



Bài 5: Hàm



Nội dung

- ❖ Biến cục bộ
- ❖ Cấu trúc điều khiển
- ❖ User-Defined Functions
- ❖ Các hàm hệ thống thông dụng

Biến cục bộ

Định nghĩa

- ❖ Biến được dùng để lưu trữ các giá trị tạm thời trong quá trình tính toán các xử lý
- ❖ Khi thoát khỏi chương trình hay tắt máy thì giá trị của các biến này không còn nữa.
- ❖ Trong T-SQL có 2 loại biến khác nhau: biến cục bộ và biến hệ thống (không có khái niệm về biến toàn cục).
- ❖ Trong phần này chúng ta chỉ đề cập đến biến cục bộ

Biến cục bộ

Khai báo

❖Cú pháp:

DECLARE @<tên biến> <kiểu dữ liệu>

❖Ví dụ:

DECLARE @TenNV Nvarchar(50), @luonggio int

❖Khác với một số ngôn ngữ lập trình khác, chúng ta không thể gán giá trị khởi tạo cho biến lúc khai báo chúng.

Biến cục bộ

Gán giá trị cho biến

- ❖ Sử dụng lệnh SET:

- ❖ Lệnh SET thường được dùng để gán giá trị cụ thể cho biến hoặc giá trị được lấy từ các bảng dữ liệu.

- ❖ Mỗi lệnh SET chỉ gán giá trị cho một biến.

- ❖ Ví dụ: Gán chuỗi 'Trần Thanh Phước' cho biến @TenNV

```
DECLARE @TenNV Nvarchar(50)  
SET @TenNV = N'Trần Thanh Phước'
```

- ❖ Ví dụ: Gán giá trị cho biến có tên là @luonggio, dữ liệu gán lấy từ bảng NhanVien.

```
DECLARE @luonggio int  
SET @luonggio = (SELECT MAX(luonggio) FROM Nhanvien)  
print @luonggio
```

Biến cục bộ

Gán giá trị cho biến

❖ Lệnh SELECT cũng thường được dùng để gán các giá trị cụ thể cho biến hoặc các giá trị được lấy ra hoặc tính toán từ dữ liệu của các cột bên trong các bảng dữ liệu.

❖ Mỗi lệnh SELECT có thể gán cùng lúc cho nhiều biến.

❖ Ví dụ: Tính tổng lương giờ, lương giờ lớn nhất của tất cả các nhân viên và gán cho biến có tên là @sumLG, @maxLG

```
DECLARE @sumLG int, @maxLG int
```

```
SELECT @sumLG=SUM(luonggio), @maxLG=MAX(luonggio)
```

```
FROM Nhanvien
```

```
print @sumLG
```

```
print @maxLG
```

Cấu trúc điều khiển

Toán tử

❖ Toán tử số học:

KÝ HIỆU	DIỄN GIẢI
+	Cộng 2 số
-	Trừ 2 số
*	Nhân 2 số
/	Chia 2 số
%	Chia lấy phần dư

❖ Toán tử so sánh:

KÝ HIỆU	DIỄN GIẢI
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
<>	Khác
!=	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

❖ Toán tử logic:

KÝ HIỆU	DIỄN GIẢI
AND	Và
OR	Hoặc
NOT	Phủ định
BETWEEN	Trong khoảng

Cấu trúc điều khiển

IF --- ELSE

❖Cú pháp:

IF <biểu thức luận lý>

<câu lệnh 1>|<khối lệnh 1>

[ELSE

<câu lệnh 2>|<khối lệnh 2>]

