



# KHAI BÁO BIẾN TRONG SQL

## Khai báo biến

Cú pháp : **DECLARE** @ten\_bien kieu\_du\_lieu

Ví dụ: **DECLARE** @hoTen **NVARCHAR**(50)

Tên Biến : Luôn bắt đầu bằng ký tự @, không có khoảng trắng, ký tự đặc biệt, ký tự số đứng đầu.

## Gán giá trị

❖ Lệnh gán giá trị cho 1 biến: **SET** @ten\_bien = giá\_trị  
**DECLARE** @hoTen **NVARCHAR**(50)  
**SET** @hoTen = **N**'Hoàng Anh Tú'

❖ Lệnh gán giá trị cho biến từ 1 câu truy vấn

**SELECT** @ten\_bien = tên\_cột **FROM** tên\_bảng

Ví dụ:

**DECLARE** @soLuong **INT**

**SELECT** @soLuong = **COUNT**(\*) **FROM** SINHVIEN





# KHAI BÁO BIẾN TRONG SQL

In giá trị của biến ra màn hình

Sử dụng lệnh PRINT để in giá trị của biến ra màn hình:

Cú pháp: **PRINT** @bien

```
DECLARE @hoTen NVARCHAR(50)
```

```
SET @hoTen = 'Hoàng Anh Tú'
```

```
PRINT @hoTen
```





# CÁC HÀM CƠ BẢN TRONG SQL

**A/ Các hàm chuyển đổi kiểu dữ liệu:** thường dùng để chuyển dữ liệu từ số, ngày sang chuỗi.

**Hàm Cast** : chuyển một kiểu dữ liệu sang kiểu bất kỳ

Cú pháp : ***Cast(Biểu\_Thức as Kiểu\_Dữ\_Liệu)***

Ví dụ : Cast (@tong as varchar(10))

Cast (@float as nvarchar(4))

**Hàm CONVERT** : chuyển từ ngày sang chuỗi

Cú pháp:

***Convert(Kiểu\_dữ\_Liệu,Biểu\_Thức[,Định\_Dạng])***

Ví dụ : Convert(char(10),Getdate(),105)

-> chuỗi ngày 18-08-2014





# CÁC HÀM CƠ BẢN TRONG SQL

**Hàm Str:** chuyển số thành chuỗi

Cú pháp: **STR(số thực, Số ký tự[, số lẻ])**

Ví dụ: **Str(12.345,5,2)->12.35**

*Chú ý:* khi nối chuỗi với số, ngày ta phải chuyển các giá trị này sang chuỗi, sau đó sử dụng dấu + để nối.

**B/ Các hàm ngày giờ**

**Hàm DatePart :** trả về một giá trị trong của ngày

Cú pháp : **DatePart (Định dạng, Ngày) ->số nguyên**

Ví dụ: **Print Datepart(yy, getdate())-> 2014**





# CÁC HÀM CƠ BẢN TRONG SQL

**Hàm DateAdd** : dùng cộng một số vào giá trị ngày và trả về một giá trị ngày

**Cú pháp:** *DateAdd(Định dạng, Số, Ngày)* -> Ngày mới

*Ví dụ*

```
declare @ngaymoi datetime
```

```
set @ngaymoi=dateadd(mm,5,getdate())
```

```
print convert(char(10),@ngaymoi,105) =>22-08-2014
```

**Hàm DateDiff:** trả về khoảng cách của hai ngày

**Cú pháp:** *Datediff(định dạng, ngày\_1, ngày\_2)* -> số nguyên

**Ví dụ:** *print Datediff(mm,getdate(),@ngaymoi) ->5*





# CÁC HÀM CƠ BẢN TRONG SQL

**Hàm DateName:** trả về chuỗi thời gian

Cú pháp : ***Datename(Định dạng, Ngày)***->*chuỗi*

Ví dụ

***print datename(dw,getdate())***->*Friday*

Hàm Getdate() ->*Trả về ngày hiện hành*

Hàm Day(),Month(),Year()-> *trả về ngày, tháng, năm*

C. Một số hàm cơ bản khác: (xem gt)





# CẤU TRÚC ĐK TRONG SQL

## Cấu trúc rẽ nhánh IF...ELSE

Cú pháp:

```
IF biểu_thức_điều_kiện BEGIN
    các_lệnh_xử_lý_khi_thỏa_điều_kiện
END
ELSE BEGIN
    các_lệnh_xử_lý_khi_không_thỏa_điều_kiện
END
```

