

HÀM

1. Các hàm thông dụng.

a. Các hàm chuyển đổi kiểu dữ liệu.

Hàm 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: PRINT 'Số nguyên tố' + CAST(123 AS VARCHAR(3))

Ví dụ:

```
SELECT MAMH, TENMH, TONG_SO_TIET= 'So tiet:' + CAST( SOTIET AS  
VARCHAR(3))  
FROM MONHOC
```

Hàm CONVERT: 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:

- Định dạng: Là một số nguyên dương chỉ định việc chuyển đổi kiểu dữ liệu ngày sang dạng chuỗi. Bảng bên dưới mô tả một số định dạng thường dùng trong hàm 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

Ví dụ:

```
SELECT HOTEN, CONVERT(VARCHAR(6), NGSINH,109)
FROM SINHVIEN
```

Hàm STR: chuyển đổi kiểu dữ liệu số sang kiểu dữ liệu chuỗi. Phải đảm bảo vùng trống để chứa các ký số khi chuyển đổi sang kiểu dữ liệu chuỗi.

Cú pháp:

STR(<số thực>,<số ký tự>[,<số lẻ>])

Ví dụ: PRINT STR(12.5, 10, 2)

Kết quả sẽ là: _____12.50

Ví dụ: In danh sách sinh viên (MASV, HOTEN), mã môn học (MAMH) và kết quả học tập (DIEM) của từng sinh viên trong đó cột điểm được định dạng có 1 chữ số thập phân.

```
SELECT SINHVIEN.MASV, HOTEN, MAMH, DIEM_MH=STR(DIEM,4,1)
FROM SINHVIEN, KETQUA
WHERE KETQUA.MASV=SINHVIEN.MASV
```

Kết quả trả về như sau:

	MASV	HOTEN	MAMH	DIEM_MH
1	SV1	Nguyen Tuan Anh	m01	6.0
2	SV2	Tran Thanh Nam	m01	8.0
3	SV2	Tran Thanh Nam	m02	7.0
4	SV3	Duong Minh Tuan	m01	9.0
5	SV3	Duong Minh Tuan	m02	4.0
6	SV3	Duong Minh Tuan	m03	5.0

b. Các hàm xử lý 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
----	---------------------	---------

Hàm DATEADD: cho phép cộng hoặc trừ một ngày chỉ định cho một đơn vị thời gian nào đó và trả về một ngày mới theo yêu cầu.

Cú pháp:

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

Ví dụ: Tăng thêm 10 ngày đối với ngày 12/03/2007

```
PRINT DATEADD(dd, 10, '12/03/2007')
```

Kết quả như sau: **Mar 22 2007 12:00AM**

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, '12/03/2007'), 103)
```

Kết quả như sau: 22/03/2007

Hàm DATEDIFF: tính khoảng cách (hiệu) của 2 ngày bất kỳ và có kết quả trả về là một số nguyên.

Cú pháp:

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

Ví dụ: Tính hiệu giữa 2 ngày: 12/05/2007 và 02/06/2007

```
PRINT DATEDIFF(dd, '12/05/2007', '02/06/2007')
```

Kết quả trả về là 21

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

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

Hàm GETDATE: lấy ngày giờ hiện tại theo ngày giờ của hệ thống .

Cú pháp: GETDATE()

Hàm DATENAME: trả về là một chuỗi thời gian đại diện của một ngày chỉ định theo một đơn vị thời gian bất kỳ.

Cú pháp:

DATENAME(<đơn vị thời gian>, <ngày chỉ định>)

Ví dụ: Cho biết thứ trong tuần của ngày 22/08/2007

```
PRINT DATENAME(dw, '22/08/2007')
```

Kết quả trả về là: Wednesday

Hàm DATEPART: trả về là một số nguyên chỉ định thời gian đại diện của một ngày theo một đơn vị thời gian bất kỳ.

Cú pháp:

DATEPART(<đơn vị thời gian>,<ngày chỉ định>)

Ví dụ: PRINT DATEPART(yy,'22/08/2007')

Kết quả trả về là: 2007

Các hàm DAY, MONTH, YEAR.

Các hàm DAY, MONTH, YEAR có kết quả trả về là một số nguyên chỉ định ngày (day), tháng

c. Các hàm toán học.

Hàm ABS

Cú pháp:

ABS(<biểu thức số>)

Hàm PI

Cú pháp: PI()

Hàm POWER

Hàm POWER được dùng để tính lũy thừa của một số nào đó theo một số mũ chỉ định trước.

Cú pháp:

POWER(<biểu thức số>,<số mũ>)

Hàm RAND

Hàm RAND có kết quả trả về là một số thực ngẫu nhiên mà hệ thống Microsoft SQL Server tự động tạo ra đảm bảo không trùng lặp.

Cú pháp:

RAND([số nguồn])

Ví dụ: PRINT RAND(14) có kết quả trả về là: 0.713834

Hàm ROUND

Cú pháp:

ROUND(<biểu thức số>,<Vị trí làm tròn>)

Ví dụ: Thực hiện làm tròn số 243.6531 với các hình thức sau:

PRINT ROUND(243.6531,1)

PRINT ROUND(243.6531,2)