Cấu trúc điều khiển

IF --- ELSE

❖ Ví dụ: Kiểm tra xem nhân viên nào có lương giờ lớn hơn 20, nếu có thì in ra thông tin nhân viên đó. Nếu không thì xuất ra câu thông báo 'Không có nhân viên có lương giờ > 20'.

```
IF(SELECT COUNT(*) FROM Nhanvien
```

```
WHERE luonggio > 20) >0
```

```
  BEGIN
```

```
    PRINT'Danh sach cac nhan vien co luong gio > 20 la:'
```

```
    SELECT * FROM Nhanvien
```

```
    WHERE luonggio > 20
```

```
  END
```

```
ELSE
```

```
  PRINT'Khong co luong gio > 20'
```

manv	hoten	dienthoai	mail	luongGio	maphong	ngaysinh
NV07	Trần Kim Bình	32423423	binh@gmail.com	30	KT	1999-02-19

Cấu trúc điều khiển

IF --- ELSE

❖ Ngoài ra chúng ta có thể sử dụng từ khoá EXISTS kết hợp với cấu trúc IF để kiểm tra điều tồn tại các dòng dữ liệu trong bảng một cách hiệu quả hơn.

❖ **Cú pháp:**

IF EXISTS (Câu lệnh SELECT)

<Câu lệnh 1>|<Khối lệnh 1>

[ELSE

<Câu lệnh 2>|<Khối lệnh 2>]

Cấu trúc điều khiển

IF --- ELSE

❖ Ví dụ: Kiểm tra xem nhân viên nào có lương giờ lớn hơn 20, nếu có thì in ra thông tin nhân viên đó. Nếu không thì xuất ra câu thông báo 'Không có nhân viên có lương giờ > 20'.

```
IF(Exists(SELECT * FROM Nhanvien WHERE luonggio > 20))  
BEGIN
```

```
    PRINT 'Danh sach cac nhan vien co luong gio > 20 la:'
```

```
    SELECT * FROM Nhanvien
```

```
    WHERE luonggio > 20
```

```
END
```

```
ELSE
```

```
    PRINT 'Khong co luong gio > 20'
```

manv	hoten	dienthoai	mail	luongGio	maphong	ngaysinh
NV07	Trần Kim Bình	32423423	binh@gmail.com	30	KT	1999-02-19

Cấu trúc điều khiển

WHILE

❖Cú pháp:

WHILE <Biểu thức luận lý>

BEGIN

<Các lệnh lặp>

END

❖Biểu thức luận lý: Thông thường là các biểu thức so sánh mà các lệnh sẽ được lặp lại trong khi giá trị của biểu thức vẫn còn đúng.

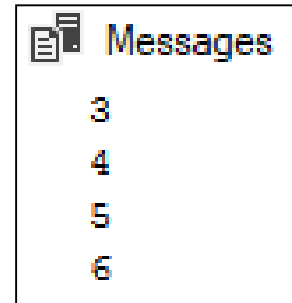
❖Các lệnh lặp: Các câu lệnh được thực hiện bên trong vòng lặp.

Cấu trúc điều khiển

WHILE

❖ Ví dụ: In ra màn hình 3, 4, 5, 6:

```
DECLARE @a int, @b int
SET @a=1
SET @b=@a+1
WHILE(@a<10)
BEGIN
    PRINT @a+@b
    IF(@a+@b=6)
        BREAK – Thoát khỏi vòng lặp
    SET @a=@a+1
END
```



User-Defined Functions

Định nghĩa

❖ Hàm do người dùng định nghĩa, không có sẵn trong SQL Server

❖ Cú pháp:

```
Create function [<shema name>.] function_name ( [ @parameter_name  
parameter_data_type] )  
returns [return data-type]  
As  
Begin  
    [Các câu lệnh]  
    Return [value]  
End
```

User-Defined Functions

Ví dụ

❖ Ví dụ 1: hàm tính tuổi

Create function ftuoi (@ns int)

