

CHƯƠNG 5

STORE PROCEDURE VÀ TRIGGER



Bài 10: Cơ bản về lập trình bằng T_SQL

- Khai báo và sử dụng biến
- Các cấu trúc lệnh
- RaiseError



IDENTIFIERS_ĐỊNH DANH

- Tên của các đối tượng đều được gọi là định danh. Trong SQL Server, có các định danh như Server, Databases, object of Database as Table, View, Index, Constraint,...
- Quy tắc định danh
 - Tối đa 128 ký tự.
 - Bắt đầu là một ký tự từ A_Z
 - Bắt đầu là một ký hiệu @, # sẽ có một ý nghĩa khác.
 - Những định danh nào có dấu khoảng trắng ở giữa thì phải kẹp trong dấu [] hoặc “ “
 - Đặt định danh sao cho ngắn gọn, đầy đủ ý nghĩa, phân biệt giữa các đối tượng với nhau, không trùng lặp, không trùng với từ khóa của T-SQL.



Tham chiếu đến các đối tượng trong SQL Server

Cú pháp:

`Server.database.Owner.Object`

Or

`Server.database..Object`

Ví dụ

`Create Table Northwind.dbo.Customers`

`Create Table Northwind..Customers`



Data type _ Batch _ Script

1. Kiểu dữ liệu (Datatype) : có hai loại

- Kiểu dữ liệu hệ thống: Do hệ thống cung cấp
- Kiểu dữ liệu do người dùng định nghĩa (User – defined datatypes.)

2. Gói lệnh (Batch)

- Bao gồm các phát biểu T-SQL và kết thúc bằng lệnh GO.
- Các lệnh trong gói lệnh sẽ được biên dịch và thực thi cùng một lúc.
- Nếu một lệnh trong Batch bị lỗi thì batch cũng xem như lỗi
- Các phát biểu Create bị ràng buộc trong một batch đơn.

Ex : use northwind
 select * from Customers

 GO

3. Kịch bản (Script)

- Một kịch bản là một tập của một hay nhiều bó lệnh được lưu lại thành một tập tin .SQL



Biến – Biến cục bộ

- Biến là một đối tượng dùng để lưu trữ dữ liệu. Biến phải được khai báo trước khi dùng.
- Có 2 loại biến: cục bộ và toàn cục
- Biến cục bộ:
 - Được khai báo trong phần thân của một bó lệnh hay một thủ tục.
 - Phạm vi hoạt động của biến bắt đầu từ điểm mà nó được khai báo cho đến khi kết thúc một bó lệnh, một thủ tục hay một hàm mà nó được khai báo.
 - Tên của biến bắt đầu bằng @





Sử dụng biến cục bộ

- Khai báo

```
DECLARE @var_name = expression  
SELECT { @var_name = expression }[,...n]
```

Ví dụ

```
DECLARE @makh CHAR(4)  
SET @makh = 'ANTON'  
SELECT * FROM Customers  
WHERE Customerid = @makh
```



Sử dụng biến cục bộ

Example :

```
DECLARE @manv int
SET @manv = 2
Go
SELECT * FROM Employees
        WHERE Employeeid = @manv
```

```
DECLARE @manv int, @country nvarchar(15)
SET @manv = 3
Set @country = 'Usa'
SELECT * FROM Employees
        WHERE Employeeid = @manv and country = @country
```




Sử dụng biến cục bộ

Example :