Ví dụ:

```
DECLARE @soLuong INT
SELECT @soLuong = COUNT(*)
FROM SINHVIEN
IF @soLuong > 0 BEGIN
    PRINT @soLuong
END
ELSE BEGIN
    PRINT N'Không có sinh viên'
END
```





# CẤU TRÚC ĐK TRONG SQL

## ❖ Kết hợp IF với EXISTS

```
IF EXISTS(SELECT * FROM SINHVIEN)
BEGIN
    PRINT N'Có dữ liệu sinh viên'
END
ELSE BEGIN
    PRINT N'Không có dữ liệu sinh viên'
END
```

## ❖ Kết hợp IF với các dạng truy vấn khác

```
IF (SELECT COUNT(*) FROM
SINHVIEN) > 0
BEGIN
    PRINT N'Có dữ liệu sinh viên'
END
ELSE
BEGIN
    PRINT N'Không có dữ liệu sinh viên'
END
```







# CẤU TRÚC ĐK TRONG SQL

## Cấu trúc Case

Case <biểu thức >

When <giá trị 1> then <biểu thức kết quả>

When <giá trị 2> then <biểu thức kết quả>

.....

Else

.....

End

**Ví dụ:** SV(Ho,Ten, Phai)

Select (Ho,Ten, GioiTinh = case phai

When 0 then “Nữ”

When 1 then “Nam”

End)

From sinhvien





# CẤU TRÚC ĐK TRONG SQL

## Cấu trúc lặp WHILE

Cú pháp:

```
WHILE biểu_thức_điều_kiện BEGIN  
    các_lệnh_xử_ly_khi_thỏa_điều_kiện  
END
```

Ví dụ : In ra dãy số từ 1 đến 99

```
DECLARE @i INT  
SET @i = 1  
  
WHILE @i < 100 BEGIN  
    PRINT @i  
  
    SET @i = @i + 1  
  
END
```





# CON TRỎ (CURSOR) TRONG SQL

## Cú pháp của Cursor luôn tuân theo 1 số bước

### Khai báo Cursor

Cách 1:

```
DECLARE tên_cursor CURSOR  
FOR lệnh_select
```

Ví dụ: 

```
DECLARE cursor_lop CURSOR  
FOR SELECT malop FROM LOP
```

Cách 2:

```
DECLARE @tên_cursor CURSOR  
SET @tên_cursor = CURSOR FOR lệnh_select
```

Ví dụ: 

```
DECLARE @cursor CURSOR  
SET @cursor = CURSOR FOR SELECT Masv, Tensv FROM SINHVIEN
```





# CON TRỎ (CURSOR) TRONG SQL

Cú pháp của Cursor luôn tuân theo 1 số bước

## 2. Mở Cursor

**OPEN** tên\_cursor    **Ví dụ:**    OPEN cursor\_lop  
OPEN @cursor

## 3. Lấy dữ liệu từ Cursor

**FETCH NEXT FROM** tên\_cursor **INTO** @biến1, @biến2,...

Kiểm tra có lấy được dữ liệu ra không bằng biến @@FETCH\_STATUS:

- Nếu @@FETCH\_STATUS = 0: lấy dữ liệu thành công.
- Nếu @@FETCH\_STATUS <> 0: lấy dữ liệu không thành công.

## 4. Đóng Cursor

**CLOSE** tên\_cursor  
**DEALLOCATE** tên\_cursor

**Lưu ý :** Tại mỗi thời điểm không thể mở 2 cursor có cùng 1 tên





# CHƯƠNG TRÌNH CON TRONG SQL

## A. Thủ tục ( Store Procedure)

### 1. Cấu trúc thủ tục trong lập trình:

#### Cú pháp:

Create Procedure <Tên thủ tục> (@biến <kiểu DL>,...)