Returns int

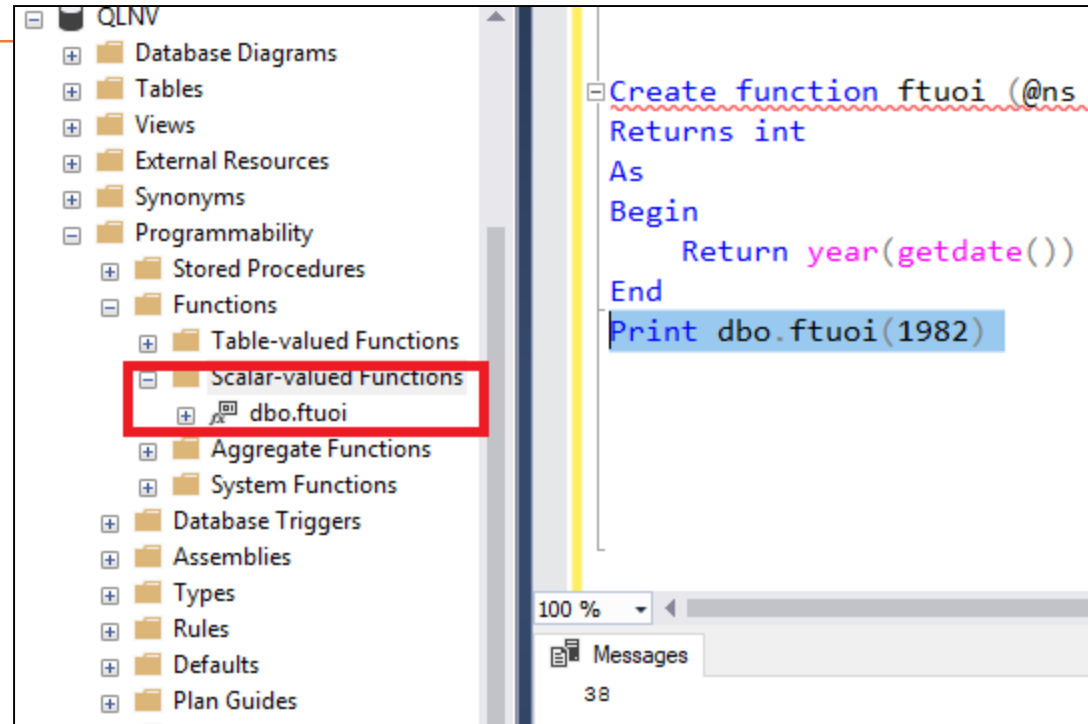
As

Begin

Return year(getdate()) - @ns

End

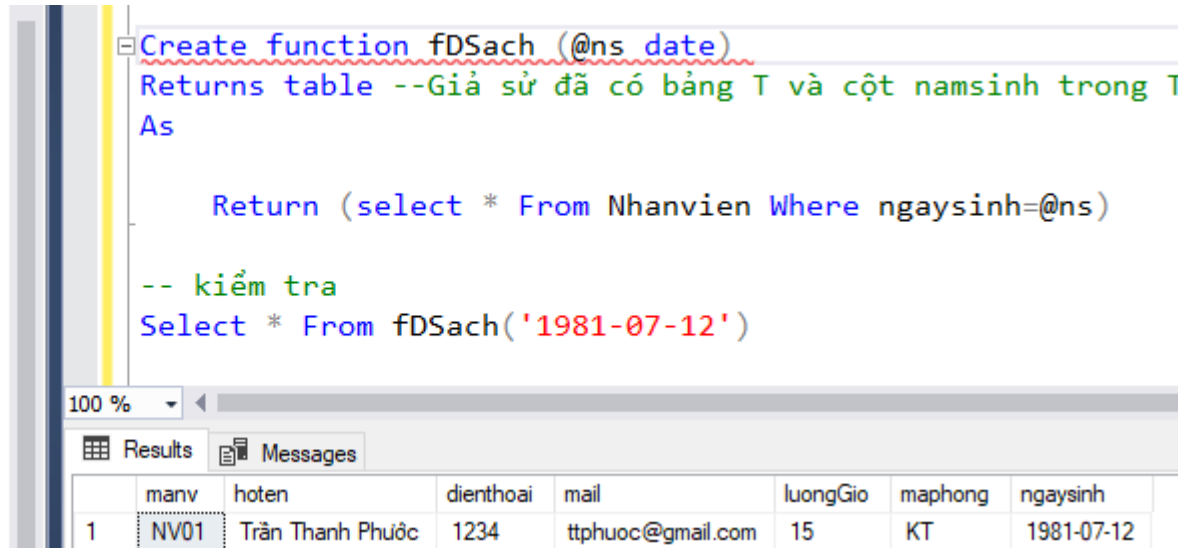
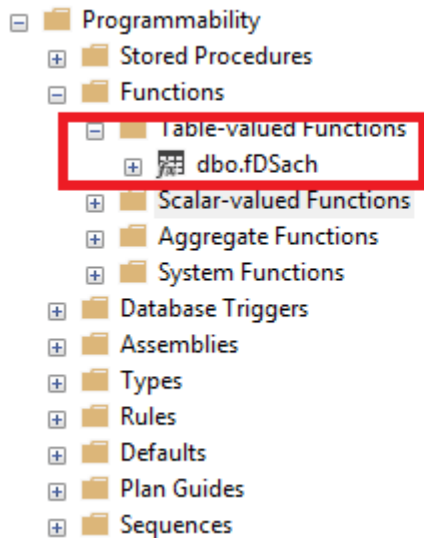
Print dbo.ftuoi(1982)



