



FUNCTIONS

(HÀM)

Khái niệm

- ❑ Một hàm trong SQL Server là một đối tượng trong CSDL, bao gồm một hoặc nhiều phát biểu T-SQL, thực hiện một số xử lý nào đó và trả về một giá trị trong lời gọi hàm qua tên hàm.
- ❑ SQL Server cung cấp sẵn nhiều hàm (Ví dụ: MONTH, DAY, YEAR,...) ngoài ra chúng ta có thể tạo thêm những hàm mới.
- ❑ Hàm có thể sử dụng như là một thành phần của biểu thức

Một số hàm thường dùng trong SQL – Toán học

- ❑ ABS(x) : Trị tuyệt đối của x
- ❑ SQRT(x) : Căn bậc hai của x
- ❑ SQUARE(x) : x^2
- ❑ POWER(y, x) : y^x
- ❑ LOG(x) : Logarit tự nhiên của x
- ❑ EXP(x) : Hàm mũ cơ số e của x: e^x .
- ❑ SIGN(x) : Lấy dấu của số x (-1: $x < 0$, 0: $x = 0$, +1: $x > 0$)
- ❑ ROUND(x,n) : Làm tròn tới n số lẻ.
- ❑ CEILING(x) : Số nguyên nhỏ nhất nhưng lớn hơn x
- ❑ FLOOR(X) : Số nguyên lớn nhất nhưng nhỏ hơn x

Một số hàm thường dùng trong SQL – Xử lý chuỗi

❑ Upper(str) : Chuyển sang chữ hoa

Ví dụ: Upper('hoa') → 'HOA'

❑ Lower(str) : Chuyển sang chữ thường

Ví dụ: Lower('DUNG') → 'dung'

❑ Len(str) : Trả về độ dài của chuỗi str

❑ Ltrim(str) : Trả về chuỗi sau khi bỏ kí tự trắng bên trái chuỗi str

❑ Rtrim(str) : Trả về chuỗi sau khi bỏ kí tự trắng bên phải chuỗi str

❑ Left(str,n): Trả về n kí tự bên trái chuỗi str

❑ Right(str,n): Trả về n kí tự bên phải chuỗi str

❑ Substring(str,m,n): Trả về chuỗi con, từ kí tự thứ m, lấy n kí tự trong chuỗi str

Ví dụ: Substring('abcdef',3,2) → 'cd'

Một số hàm thường dùng trong SQL – Hàm ngày tháng

- ❑ GetDate(): Trả về ngày giờ hệ thống
- ❑ Day(*date*): Trả về ngày trong tháng của ngày date
- ❑ Month(*date*): Trả về tháng của ngày date
- ❑ Year(*date*): Trả về năm của ngày date

<ul style="list-style-type: none">❑ DATEPART(datepart, date): trích ra phần tương ứng với datepart từ biểu thức ngày date.❑ DATEADD(datepart, number, date): trả về một ngày bằng biểu thức date cộng thêm với number.❑ DATEDIFF(datepart, date1, date2): trả về một số với đơn vị tính là datepart bằng date2 – date1.	Datepart	Abbreviations
	Year	yy, yyyy
	Quarter	qq, q
	Month	mm, m
	Day of year	dy, y
	Day of Month	dd, d
	Week of year	wk, ww
	Weekday	dw
	Hour	hh
	Minute	mi, n
	Second	ss, s
	Millisecond	Ms

- ❑ **Chú ý:** Để định lại thứ tự các phần của kiểu ngày cho phù hợp . Sử dụng lệnh: **Set Dateformat** *mdy | dmy | ymd | ydm | myd | dym*

Một số hàm thường dùng trong SQL – Hàm kiểm tra dữ liệu

- ❑ ISDATE(variable | column name): Kiểm tra dạng ngày hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.
- ❑ ISNUMERIC(variable | column name): Kiểm tra dạng số hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.
- ❑ ISNULL(expression, value): Trả về giá trị value nếu expression có giá trị NULL, ngược lại trả về giá trị của expression. Giá trị trả về cùng kiểu với expression.

Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu DL

❑ `Cast(expression AS data_type[(length)])`

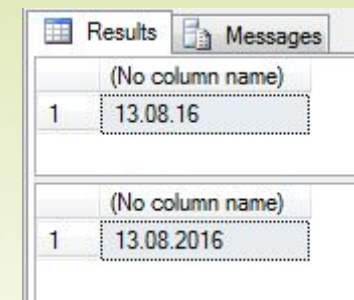
Ví dụ: `Cast(('20'+ '10') as Int) + 10 => 2020`

❑ `Convert(data_type[(length)], expression [, style])`

Style: Dạng thức kiểu ngày mà chúng ta muốn khi chuyển đổi dữ liệu kiểu datetime, datetime2 hoặc smalldatetime tới kiểu ký tự (char, varchar, nchar, or nvarchar). Hoặc dạng chuỗi mà chúng ta muốn khi chuyển dữ liệu kiểu số (float, real, money, or smallmoney) sang kiểu ký tự (char, varchar, nchar, or nvarchar).

Ví dụ:

```
Select CONVERT(varchar(10), GETDATE(), 4)  
Select CONVERT(varchar(10), GETDATE(), 104)
```



The screenshot shows the 'Results' tab of a SQL Server query window. It contains two query results. The first query uses style 4, resulting in the date '13.08.16'. The second query uses style 104, resulting in the date '13.08.2016'.

(No column name)	
1	13.08.16

(No column name)	
1	13.08.2016

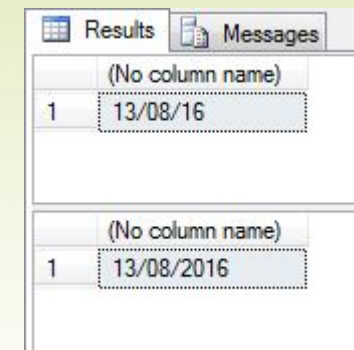
Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu DL

❑ Giá trị *Style*: để chuyển kiểu ngày giờ sang kiểu chuỗi ký tự

Without century (yy)	With century (yyyy)	Standard	Input/Output
-	0 or 100	Default	mon dd yyyy hh:miAM (or PM)
1	101	USA	mm/dd/yy
2	102	ANSI	yy.mm.dd
3	103	British/French	dd/mm/yy
4	104	German	dd.mm.yy
5	105	Italian	dd-mm-yy
6	106	-	dd mon yy
7	107	-	mon dd, yy
8	108	-	hh:mm:ss
-	9 or 109	Default+milliseconds	mon dd yyyy hh:mi:ss:mmmAM (or PM)
10	110	USA	mm-dd-yy

❑ Ví dụ:

```
Select CONVERT (varchar(10), GETDATE (), 3)
Select CONVERT (varchar(10), GETDATE (), 103)
```



(No column name)	
1	13/08/16
(No column name)	
1	13/08/2016

Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu DL

- ❑ Giá trị *Style*: để chuyển kiểu Float, Real sang kiểu chuỗi ký tự

0 (default)	Biểu diễn số nhỏ với số ký tự tối đa là 7, biểu diễn số lớn dưới dạng khoa học với độ dài cố định 12 ký tự
1	Luôn biểu diễn số dưới dạng khoa học với độ dài cố định 14 ký tự
2	Luôn biểu diễn số dưới dạng khoa học với độ dài cố định 22 ký tự

- ❑ Ví dụ:

```
declare @n float = 34211213643787.834728
select CONVERT(varchar(50),@n,0)
select CONVERT(varchar(50),@n,1)
select CONVERT(varchar(50),@n,2)
```

Results		Messages	
(No column name)			
1	3.42112e+013		
(No column name)			
1	3.4211214e+013		
(No column name)			
1	3.421121364378783e+013		

