @d);
    END
    RETURN;
END
GO
--Test

```

```
SELECT *
FROM dbo.fn_GetRevenueByFullDates('2017/12/1', '2017/12/31')
```

## BÀI TẬP:

- Viết hàm fn\_GetFirstDateOfWeek để lấy ngày đầu tuần của một ngày bất kỳ (đầu tuần là thứ 2)  
VD: fn\_GetFirstDateOfWeek('2023/10/18') —> '2023/10/16'
- Viết hàm fn\_GetRevenueByDateOfMonth có chức năng trả về một bảng cho biết doanh thu bán hàng của từng ngày trong tháng @month năm @year. Yêu cầu phải thống kê đủ tất cả các ngày trong tháng.

Viết bằng 2 cách:

- Cách 1: Sử dụng inline function
  - Cách 2: Sử dụng multi-statement function
- Viết hàm fn\_GetUnsignString để chuyển một chuỗi ký tự có dấu (tiếng Việt) thành chuỗi không dấu

VD: fn\_GetUnsignString(N'Trường Đại học Khoa học')

—> 'Truong Dai hoc Khoa hoc'

```
-- C1
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetFirstDateOfWeek')
DROP FUNCTION fn_GetFirstDateOfWeek;
GO

CREATE FUNCTION fn_GetFirstDateOfWeek
(
    @day date
)
RETURNS date
AS
BEGIN
    DECLARE @ngayDauTuan date = DATEADD(DAY
        , CHOOSE(DATEPART(WEEKDAY, @day), -6, 0, -1, -2, -3, -4, -5)
        , @day);
    RETURN @ngayDauTuan
END
GO

SELECT DBO.fn_GetFirstDateOfWeek('2023/10/18')
```

```
-- Cách 1: Sử dụng inline function
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetRevenueByDateOfMonth')
DROP FUNCTION fn_GetRevenueByDateOfMonth;
GO

CREATE FUNCTION fn_GetRevenueByDateOfMonth
(
    @month int,
    @year int
)
RETURNS TABLE
AS
RETURN
(
    SELECT o.OrderDate, SUM(od.Quantity * od.SalePrice) AS Revenue
    FROM Orders as o JOIN OrderDetails as od
    on o.OrderId = od.OrderId
    WHERE MONTH(o.OrderDate) = @month and YEAR(o.OrderDate) = @year
    GROUP BY o.OrderDate
)
GO

SELECT *
FROM dbo.fn_GetRevenueByDateOfMonth(12, 2017)
```

```

Cách 2
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'fn_GetRevenueByDateOfMonth')
    DROP FUNCTION fn_GetRevenueByDateOfMonth;
GO

CREATE FUNCTION fn_GetRevenueByDateOfMonth
(
    @month int,
    @year int
)
RETURNS @DoanhThuNgay TABLE
(
    NgayOrder date,
    DoanhThu money
)
AS
BEGIN
    DECLARE @tblNgay Table
    (
        Ngay date
    )
    DECLARE @ngay date = DATEFROMPARTS(@year, @month, 1)
    WHILE (month(@ngay) = @month)
        BEGIN
            insert into @tblNgay values (@ngay);
            set @ngay = DATEADD(day, 1, @ngay);
        END
    INSERT @DoanhThuNgay (NgayOrder, DoanhThu)
    SELECT t1.Ngay, ISNULL (t2.revenue, 0) as revenue
    FROM @tblNgay AS t1
    LEFT JOIN
    (
        SELECT o.OrderDate, SUM(od.Quantity * od.SalePrice) as Revenue
        FROM Orders AS o
        join OrderDetails as od ON o.OrderId = od.OrderId
        WHERE month(o.OrderDate) = @month and YEAR(o.OrderDate) = @year
        GROUP BY o.OrderDate
    ) AS t2
    ON t1.Ngay = t2.OrderDate

    RETURN;
END
GO
SELECT *
FROM dbo.fn_GetRevenueByDateOfMonth(8, 2017)

```