User-Defined Functions

Ví dụ

❖ Ví dụ 2: Tạo bảng tạm từ câu truy vấn (trả về Table)



Các hàm hệ thống thông dụng

Hàm chuyển đổi kiểu dữ liệu

❖ Cast:

❖ Chuyển đổi một biểu thức nào đó sang một kiểu dữ liệu bất kỳ theo yêu cầu

❖ Cú pháp:

CAST (<biểu thức> AS <kiểu dữ liệu>)

❖ Ví dụ:

SELECT Manv, hoten,

LuongGio= CAST(luonggio AS VARCHAR(10)) + ' USD'

FROM Nhanvien

	Manv	hoten	LuongGio
1	NV01	Trần Thanh Phước	15 USD
2	NV02	Nguyễn Minh Hải	10 USD
3	NV03	Hồ Văn Đức	20 USD
4	NV04	Phạm Thanh Huy	10 USD
5	NV05	Hồ Hồng Vân	5 USD
6	NV06	Lê Văn Hà	10 USD
7	NV07	Trần Kim Bình	30 USD

Các hàm hệ thống thông dụng

Hàm chuyển đổi kiểu dữ liệu

❖ Convert:

❖ Cho phép chuyển đổi một biểu thức nào đó sang một kiểu dữ liệu bất kỳ như mong muốn và có thể theo một định dạng nào đó.

❖ Cú pháp:

CONVERT(<kiểu dữ liệu>,<biểu thức>[,định dạng])

❖ Ví dụ:

```
SELECT hoten, CONVERT(VARCHAR(20), NGaySINH,103) as  
NgaySinh  
FROM Nhanvien
```

hoten	NgaySinh
Trần Thanh Phước	12/07/1981
Nguyễn Minh Hải	28/02/1983
Hồ Văn Đức	05/03/1973
Phạm Thanh Huy	20/12/1988
Hồ Hồng Vân	02/02/1981
Lê Văn Hà	19/02/2000
Trần Kim Bình	19/02/1999

Các hàm hệ thống thông dụng

Hàm chuyển đổi kiểu dữ liệu

❖ Convert:

Định dạng năm (yy)	Định dạng năm (yyyy)	Dạng hiển thị dữ liệu
1	101	mm/dd/yy
2	102	yy.mm.dd
3	103	dd/mm/yy
4	104	dd.mm.yy
5	105	dd-mm-yy
6	106	dd mon yy
7	107	mon dd yy
8	108	hh:mm:ss
9	109	mon dd yyyy hh:mm:ss
10	110	mm-dd-yy
11	111	yy/mm/dd
12	112	Yymmdd
13	113	dd mon yyyy hh:mm:ss
14	114	hh:mm:ss:mmm
	21 hoặc 121	yyyy-mm-dd hh:mi:ss:mmm
	20 hoặc 120	yyyy-mm-dd hh:mi:ss

Các hàm hệ thống thông dụng

Hàm chuyển đổi kiểu dữ liệu

❖STR: Cú pháp:

STR(<số thực>,<số ký tự>[,số lẻ])

❖Số thực: Là một biểu thức có kiểu dữ liệu số thực.