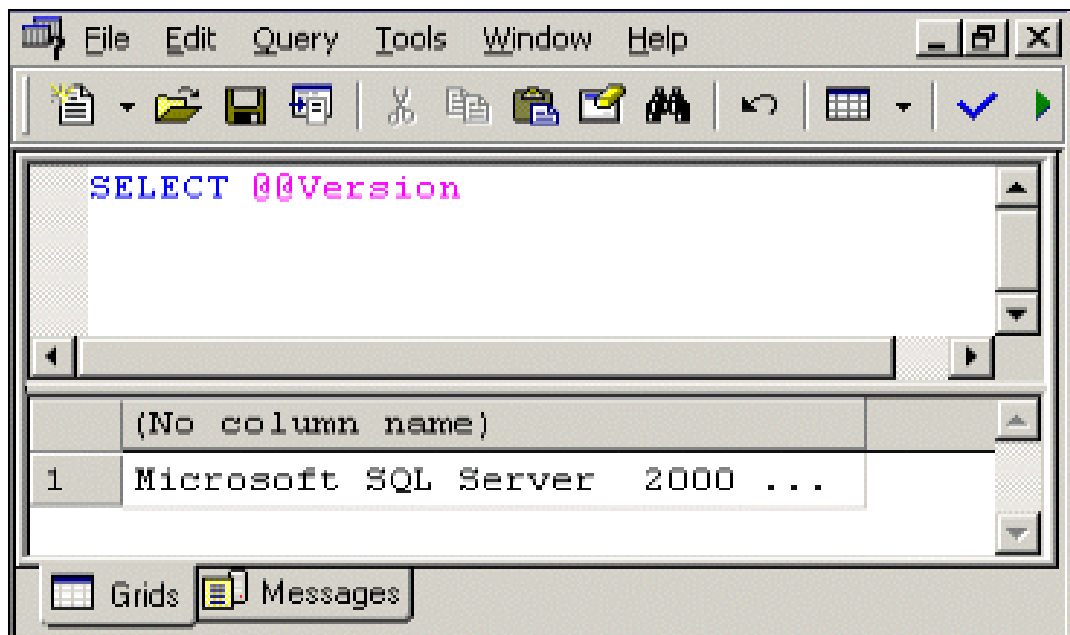
```
DECLARE @tong int
Select @tong = Sum(quantity * Unitprice) From [Order details]
SELECT @tong as tongtien
Print 'Tong tien =' +convert(varchar(20),@tong)
```

```
DECLARE @masp int
Select @masp = productid from Nortwind..Products
Select @masp
```

```
DECLARE @masp int
Select @masp = productid from Nortwind..Products
Order by productid desc
Select @masp
```

Biến toàn cục

- Biến toàn cục được định nghĩa như hàm hệ thống. Các biến này không có kiểu. Tên bắt đầu bằng @@





Các biến toàn cục

Variable	Return value
@@Trancount	Number of transactions currently open on the connection
@@Servername	Name of local servers running SQL Server
@@Rowcount	Number of rows affected by the latest SQL statement
@@Identity	Return last Number Identity
@@Error	Return order number Error when SQL exculcate, return 0 when The command completed successfully
@@Fetch_status	Return status of Fetch command of pointer variable (0 :Success, -1 : Mistake or exceed range, -2 : Unsuccess



Các biến toàn cục

Example

How many are transaction opening

```
If (@@Tranccount>0)
```

```
Begin
```

```
    Raiserror ('Take can not be executed within a trasaction',10,1)
```

```
    Return
```

```
End
```

Number of rows affected by the latest SQL statement

1) Use Northwind

```
Update Employees Set LastName = 'Brooke' where LastName = 'Lan'
```

```
If (@@RowCount =0)
```

```
Begin
```

```
    Print 'Warning : No rows were update'
```

```
    Return
```

```
End
```

2) Update Customers Set Phone = '030' +Phone

```
Where Country = 'Germany'
```

```
Print @@Rowcount
```



Các biến toàn cục

Example

Tra ve so Identitidey phat sinh sau cung

```
Create table hd (mahd int identity Primary key, ghichu varchar(20))
```

```
Create table cthd(Mahd int, masp char(10), soluong int)
```

```
insert into hd Values ('Record 1')
```

```
insert into hd Values ('Record 2')
```

```
Declare @maso int
```

```
Set @maso = @@identity
```