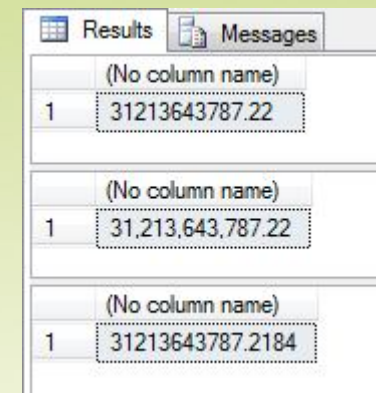
Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu DL

- ❑ *Giá trị Style*: để chuyển kiểu SmallMoney, Money sang kiểu chuỗi ký tự

0 (default)	Không dấu phẩy phân cách hàng ngàn, có 2 chữ số thập phân Ví dụ: 4235.98
1	Có dấu phẩy phân cách hàng ngàn và 2 chữ số thập phân Ví dụ: 3,510.92
2	Không dấu phẩy phân cách hàng ngàn, có 4 chữ số thập phân Ví dụ: 4235.9819

- ❑ Ví dụ:

```
declare @m money = 31213643787.218389
select CONVERT(varchar(50), @m, 0)
select CONVERT(varchar(50), @m, 1)
select CONVERT(varchar(50), @m, 2)
```



Results	
(No column name)	
1	31213643787.22
Messages	
(No column name)	
1	31,213,643,787.22
(No column name)	
1	31213643787.2184

Một số biến hệ thống thường dùng trong SQL

- ❑ Biến @@ROWCOUNT: Trả về số dòng bị ảnh hưởng bởi câu lệnh gần nhất
- ✓ Ví dụ: cập nhật dữ liệu với UPDATE và dùng @@ROWCOUNT để xác định số dòng đã được thay đổi.

Create Proc CapNhatDVT_SP

@masp int

As

UPDATE SanPham Set DVT = 'Kg' WHERE MaSP = @masp

IF @@ROWCOUNT = 0

Print 'Không có dòng nào được cập nhật!'

Go

Một số hàm hệ thống thường dùng trong SQL

❑ Biến **@@ERROR** chứa mã lỗi cho phát biểu vừa thực hiện. Nó tự động xóa và đặt lại mã lỗi với mỗi phát biểu được thực hiện. Giá trị **0** nếu phát biểu thành công. Có thể dùng biến **@@ERROR** để xác định lỗi hoặc làm điều kiện kết thúc thủ tục.

✓ Ví dụ: **Create Proc ThemCTHD**

@mahd char(5),

@masp int,

@sl int,

@dongia int

As

Insert Into CTHD(MaHD, MaSP, SL, DonGia) Values (@mahd,
@masp, @sl, @dongia)

If (@@Error <> 0)

Begin

RaisError(N'Lỗi thêm chi tiết hóa đơn',16,1)

Return 0

End

Go

Một số hàm hệ thống thường dùng trong SQL

- ❑ Phát biểu **RAISERROR** dùng xuất thông báo lỗi lấy từ table **sysmessages** hoặc một thông báo với nội dung nào đó

RAISERROR ({ *msg_id* | *msg_str* } { , *severity* , *state* } [, *argument* [,...*n*]])
[**WITH** *option* [,...*n*]]

- ✓ Ví dụ:

```
Insert Into CTHD(MaHD, MaSP, SL, DonGia) Values ('10434', 3, 10, 4000)
If (@@Error <> 0)
Begin
    RaisError(N'Lỗi thêm chi tiết hóa đơn',16,1)
Return 0
End
```

Một số kiểu hàm tự tạo

Một số kiểu hàm

- ❑ **Scalar-valued Functions:** tương tự như hàm có sẵn (built-in function), trả về một giá trị đơn.
- ❑ **Inline Table-valued Functions,** tương tự như view (nhưng có tham số), trả về một table là kết quả của một lệnh SELECT.
- ❑ **Multi-statement Table-valued Functions,** tương tự như Stored procedure, trả về một Table mới là kết quả của lệnh INSERT.