❖Số ký tự: Số khoảng trắng dùng để chứa các ký số sau khi chuyển sang kiểu dữ liệu chuỗi.

❖Số lẻ: Chỉ định số thập phân

❖Ví dụ:

PRINT STR(12.5, 10, 2)

Kết quả sẽ là: _ _ _ _ _12.50

Các hàm hệ thống thông dụng

Ngày giờ

❖ Bảng mô tả viết tắt của các đơn vị thời gian.

Từ viết tắt	Ý nghĩa	Miền giá trị
yy	Năm	1900 – 9999
qq	Quý	1 – 4
mm	Tháng	1 – 12
dd	Ngày trong tháng	1 – 31
dy	Ngày trong năm	1 – 366
wk	Tuần	1 – 53
dw	Ngày trong tuần	1 – 7
hh	Giờ trong ngày	0 – 23
mi	Phút trong giờ	0 – 59
ss	Giây trong phút	0 – 59
ms	Phần trăm mili giây	0 – 999

Các hàm hệ thống thông dụng

Ngày giờ

❖ DateAdd: Cộng

DATEADD(<đơn vị thời gian>, <số nguyên>, <ngày chỉ định>)

❖ Đơn vị thời gian: Là đơn vị dùng cho việc giảm hoặc tăng ngày (dd), tháng (mm), năm (yy)...

❖ Số nguyên: Là một số nguyên âm hoặc dương chỉ định việc giảm hoặc tăng theo đơn vị thời gian.

❖ Ngày chỉ định: Là một biểu thức, tên cột dữ liệu, giá trị cụ thể có kiểu dữ liệu ngày.

❖ Ví dụ: Tăng thêm 10 ngày đối với ngày 12/03/2020

PRINT DATEADD(dd, 10, '2020/03/12')

Mar 22 2020 12:00AM
22/03/2020

❖ Để có kết quả như định dạng dd/mm/yy chúng ta sử dụng kết hợp hàm convert như sau:

PRINT CONVERT(VARCHAR(10), DATEADD(dd, 10, '2020/03/12'), 103)

Các hàm hệ thống thông dụng

Ngày giờ

❖ DateDiff: Trừ

DATEDIFF(<đơn vị thời gian>, <ngày 1>, <ngày 2>)

❖ Đơn vị thời gian: Là đơn vị có thể là ngày (dd), tháng (mm), năm(yy)...

❖ Ngày 1, ngày 2: Là các biểu thức, tên cột dữ liệu, giá trị cụ thể có kiểu dữ liệu ngày.

❖ Ví dụ: In ra danh sách nhân viên (MANV, HOTEN, NGAYSINH) và tuổi của mỗi nhân viên chúng ta sử dụng hàm DATEDIFF như sau:

```
SELECT MANV, HOTEN, NGAYSINH =  
CONVERT(VARCHAR(10), NGAYSINH, 103),  
TUOI = DATEDIFF(yy, NGAYSINH, GETDATE())  
FROM NHANVIEN
```

MANV	HOTEN	NGAYSINH	TUOI
NV01	Trần Thanh Phước	12/07/1981	39
NV02	Nguyễn Minh Hải	28/02/1983	37
NV03	Hồ Văn Đức	05/03/1973	47
NV04	Phạm Thanh Huy	20/12/1988	32
NV05	Hồ Hồng Vân	02/02/1981	39
NV06	Lê Văn Hà	19/02/2000	20
NV07	Trần Kim Bình	19/02/1999	21