As

Các lệnh xử lý

#### Cách gọi thủ tục :

C1.<Tên thủ tục> giá trị 1,gia trị 2

C2.Exec <Tên thủ tục> giá trị 1,gia trị 2





# CHƯƠNG TRÌNH CON TRONG SQL

## A. Thủ tục ( Store Procedure)

**Ví dụ:** Tính tổng số chẵn từ 1 -> n

```
Create Proc Tong (@n int)
```

```
As
```

```
Declare @t int, @x int
```

```
Set @t = 0 ; Set @x = 1
```

```
While (@x <= @n)
```

```
begin
```

```
if ((@x % 2) = 0)
```

```
set @t = @t + @x
```

```
set @x = @x + 1 end
```

```
Print @t
```

**Exec** Tong 4 => 6





# CHƯƠNG TRÌNH CON TRONG SQL

## A. Thủ tục ( Store Procedure)

### 2. Cấu trúc thủ tục trong xử lý dữ liệu:

#### a. Thêm dữ liệu vào bảng:

#### Cú pháp:

Create Procedure <Tên thủ tục> (@biến <kiểu DL>,...)

As

Begin

If not exists (select tenKC from Ten\_Bang where tenKC=@bienKC)

Begin

Insert into Ten\_Bang(Ten\_cot1[,...]) values( @Ten\_cot1[,...])

Print 'Đã chèn thành công'

End

Else

Print 'Đã có dữ liệu'

End





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ:** Cho table Lop (MaLop, TenLop, SiSo)

Hãy VTT thêm vào 1 lớp mới

```
Create Proc Them (@MaLop nchar(10), @TenLop nchar(10), @SiSo int)
```

```
As
```

```
Begin
```

```
If not exists ( select MaLop from Lop where MaLop=@MaLop)
```

```
Begin
```

```
Insert into Lop (MaLop, TenLop,SiSo) values (@MaLop,@TenLop,@SiSo)
```

```
Print 'da chen thanh cong'
```

```
End
```

```
Else
```

```
Print ' da co du lieu'
```

```
End
```

```
Exec Them '1','CNTT2014',60
```







# CHƯƠNG TRÌNH CON TRONG SQL

## b. Cập nhật dữ liệu vào bảng:

### Cú pháp:

Create Procedure <Tên thủ tục> (@biến <kiểu DL>,...)

As

Begin

If exists (select tenTSV from Ten\_Bang where tenTSV=@bienTSV)

Begin

Update Ten\_Bang

Set Ten\_Cot\_Cu = @Ten\_Cot\_Moi

Where tenTSV = @bienTSV

Print 'Đã cập nhật thành công'

End

Else

Print 'Cập nhật không thành công'

End





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ:** Cho table Lop (MaLop, TenLop, SiSo)  
Hãy VTT cập nhật sĩ số cũ thành sĩ số mới với **TSV** là **MaLop**.

```
Create Proc CapNhat (@MaLop nchar(10), @SiSoMoi int)
As
Begin
If exists ( select MaLop from Lop where MaLop=@MaLop)
Begin
    Update Lop
    Set SiSo=@SiSoMoi
    Where MaLop=@MaLop
    Print 'Cap nhat thanh cong'
End
Else
Print ' Cap nhat khong thanh cong'
End
Exec CapNhat '1',70
```





# CHƯƠNG TRÌNH CON TRONG SQL

## c. Xóa dữ liệu trong bảng:

### Cú pháp:

Create Procedure <Tên thủ tục> (@biến <kiểu DL>,...)

As

Begin

If exists (select tenTSV from Ten\_Bang where tenTSV=@bienTSV)

Begin

Delete from Ten\_Bang

Where tenTSV = @bienTSV

Print 'Đã xóa thành công'

End

Else

Print 'Xóa không thành công'

End





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ:** Cho table Lop (MaLop, TenLop, SiSo)

Hãy VTT xóa 1 lớp với **TSV là MaLop**.

```
Create Proc Xoa (@MaLop nchar(10))
```

```
As
```

```
Begin
```

```
If exists ( select MaLop from Lop where MaLop=@MaLop)
```

```
Begin
```

```
    Delete from Lop
```

```
    Where MaLop=@MaLop
```

```
    Print 'Da xoa thanh cong'
```

```
End
```

```
Else
```

```
Print ' Xoa khong thanh cong'
```

```
End
```

```
Exec Xoa '1'
```





# CHƯƠNG TRÌNH CON TRONG SQL

## d. Truy vấn dữ liệu trong bảng:

### Cú pháp:

Create Procedure <Tên thủ tục> (@biến <kiểu DL>,...)

As

Begin

Câu lệnh SQL

End

**Ví dụ:** Cho các table sau:

SinhVien(Masv, Tensv, MaLop)

Lop (MaLop, TenLop)

Hãy VTT cho biết tổng số sinh viên của lớp với TSV là MaLop.