Tạo Scalar-valued Function

- ❑ Trả về một giá trị đơn thuộc một kiểu dữ liệu nào đó (int, float, char, varchar,...)

Cú pháp:

```
CREATE FUNCTION <function_name> (@param_1 Data_type_1, @param_2 Data_type_2, ....)
RETURNS <Function_Data_Type>
AS
BEGIN
    ....
    RETURN <@resultVar>
END
```

- ❑ Trong đó:

- *function_name*: Tên hàm
- *@param_i Data_type_i*: tên các tham số - kiểu dữ liệu của tham số Input thứ i nếu có.
- *<Function_Data_Type>*: kiểu dữ liệu của kết quả hàm trả về
- *<@resultVar>*: giá trị hàm trả về

Tạo Scalar-valued Function

- ✓ Ví dụ tạo một Scalar Function: Tạo hàm trả về tổng trị giá của một hóa đơn truyền vào
`Create Function TriGiaHD (@mahd char(5)) Returns int`

`As`

`Begin`

`Declare @kq int`

`Select @kq = (Select Sum(SoLuong*DonGia) From CHTD Where MaHD = @mahd)`

`If @kq is Null`

`Set @kq=0`

`Return @kq`

`End`

- ✓ **Gọi hàm:** `Select dbo.TriGiaHD('10144')`

	MaHD	MaSP	SoLuong	DonGia
	10144	1	35	374
	10144	3	10	16
	10144	6	20	39
	10145	1	12	374
	10145	2	8	65
	10145	6	20	39
	10148	7	24	8
	10148	8	12	101
	10150	6	9	39
	10156	1	25	374
	10156	2	25	65

Results		Messages
(No column name)		
1	14030	

Tạo Inline Table-valued Function

❑ Cú pháp:

CREATE FUNCTION <Function_Name> (@param1 Data_type1, @param2 Data_type2,)

RETURNS TABLE

AS

RETURN

(

SELECT

)

Thân của hàm

❑ Trong đó:

- *Table*: là kiểu dữ liệu trả về của hàm.
- *Thân của hàm*: là lệnh Return với một câu lệnh Select duy nhất.

Tạo Inline Table-valued Function

- ✓ Ví dụ: Tạo hàm ChiTietHD hiển thị thông tin chi tiết của một hóa đơn được truyền vào.

Create Function ChiTietHD(@mahd char(5)) Returns Table

As

Return

(

Select CTHD.MaHD, CTHD.MaSP, TenSP, NgayLapHD, SoLuong, DonGia

From HoaDon Inner Join CTHD On HoaDon.MaHD = CTHD.MaHD Inner Join SanPham On

CTHD.MaSP = SanPham.MaSP

Where CTHD.MaHD = @mahd

)

Go

- ✓ Gọi thực thi hàm: Select * From dbo.ChiTietHD('10144')

	MaHD	MaSP	TenSP	NgayLapHD	SoLuong	DonGia
1	10144	1	Rượu	1997-01-06 00:00:00.000	35	374
2	10144	3	Bánh kem	1997-01-06 00:00:00.000	10	16
3	10144	6	Nem	1997-01-06 00:00:00.000	20	39

Tạo Multi-statement Table-valued Function

❑ Cú pháp:

CREATE FUNCTION <Function_Name> (<@param1> <data_type1>, <@param2>...)

RETURNS <@Table_Variable_Name> TABLE (<Column_1> <data_Type1>, <column2> ...)

AS

BEGIN

Biến đại diện cho Table kết quả

Định nghĩa các cột của Table kết quả

-- Các lệnh điền (các dòng) vào table kết quả

RETURN

END

❑ Trong đó:

- Biến <@Table_Variable_Name> trong mệnh đề RETURNS có phạm vi sử dụng trong hàm, được sử dụng như một tên bảng, chứa kết quả trả về của hàm.
- Cấu trúc bảng kết quả trả về bởi hàm được xác định dựa vào định nghĩa các cột của bảng trong phát biểu RETURNS.