Các hàm hệ thống thông dụng

Ngày giờ

- ❖ GETDATE(): Ngày giờ hiện hành
- ❖ DATENAME(<đơn vị thời gian>,<ngày chỉ định>): Chuỗi thời gian đại diện của một ngày chỉ định theo một đơn vị thời gian bất kỳ.
- ❖ Đơn vị thời gian: Là đơn vị dùng để chỉ định lấy ra chuỗi thời gian đại diện.
- ❖ Ngày chỉ định: Là một biểu thức, tên cột dữ liệu hoặc giá trị cụ thể có kiểu dữ liệu ngày.
- ❖ Ví dụ:
 - ❖ Lấy ngày hiện hành
 - ❖ Thứ trong tuần của ngày hiện hành.

```
PRINT CONVERT(VARCHAR(10),GETDATE(),103)  
PRINT DATENAME(dw,getDate())
```

16/04/2020 Thursday

Các hàm hệ thống thông dụng

Ngày giờ

❖ DAY, MONTH, YEAR: Lấy về ngày tháng năm

❖ DAY(<ngày chỉ định>)

❖ MONTH(<ngày chỉ định>)

❖ YEAR(<ngày chỉ định>)

❖ Ví dụ:

```
print day(getDate())
```

```
print month(getDate())
```

```
print year(getDate())
```

16

4

2020

Các hàm hệ thống thông dụng

Toán học

- ❖ $\text{ABS}(\text{<biểu thức số>})$: Giá trị tuyệt đối
- ❖ $\text{POWER}(\text{<biểu thức số>}, \text{<số mũ>})$: Lũy thừa
- ❖ $\text{ROUND}(\text{<biểu thức số>}, \text{<Vị trí làm tròn>})$: Làm tròn
- ❖ $\text{SQRT}(\text{<biểu thức số>})$: căn bậc 2
- ❖ Ví dụ:

```
print ABS(-3)
print POWER(2,4)
print ROUND(4.165,1)
print SQRT(17)
```

```
3
16
4.200
4.12311
```

Các hàm hệ thống thông dụng

Chuỗi

- ❖ UPPER (chuỗi): Chuyển thành chuỗi hoa
- ❖ LOWER (chuỗi): Chuyển thành chuỗi thường
- ❖ LTRIM (chuỗi): Xóa khoảng trắng thừa bên trái chuỗi
- ❖ RTRIM (chuỗi): Xóa khoảng trắng thừa bên phải chuỗi
- ❖ LEN (chuỗi): Chiều dài chuỗi
- ❖ REVERSE (chuỗi): Đảo ngược chuỗi
- ❖ REPLACE(<chuỗi nguồn>,<chuỗi tìm>,<chuỗi thay thế>): Thay thế chuỗi

Các hàm hệ thống thông dụng

Chuỗi

❖ Ví dụ:

```
declare @s varchar(50)
set @s = '  Tran Thanh Phuoc  '
print @s
set @s = Ltrim(rtrim(@s))
print @s
print UPPER (@s)
print LOWER (@s)
print LEN (@s)
print REVERSE (@s)
print REPLACE(@s,'n','m')
```

```
Tran Thanh Phuoc
Tran Thanh Phuoc
TRAN THANH PHUOC
tran thanh phuoc
16
couhP hnahT narT
Tram Thamh Phuoc
```

Các hàm hệ thống thông dụng

Chuỗi

- ❖ LEFT(<chuỗi nguồn>, <số ký tự>)
- ❖ RIGHT(<chuỗi nguồn>, <số ký tự>)
- ❖ SUBSTRING(<chuỗi nguồn>, <vị trí>, <số ký tự>)

❖ Ví dụ:

```
declare @s varchar(50)
set @s = 'Tran Thanh Phuoc'
print left(@s, 4)
print right(@s, 7)
print substring (@s, 6, 5)
```

```
Tran
h Phuoc
Thanh
```



Bài tập

Làm bài tập 1 – Lab 4

**Thank you and
happy learning !!!**