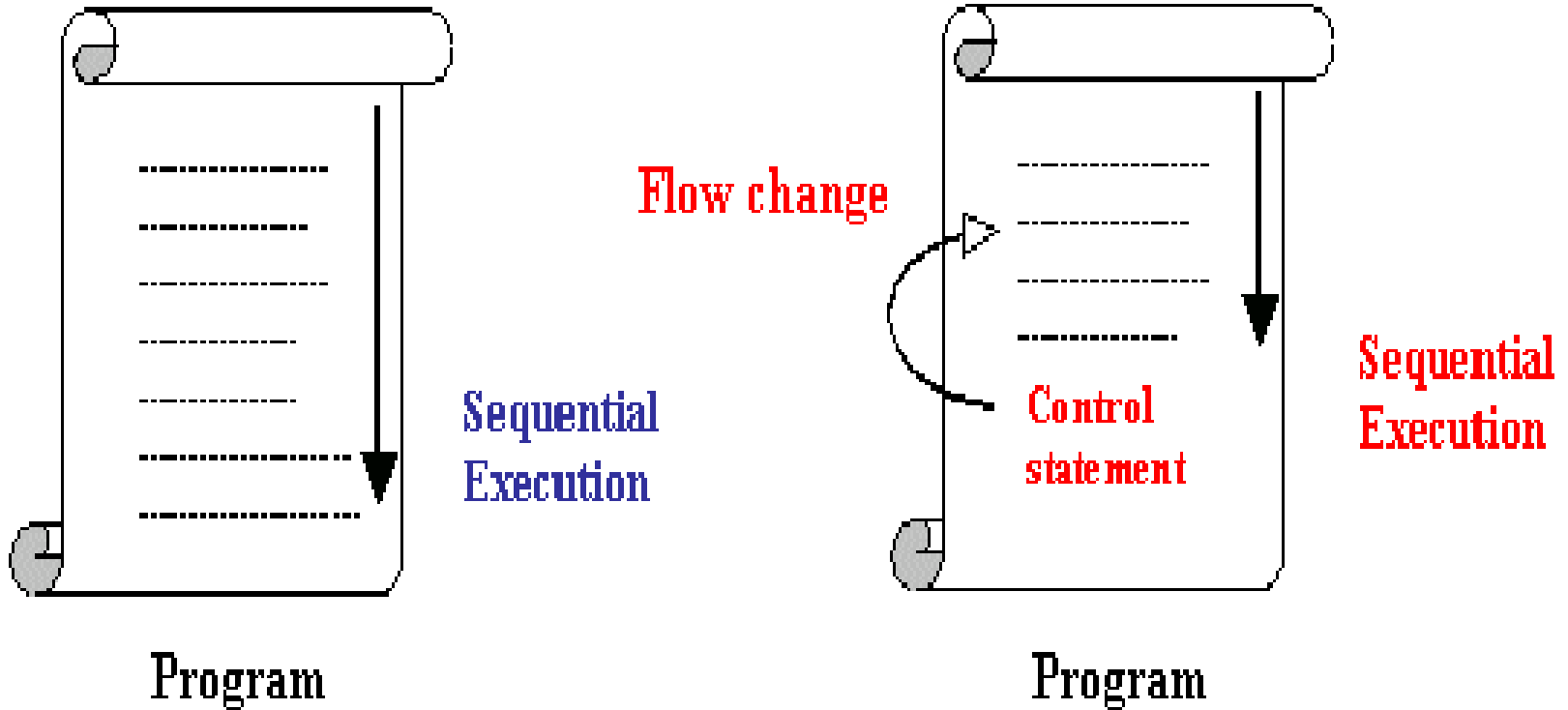
```
insert into cthd Values (@maso,'sp001',5)
```

```
insert into cthd Values (@maso,'sp002',12)
```

```
Select * from hd
```

```
Select * from cthd
```

Cấu trúc điều khiển



Cấu trúc điều khiển

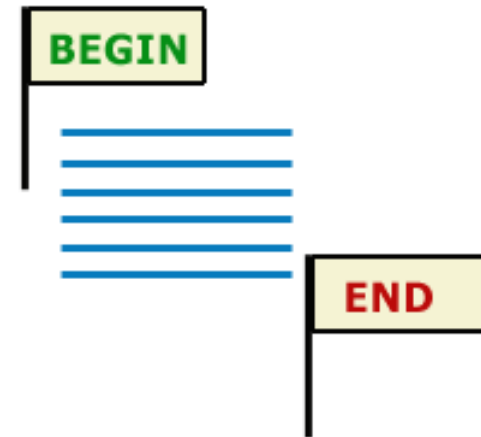
- Khối **BEGIN...END**: Nếu nhiều phát biểu cần thực thi với nhau thì đặt các phát biểu này trong Begin...End

Cú pháp:

BEGIN

statement | statement_block

END



- **RETURN**: Trả về một giá trị, lệnh này nằm trong một block hay procedure. Nếu gặp phát biểu Return, quá trình xử lý kết thúc

Cú pháp

Return [Integer_expression]



Cấu trúc điều khiển

- **Lệnh PRINTER:** Dùng để in thông tin ra màn hình kết quả của SQL.