Tạo Multi-statement Table-valued Function

- ✓ Tạo hàm cho biết số lượng sinh viên mỗi ngành trong từng khóa học, gồm có 03 khóa (01, 02, 03), tùy theo tham số đầu vào là khóa học nào.

Create Function SLSV_Nganh_KhoaHoc(@khoadt char(10))

Returns @kq Table

```
(  
    MaNganh char(7),  
    SLSV int  
)
```

As

Begin

Insert Into @kq Select MaNganh, Count(MaSV) As SLSV From SinhVien

Where KhoaHoc = @khoadt

Group By MaNganh

Return

End

BAOBAO-PC.QLDaoTao - dbo.SinhVien		SQLQuery1.sql - BA...C.master (sa (53))						
	MaSV	HoSV	TenSV	PhaiNu	NgaySinh	MaTinh	MaNganh	KhoaHoc
	01002	Nguyễn Thị Bình	Yên	True	010587	TP	KT	01
	01004	Đào Thúy	Phượng	True	290387	BP	KT	01
	01005	Đặng Bình	Bình	False	300488	TN	HTT	01
	01006	Huỳnh Tấn	Dũng	False	080188	BT	HTT	01
	02001	Lý Thúy	Nga	True	240288	CM	KT	02
	02002	Ngô Thị Trùng	Hương	True	140488	TN	QT	02
	02003	Ngô Trường	An	False	271088	TN	KT	02
	02004	Nguyễn Dân	Trí	False	090887	TN	KT	02
	02006	Lê Thị Thúy	An	True	270488	TG	CK	02
	03003	Ngô Thuý	Lê	True	280588	TG	QT	03
	03004	Nguyễn Châu	Thanh	False	151088	BT	CK	03
	03005	Nguyễn Thị Mỹ	Anh	True	091188	BT	CK	03

Tạo Multi-statement Table-valued Function

SQLQuery4.sql - B...LDaoTao (sa (53))*

```
Create Function SLSV_Nganh_KhoaHoc (@khoa dt char(10))
Returns @kq Table
(
    MaNghanh char(7),
    SLSV int
)

As
Begin
    Insert Into @kq Select MaNghanh, Count (MaSV) As SLSV From SinhVien
    Where KhoaHoc = @khoa dt
    Group By MaNghanh
    Return
End
Go

Select * From dbo.SLSV_Nganh_KhoaHoc ('01')
Select * From dbo.SLSV_Nganh_KhoaHoc ('02')
Select * From dbo.SLSV_Nganh_KhoaHoc ('03')
```

Results Messages

	MaNghanh	SLSV
1	HTT	2
2	KT	2

Khóa 01

	MaNghanh	SLSV
1	CK	1
2	KT	3
3	QT	1

Khóa 02

	MaNghanh	SLSV
1	CK	2
2	QT	1

Khóa 03

Tạo Multi-statement Table-valued Function

- ✓ Ví dụ: Tạo hàm ThongKe trả về tổng số lượng mỗi mặt hàng đã nhập, đã xuất tùy theo giá trị của tham số truyền vào ('Nhap' hay 'Xuat').

Create Function ThongKe (@tuychon char(4))

Returns @kq Table

```
(  
    MaMH char(20),  
    TongSoLG int  
)
```

As

Begin

If @tuychon = 'Nhap'

Insert Into @kq Select MaMH Sum(SLNhap) as TongSoLG From CTPN

Group By MaMH

Else

If @tuychon = 'Xuat'

Insert Into @kq Select MaMH, Sum(SLXuat) as TongSoLG From CTPX

Group By MaMH

Return

End

Gọi Hàm

Select * From dbo.ThongKe('Nhap')

Select * From dbo.ThongKe('Xuat')