# CHƯƠNG TRÌNH CON TRONG SQL

```
Create Proc Sosv (@MaLop nchar(10))
As
Declare @dem int
Begin
    Select @dem = ( Select count(Masv)
                    From SinhVien, Lop
                    Where SinhVien.MaLop=Lop.MaLop and
                        Lop.MaLop=@MaLop)
    Print 'So sinh vien cua lop co ma' + @MaLop + 'la'+cast(@dem as char(10))
End
Exec Sosv '1'
```





# CHƯƠNG TRÌNH CON TRONG SQL

## 3. Sửa đổi thủ tục:

Để sửa đổi các thông tin trong thủ tục ta thay từ khóa Create bằng từ khóa Alter và các thay đổi trong thủ tục đó.

## 4. Xóa thủ tục:

### Cú pháp:

```
Drop PROC Ten_Thu_Tuc
```





# CHƯƠNG TRÌNH CON TRONG SQL

## B. Hàm (Function):

### 1. Các loại hàm :

- **Hàm xác định (deter-ministic):** luôn trả về 1 giá trị khi nhận các giá trị truyền vào như nhau.
- **Hàm không xác định (non-deterministic):** cho giá trị khác nhau tùy thời gọi như hàm Getdate(),...







# CHƯƠNG TRÌNH CON TRONG SQL

## 2. Xây dựng và cách gọi hàm :

**A/ Hàm trả về giá trị:** giá trị trả về có kiểu dữ liệu là một trong các kiểu của SQL Server.

### Cú pháp:

Create Function Ten\_Ham[(Các tham số)]

Returns Kiểu\_dữ\_liệu\_trả\_về

As

Begin

<các xử lý>

Return ket\_qua

End

### Cách gọi Hàm:

Select dbo.Ten\_Ham[(Các tham số)]





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ:** Viết hàm tìm số lớn nhất của 2 số a, b.

```
Create Function sln(@a int,@b int)
```

```
Returns int
```

```
As
```

```
Begin
```

```
Declare @max int
```

```
Set @max=@a
```

```
If @b>@max
```

```
SET @max=@b
```

```
Return @max
```

```
End
```

Thực thi: `select dbo.sln(2,4) as số_lớn_nhất_là`





# CHƯƠNG TRÌNH CON TRONG SQL

**B/ Hàm trả về dữ liệu được lấy từ các bảng trong CSDL:**  
giống như view nhưng có tham số vào.

**Cú pháp :**

Create Function Tên\_Hàm[(Tham số vào)]

Returns Table

As

Return (Câu lệnh Select)

**Cách gọi Hàm:**

Select \* from dbo.Tên\_Hàm[(Tham số vào)]





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ 1:** Cho CSDL sau:

SV (Masv, Hoten, Dchi, Makhoa)

Khoa (Makhoa, Tenkhoa)

Viết hàm in ra thông tin sinh viên gồm: Masv, Hoten, Dchi, Tenkhoa **với tham số vào là Makhoa**

Create Function DSSV (@Makhoa nchar(10))

Returns table

As

Return (select Masv, Hoten, DChi, Tenkhoa

from SV, Khoa

where SV.MaKhoa=Khoa.Makhoa and

Khoa.Makhoa=@Makhoa)

**Thực thi**

**Select \* from dbo.DSSV('1')**





# CHƯƠNG TRÌNH CON TRONG SQL

**Ví dụ 2:** Cho CSDL sau:

SV (Masv, Hoten, Dchi, Makhoa)

Khoa (Makhoa, Tenkhoa)

Viết hàm cho biết tổng số sinh viên của Khoa với tham số vào là **Tenkhoa**.

Create Function SSV (@Tenkhoa nchar(10))

Returns table

As

Return (select khoa.makhoa, count(Masv) as Số\_SV from SV, Khoa  
where SV.MaKhoa=Khoa.Makhoa and  
Khoa.Tenkhoa=@Tenkhoa  
group by khoa. makhoa)

Thực thi

Select \* from dbo.SSV('CNTT')





# CHƯƠNG TRÌNH CON TRONG SQL

## 3/ Sửa đổi Hàm:

Để sửa đổi các thông tin trong Hàm ta thay từ khóa Create bằng từ khóa Alter và các thay đổi trong Hàm đó.