Cú pháp:

PRINT 'any ASCII text'|@local_variable|@@Function |string_expr

Ví dụ:

Print 'Hello'

Print n'Chao ban'

Print getdate()

Print @@version

Set @ten ='Nguyen Minh'

Print @ten

Cấu trúc điều khiển

- Cấu trúc điều khiển IF...ELSE: Cho phép thực thi một hay nhiều lệnh tùy thuộc vào một điều kiện nào đó.

- Cú pháp:

If Condition

statements

[Else [Condition 1]

statements]

- Ví dụ :

```
If (select Count(*) From Customers where Country ='Germany')>0
```

```
    print 'Co khách hàng ở Germany'
```

```
Else
```

```
    print 'Khong có khách hàng ở Germany'
```





Cấu trúc điều khiển

Ví dụ 2:

```
Declare @msg varchar(100)
```

```
If (select Count(unitprice) From Products where QuantityPerunit like  
    '%box%' )>0
```

```
Begin
```

```
Set NOCOUNT on
```

```
Set @msg = 'Co vai sp có don vi tinh co chu box. Cac sp do la'
```

```
select @msg
```

```
select ProductName From Products
```

```
    where QuantityPerunit like '%box% '
```

```
end
```

```
Else
```

```
print 'Khong co sp nao co dvt co chu box'
```



Cấu trúc điều khiển

Example 3:

```
use Pubs
```

```
If (select avg(unitprice) From Products where QuantityPerunit like  
    '%box%' )>0
```

```
Begin
```

```
    Set NOCOUNT on
```

```
    Set @msg = 'Co vai sp có don vi tinh co chu box. Cac sp do la'
```

```
    select @msg
```

```
    select ProductName From Products
```

```
        where QuantityPerunit like '%box% '
```

```
end
```

```
Else
```

```
    print 'Khong co sp nao co dvt co chu box'
```

Cấu trúc điều khiển

■ **CASE** : là một biểu thức điều kiện được áp dụng bên trong một phát biểu khác. Case trả về các giá trị khác nhau tùy vào điều kiện hay một khiên nào đó.

- Cú pháp 1 :

Case input_expression

When when_expression Then result_expression[...n]

[

ELSE else_result_expression

]

End

- Cú pháp 2 :

Case Boolean_expression

When Boolean_expression Then result_expression[...n]

[

ELSE else_result_expression

]

End



Cấu trúc điều khiển

- Example 1 :

```
Declare @a int, @b int, @Hieu int
```

```
Set @a = 15
```

```
Set @b =27
```

```
Set @hieu = Case
```

```
    When @a<@b then @b-@a
```

```
    When @a>@b then @a-@b
```

```
    else 0
```

```
end
```

```
print 'hieu='+convert(varchar(20),@hieu)
```

- Example 2 :

```
Select ProductName, Unitprice,
```

```
'Classification'=CASE
```

```
    when Unitprice<10 then 'Low price'
```

```
    When Unitprice Between 10 and 20 then 'Moderately Price'
```

```
    when Unitprice>20 then 'Expensive'
```

```
    else 'Unknown'
```

```
end
```

```
From Products
```



Cấu trúc điều khiển

```
Select productid, Quantity, UnitPrice, [discount%]=
CASE
    When Quantity <=5 then 0.05
    When Quantity between 6 and 10 then 0.07
    When Quantity between 11 and 20 then 0.09
Else 0.1
end
From [Order Details]
Order by Quantity, Productid
```



Cấu trúc điều khiển

- GOTO: redirects the flow of program execution to a specified location (label)

- Example

Declare @a int, @b int, @Hieu int

Set @a = 39

Set @b = 10

hieu_loop:

if @a>@b

begin

Set @hieu = @A - @B

print 'a='+convert(varchar(20),@a)

print 'b='+convert(varchar(20),@b)

print 'hieu='+convert(varchar(20),@hieu)