```
PRINT ROUND(243.6531,-1)
Kết quả lần lượt là:  243.7000
                     243.6500
                     240.0000
```

Hàm SQRT

Hàm SQRT được dùng để tính căn bậc 2 của một số bất kỳ và kết quả trả về là một số dương.

Cú pháp: SQRT(<biểu thức số>)

Ví dụ: PRINT SQRT(81) có kết quả trả về là 9

d. Các hàm xử lý chuỗi.

Find (<a>, <s>): trả về vị trí đầu tiên của chuỗi <a> trong chuỗi <s>.

CharIndex (<ký tự>, <chuỗi>): trả về vị trí đầu tiên của <ký tự> trong <chuỗi>.

Ví dụ: charIndex (' ', 'Ly Hung') -> 3

Các hàm UPPER, LOWER

Hàm UPPER và LOWER được dùng để chuyển đổi một chuỗi ký tự thành chữ in (UPPER) hoặc chữ thường (LOWER).

Cú pháp: UPPER(<chuỗi dữ liệu>)

LOWER(<chuỗi dữ liệu>)

Ví dụ: In danh sách họ tên sinh viên bằng chữ in, địa chỉ bằng chữ thường.

```
SELECT TEN_SV=UPPER(HOTEN), DIA_CHI=LOWER(DCHI)
FROM SINHVIEN
```

Các hàm LEFT, RIGHT, SUBSTRING

Các hàm này có kết quả trả về là một chuỗi con được trích ra từ một chuỗi nguồn cho trước. Chuỗi con được trích ra bắt đầu từ vị trí bên trái (LEFT), bên phải (RIGHT), hoặc tại bất kỳ một vị trí nào (SUBSTRING).

Cú pháp:

LEFT(<chuỗi nguồn>, <số ký tự>)

RIGHT(<chuỗi nguồn>, <số ký tự>)

SUBSTRING(<chuỗi nguồn>, <vị trí>, <số ký tự>)

Ví dụ: Sử dụng các hàm trích chuỗi nguồn "Microsoft SQL Server "

```
PRINT LEFT('Microsoft SQL Server',9)
PRINT SUBSTRING('Microsoft SQL Server',11,3)
PRINT RIGHT('Microsoft SQL Server',6)
```

Kết quả như sau: Microsoft

SQL Server

Các hàm LTRIM, RTRIM

Các hàm này có kết quả trả về là một chuỗi đã cắt bỏ khoảng trắng ở đầu chuỗi (LTRIM) hoặc các khoảng trắng ở cuối chuỗi (RTRIM).

Cú pháp:

LTRIM(<chuỗi dữ liệu>)

RTRIM(<chuỗi dữ liệu>)

Ví dụ:

```
PRINT('  HELLO  WORLD')
PRINT LTRIM('  HELLO  WORLD')
PRINT RTRIM('HELLO  ')+ 'WORLD'
```

Hàm SPACE

Hàm SPACE có kết quả trả về là một chuỗi chứa N ký tự trắng.

Cú pháp: SPACE(N)

Trong đó:

Ví dụ: Sử dụng hàm SPACE để tạo ký tự trắng như sau:

```
PRINT 'HELLO'+ SPACE(3)+'WORLD'
```

Kết quả trả về là: HELLO WORLD

Hàm REPLICATE

Hàm này có kết quả trả về là một chuỗi chứa các ký tự được lặp lại N lần.

Cú pháp:

REPLICATE(<chuỗi lặp>,N)

Ví dụ: Thực hiện lặp lại chuỗi 'Hello' 3 lần:

```
PRINT REPLICATE('HELLO! ',3)
```

Kết quả như sau: HELLO! HELLO! HELLO!

Hàm LEN

Hàm LEN được sử dụng để lấy chiều dài của một chuỗi. Kết quả trả về là một số nguyên dương cho biết chiều dài chuỗi.

Cú pháp: LEN(<chuỗi dữ liệu>)

Ví dụ: Tính chiều dài của chuỗi 'Công nghệ thông tin'

```
PRINT LEN('Cong nghe thong tin')
```

Kết quả trả về là: 19

Hàm REVERSE

Hàm REVERSE được dùng để đảo ngược một chuỗi cho trước.

Cú pháp: REVERSE(<chuỗi dữ liệu>)

Ví dụ: Đảo ngược chuỗi sau: 'TIN HOC'