## 4/ Xóa Hàm:

### Cú pháp:

Drop FUNCTION Ten\_Ham





# TRIGGER TRONG SQL

## 1. Tổng quan:

Trigger là một dạng đặc biệt của thủ tục nội tại nhưng :

- Trigger không có tham số
- Không gọi thực hiện bằng lệnh Exec, mà tự động kích hoạt khi dữ liệu trên bảng có liên quan đến Trigger được cập nhật
- Một Trigger được tạo cho một bảng và dùng để kiểm tra các ràng buộc toàn vẹn phức tạp hoặc cập nhật dữ liệu của các bảng liên quan.





# TRIGGER TRONG SQL

## 2. Xây dựng Trigger:

### Cú pháp

Create Trigger Ten\_trigger on Ten\_Bang

For Insert[,Update,Delete]

AS

<Tập lệnh T-SQL>







# TRIGGER TRONG SQL

**Ví dụ 1:** Cho table NV (Manv, Tennv, Luong)

Hãy tạo trigger tăng 30% lương cho các nhân viên

Create TRIGGER tangluong

ON NV FOR INSERT, UPDATE

AS

BEGIN

UPDATE NV

SET Luong=Luong+Luong\*30/100

END

**Thực thi:** Update NV

Set Luong=Luong+Luong\*30/100





# TRIGGER TRONG SQL

Thực thi trigger:

B1: Hủy bỏ ràng buộc trên các table.

B2: Gọi lệnh xóa, thêm hoặc cập nhật

B3: Tạo lại các ràng buộc trên các table.

Ứng dụng thực thi cho ví dụ 4:

B1:     Alter table DiaDiem nocheck constraint all  
          Alter table NhanVien nocheck constraint all

B2:     Delete DiaDiem  
          Where MaDD=1

B3:     Alter table DiaDiem check constraint all  
          Alter table NhanVien check constraint all

Cuối cùng sử dụng lệnh Select để kiểm tra.





# TRIGGER TRONG SQL

## Hai bảng tạm Inserted và Deleted

- *Khi thêm dữ liệu vào bảng, dữ liệu được thêm vào sẽ đưa vào bảng tạm Inserted*
- *Khi xóa dữ liệu ra bảng, dữ liệu xóa sẽ đưa vào bảng tạm Deleted*
- *Thao tác cập nhật được xử lý :*
  - ✓ *Xóa dòng dữ liệu cũ (dữ liệu cũ sẽ đưa vào bảng Deleted)*
  - ✓ *Thêm dòng dữ liệu mới (dữ liệu mới sẽ đưa vào bảng Inserted)*
- *Để lấy dữ liệu vừa mới cập nhật vào bảng ta truy cập vào bảng tạm Deleted hoặc Inserted, chỉ truy cập được hai bảng này trong Trigger*





# TRIGGER TRONG SQL

**Ví dụ 2:** Cho database

DiaDiem(MaDD,TenDD)

NhanVien(MaNv,HoTen,MaDD,MaBP)

BoPhan(MaBP,TenBP,MaDD)

Thêm nhân viên kiểm tra địa điểm

Create trigger them on NhanVien for insert

As

If not Exists(select DiaDiem.MaDD)

From inserted,diadiem

Where inserted.madd=diadiem.madd)

Begin

Print 'Đã thêm thành công'

End





# TRIGGER TRONG SQL

**Ví dụ 3:** Cho database

DiaDiem(MaDD,TenDD)

NhanVien(MaNv,HoTen,MaDD,MaBP)

BoPhan(MaBP,TenBP,MaDD)

Cập nhật cha thì cập nhật con

Create trigger capnhat on DiaDiem for update

As

If update(MaDD)

Update NhanVien

Set Nhanvien.MaDD=inserted.MaDD

From inserted,deleted

Where deleted.MaDD=NhanVien.MaDD





# TRIGGER TRONG SQL

**Ví dụ 4:** Cho database

DiaDiem(MaDD,TenDD)

NhanVien(MaNv,HoTen,MaDD,MaBP)

BoPhan(MaBP,TenBP,MaDD)

Tạo trigger khi xóa địa điểm trên table địa điểm thì table nhân viên sẽ xóa theo

Create trigger xoa on DiaDiem for delete

As

Delete Nhanvien

From nhanvien,deleted

Where Nhanvien.MaDD=deleted.MaDD