Set @a = @hieu

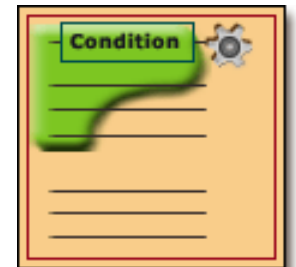
goto hieu_loop

print 'a='+convert(varchar(20),@a)

print 'b='+convert(varchar(20),@b)

print 'hieu='+convert(varchar(20),@hieu)

end



Cấu trúc điều khiển

- **Phát biểu lặp WHILE:** Vòng lặp sẽ thực thi cho đến khi biểu thức điều kiện (Boolean expression) trong While mang giá trị False.
- Syntax 1 :

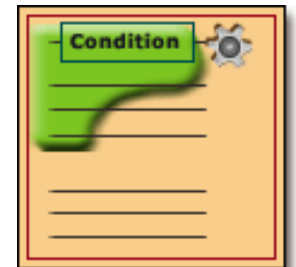
WHILE Boolean_expression

{sql_statement | statement_block}

[BREAK]

{sql_statement | statement_block}

[CONTINUE]



Cấu trúc điều khiển

- Example :

use northwind

While (Select avg(unitprice) from [Order Details]) <\$50

Begin

 Update [order Details]

 SET Unitprice = Unitprice *2

 Select Max(Unitprice) From [Order Details]

 If (Select Max(Unitprice) From [Order Details])>\$50

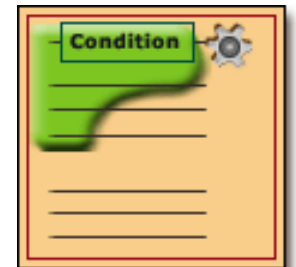
 BREAK

 Else

 CONTINUE

end

Print 'Too much for the market to bear'



Cấu trúc điều khiển

- WAITFOR: SQL Server tạm dừng một thời gian trước khi xử lý tiếp các phát biểu sau đó.

- Cú pháp :

WAITFOR {DELAY 'time' | TIME 'time'}

Time : hh:mm:ss

Delay 'time': hệ thống tạm dừng trong khoảng thời gian time

TIME 'time': hệ thống tạm dừng trong khoảng thời gian time chỉ ra



Ví dụ

```
WAITFOR DELAY '00:00:02'
```

```
SELECT EmployeeID FROM Northwind.dbo.Employees
```



Cấu trúc điều khiển

- Lệnh **RAISERROR**: phát sinh lỗi của người dùng

Cú pháp

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

- **Msg_id**: Là thông báo, nó được lưu trong bảng sysmessage. Mã thông báo của người dùng phải bắt đầu từ trên 50000
- **Msg_str**: Nội dung thông báo, tối đa 400 ký tự.
- Để truyền tham số vào trong thông báo thì dùng dạng %<Loại ký tự>
- Loại ký tự là d,I,o,x,X hay u



Cấu trúc điều khiển

- Thêm một lỗi mới của người dùng định nghĩa

Syntax

```
Sp_AddMessage msg_id,  
severity,'msg'[, 'language'][, 'with_log'][, 'replace']
```

- Xóa một lỗi mới của người dùng định nghĩa

Syntax

```
Sp_DropMessage msg_id
```



Cấu trúc điều khiển

Ví dụ

- SP_ADDMESSAGE 50001,10,'KHONG TIM THAY MAU TIN %D TRONG %LS'
- SP_ADDMESSAGE 50002,16,'KHONG XOA DUOC %S VI %S CO TON TAI TRONG %LS'
- SP_ADDMESSAGE 50003,16,'MOT LOP CHI CO TOI DA %D HOC SINH'
- SP_ADDMESSAGE 50004,16,'DON GIA BAN PHAI LON HON DON GIA GOC'
- --XEM THONG BAO LOI VUA XAY DUNG(COI LAI SAI)
- SP_HELPTEXT 'SYSMESSAGE'
- SELECT * FROM SYSMESSAGE WHERE ERROR =50002



RAISERROR statement

- --CAU 5 :XAY DUNG CAU THONG BAO LOI BANG RAISERROR
- RAISERROR (50001,10,1,4,'SANPHAM')
- DECLARE @@MA INT
- DECLARE @@TEN NVARCHAR
- SET @@TEN ='SANPHAM'
- SET @@MA =8
- SELECT productid FROM products WHERE productid=@ @MA
- IF (@ @ROWCOUNT=0)
- BEGIN
- RAISERROR (50001,10,1,@ @MA,@ @TEN)
- END
- GO