```
PRINT REVERSE('TIN HOC')
```

Kết quả trả về là: COH NIT

3.1.1.1.Hàm STUFF

Hàm STUFF có kết quả trả về là một chuỗi mới sau khi đã huỷ bỏ một số ký tự hiện có và thêm vào một chuỗi con khác tại vị trí vừa huỷ bỏ.

Cú pháp:

```
STUFF(<chuỗi nguồn>,<vị trí>,<chiều dài>,<chuỗi con>)
```

Trong đó:

- Chuỗi nguồn: Là một chuỗi dữ liệu.
- Vị trí: Là một số nguyên chỉ định vị trí bắt đầu huỷ bỏ các ký tự bên trong chuỗi nguồn.
- Chiều dài: Là một số nguyên chỉ định bao nhiêu ký tự sẽ bị huỷ bỏ bên trong chuỗi nguồn được đếm từ ký tự đầu tiên bên trái chuỗi nguồn.
- Chuỗi con: Là một chuỗi sẽ được thêm vào chuỗi nguồn tại vị trí huỷ bỏ ở trên.

Ví dụ: Sử dụng hàm STUFF như sau:

```
PRINT STUFF('123456789',3,4,'ABC')
```

Kết quả trả về như sau: 12ABC789

Hàm REPLACE

Hàm REPLACE cho phép tìm và thay thế (nếu có) một chuỗi con trong chuỗi nguồn bằng một chuỗi khác.

Cú pháp:

```
REPLACE(<chuỗi nguồn>,<chuỗi tìm>,<chuỗi thay thế>)
```

Trong đó:

- Chuỗi tìm: Là chuỗi con cần tìm xem có xuất hiện trong chuỗi nguồn hay không.

- Chuỗi thay thế: Là chuỗi sẽ được thay thế khi tìm thấy chuỗi tìm trong chuỗi nguồn.

Ví dụ: Thực hiện việc tìm kiếm và thay thế như sau:

```
PRINT REPLACE('MON HOC THCB','THCB','CSDL')
```

Kết quả trả về như sau: MON HOC CSDL

Hàm CHAR

Hàm CHAR có kết quả trả về là một ký tự tương ứng trong bảng mã ASCII

Cú pháp: CHAR(số nguyên)

Ví dụ: PRINT CHAR(35) có kết quả là: #

Hàm ASCII

Hàm ASCII có kết quả trả về là một số nguyên có phạm vi từ 0-255 tương ứng trong bảng mã ASCII, tham số đầu vào là một ký tự.

Cú pháp: ASCII(<ký tự>)

Ví dụ: PRINT ASCII('A') có kết quả trả về là: 65

2. Hàm tự tạo

Cú pháp

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

Ví dụ 1: hàm tính tuổi

```
Create function ftuoi (@ns int)  
    Returns int  
  
As  
Begin  
    Return year(getdate()) - @ns  
End  
Go  
Print dbo.ftuoi(1982)
```

Ví dụ 2: tạo bảng tạm từ câu truy vấn

```
Create function fDSach (@ns int)  
Returns table  
--Giả sử đã có bảng T và cột namsinh trong T
```



```

As
Return (select * From T Where namsinh=@ns)
go
-- kiểm tra
Select * From fDSach(1982)

```

3. Bài tập có hướng dẫn

Cho CSDL quản lý sinh viên (file buoi8_GuiSV.sql)

```

Lop (malop, tenlop)
Sinhvien (masv, hoten, ngaysinh, malop)
Monhoc (mamh, tenmh)
Ketqua (masv, mamh, diem)

```

Yêu cầu: Viết các hàm sau:

- Trả về họ tên và ngày sinh của sinh viên dựa vào mã lớp, trong đó phải viết hoa ký tự đầu tiên của họ tên, ngày sinh hiển thị theo dạng dd/mm/yyyy
- Hiển thị kết quả với điểm gồm 2 chữ số thập phân
- Hiển thị họ tên, ngày sinh, thứ (trong tuần, tiếng Anh) mà sinh viên được sinh ra
- Hiển thị thông tin sinh viên gồm: masv, họ lót, tên, năm sinh của sinh viên.
- Trả về mã sinh viên lớn nhất khi cho biết mã lớp. Cho biết: Masv = <malop> + “số thứ tự” (gồm 3 số). Ví dụ: lớp “TH01” thì masv có thể là: TH01001, TH01002; trong đó mã sinh viên TH01002 là mã sinh viên lớn nhất.
- Thêm một sinh viên mới. Chỉ cần truyền vào các tham số: họ tên, ngày sinh và mã lớp; mã sinh viên sẽ tự động phát sinh và được thêm vào. *Viết thủ tục*
- Hiển thị các sinh viên có điểm trung bình ≥ 5 . Thông tin gồm: mã sinh viên, họ tên, điểm trung bình (điểm trung bình gồm 1 chữ số thập phân)

Bài giải:

```

--a
Create function caua (@ml varchar(10))
Returns table
As
Return (select (Upper(left(hoten,1)) +
substring(hoten,2,len(hoten)-1)) as hoten, CONVERT(VARCHAR(10),
ngaysinh,103) as Ngaysinh
From sinhvien Where malop=@ml)
go
--drop function caua
select * from caua ('TH01')
--b
create function caub()
returns table
as
return (select masv, mamh, str(diem,5,2) as diem from ketqua)
go
select * from caub()
--c

```

```

Create function cauc()
Returns table
As
    Return (select hoten, ngaysinh, datename(dw,ngaysinh) as
Thu from sinhvien)
go

select * from cauc()
--d
Create function caud ()
Returns table
As
    Return (select masv,
reverse(substring(reverse(hoten),charindex(' ',reverse(hoten))+1,
len(hoten)- len(left(reverse(hoten),charindex('
',reverse(hoten)))))) as Holot,
    reverse(left(reverse(hoten),charindex(' ',reverse(hoten)))) as
ten from sinhvien)
go
select * from caud()

-- cau e
drop function caue
Create function caue (@ml varchar(10))
Returns varchar(10)
As
Begin
    return(select top 1 masv from sinhvien where malop = @ml
order by masv desc)
End
go
--drop function caul
print dbo.caue('TH01')
--f
Create proc cauf @ht nvarchar(30), @ns datetime, @ml varchar(10)
As
    declare @masv varchar(10)
    set @masv = dbo.caue(@ml)
    declare @stt int
    set @stt = cast(right(@masv,3) as int) + 1

    if @stt < 10
    begin
        set @masv = @ml + '00' + cast (@stt as varchar(1))
    end
    else if @stt < 100
    begin

```

```

        set @masv = @ml + '0' + cast (@stt as varchar(2))
    end
    else
    begin
        set @masv = @ml + cast (@stt as varchar(3))
    end

    insert into sinhvien values (@masv, @ht, @ns, @ml)
Go
--drop proc cauf
exec cauf N'Nguyễn Lâm', '3/30/1990', 'TH01'
-- cau g
create function caug()
returns table
as
    return (select sinhvien.masv, hoten, str(avg(diem),5,2) as
dtb
    from sinhvien, ketqua where sinhvien.masv = ketqua.masv
    group by sinhvien.masv, hoten)
    having avg(diem) >=5
go

select * from caug()

```

4. Bài tập tự làm

Viết các hàm sau:

- Lấy ra họ, tên, họ lót (nếu có)
- Hiển thị họ tên, ngày sinh, thứ (trong tuần, tiếng Việt) mà sinh viên được sinh ra
- Tạo bảng dữ liệu chứa các sinh viên có điểm trung bình < 5. Thông tin gồm: mã sv, họ tên, năm sinh, điểm trung bình (1 số thập phân)
- Hiện thị mã lớp, tên lớp từ mã sinh viên.
- Hiện thị điểm trung bình của một môn học
- Cho biết điểm trung bình cao nhất của sinh viên khi biết mã lớp.
- Tạo mã lớp tự động theo nguyên tắc: mã lớp = “mã khoa” + <số thứ tự> (2 chữ số). Ví dụ: lớp thuộc khoa tin học mới nhất là TH10 thì lớp tiếp theo sẽ là TH11. Hàm nhận vào 1 tham số là mã khoa và trả về mã lớp thuộc khoa đó.

5. Bài tập về nhà

Cho CSDL quản lý NGK với các bảng

- NSX (maNSX, ten NSX)
- NGK (MaNGK, TenNGK, DVT, soluong, dongia, Maloi)
- Hoadon (sohd, ngaylap)
- CTHD (sohd, MaNGK, soluong, dongia)

Yêu cầu: Viết các hàm sau:

- Phát sinh số hoá đơn mới, cho biết: số hoá đơn được tạo dựa vào ngày lập và số thứ tự. Ví dụ: số hoá đơn có mã là: 2303161 nghĩa là hoá đơn 1 ngày 23/03/16. Hoá đơn tiếp theo cùng ngày sẽ là: 2303162
- Thêm một hoá đơn mới, với Số hoá đơn như câu a, ngày lập là ngày hiện hành

-
- c. Thêm 1 NGK mới, Mã NGK tự phát sinh theo nguyên tắc: mã NSX + “số thứ tự” (gồm 3 chữ số). Ví dụ mã loại “COCA”, số thứ tự hiện hành là 10 thì NGK Coca tiếp theo là: COCA011.
 - d. Thêm một CTHD, lưu ý: số lượng phải \leq số lượng trên bảng NGK, đơn giá = đơn giá NGK * 1.5
 - e. Cho biết tổng tiền của một hoá đơn nào đó
 - f. Cho biết các NGK được bán vào tháng 3 năm 2016.
-