



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

# مقدماتی بر استفاده پیشرفته تر از SQL Server

آزمایشگاه پایگاه داده  
حمیدرضا رمضانی

[h.ramezany72@gmail.com](mailto:h.ramezany72@gmail.com)

# دستور Case

مقدار خاصی را بر اساس شرایط تعریف شده ، در دستور Select ، باز می گرداند.  
قالب کلی این دستور به اشکال زیر است :  
حالت اول :

Case	عبارت			
	When	مقدار اولیه	Then	مقدار بازگشتی
	...			
	[Else]			
End	[As	نام]		

حالت دوم :

Case				
	When	عبارت شرطی	Then	مقدار بازگشتی
	...			
	[Else]			
End	[As	نام]		

# مثال از دستور CASE

- `select boatcolor,bname=(case boatname`
- `when 'R1' then 'AAAA'`
- `when 'R2' then 'BBBB'`
- `when 'R3' then 'CCCC'`
- `when 'R5' then 'DDDD'`
- `else 'others'`
- `end)`
- `from session3_DB.dbo.tblBoat;`

	boatcolor	bname
1	red	AAAA
2	blue	BBBB
3	Green	CCCC
4	gray	DDDD

boatname	boatcolor	boatrank
R1	red	6
R2	blue	10
R3	Green	12
R5	gray	9

# WAITFOR

```
▪ WAITFOR {DELAY 'time' | TIME 'time'}  
  
--Pause for ten seconds  
WAITFOR delay '000:00:10';  
  
PRINT 'Done';  
  
--Pause until a certain time  
WAITFOR time '12:00:00';  
  
PRINT 'It is noon';
```

# کار با متغیر ها

تعریف متغیر : یکی از اشیا در SQL Server است که برای حمل و نگهداری مقادیر مفرد (یکتا Single) به کار می روند  
متغیر ها در T-SQL به منظور های زیر به کار می روند:

۱- به عنوان یک شمارنده که معمولاً تعداد دفعات اجرای یک حلقه را مشخص می کنند .

۲- برای حمل داده های استفاده شده در یک "کنترل جریان" (Control-of-Flow).

۳- ذخیره داده ها به منظور بازگرداندن توسط یک Function و یا Stored Procedure

از علامت @ به منظور معرفی یک متغیر استفاده می شود .

برای تعریف یک متغیر از قالب زیر استفاده می شود :

**Declare    @ *VariableName* [data type]**

برای مقدار دهی به متغیر ها از یکی از قالب های زیر استفاده می شود :

**Select        @ *VariableName* =    *Value***

یا

**Set            @ *VariableName* =    *Value***



# کار با متغیر ها

مثال :

```
Declare @Number      Int
Set      @Number      =      10
Select   @Number      =      10
```

مثال :

```
Use [Lab-EasyShop]
Declare @Title      nVarChar(100),
                @Titles      nVarChar(Max)

Set      @Titles      =      N"
Select   @Title      =      Title
        From      Category
        Where ID = 1
Select   @Titles      =      Title + N',' + @Titles
        From      Category
```

مثال :

```
DECLARE @DBNAME VARCHAR(50)  
SET @DBNAME = 'TEST'
```

```
CREATE DATABASE @DBNAME
```

# دستور *If ... Else*

برای تعریف شرط بکار می رود .

قالب کلی دستور به شکل زیر است :

**If** عبارت شرطی

کد دستورات مورد نظر

**[ELSE]**

[کد دستورات مورد نظر]

مثال :

```
Declare    @Number    Int
Set        @Number    = 15
```

```
if @Number % 2 = 0
    Print 'Even'
Else
    Print 'Odd'
```



# دستور *IF ... Else*

مثال :

مطلوبست جابجایی دو مقدار در متغیر ، چنانچه متغیر اولی از دومی  
بزرگتر بود بدون استفاده از هیچ متغیر واسط .

حمیدرضا رمضانی  
آزمایشگاه پایگاه داده



# دستور *IF ... Else*

مطلوبست جابجایی دو مقدار در متغیر ، چنانچه متغیر اولی از دومی بزرگتر بود بدون استفاده از هیچ متغیر واسط .

```
Declare    @FirstNumber  Int ,
           @SecondNumber Int

Select    @FirstNumber = 100 ,
          @SecondNumber = 1

If        @FirstNumber > @SecondNumber
Begin
    Set    @FirstNumber    = @FirstNumber + @SecondNumber
    Set    @SecondNumber   = @FirstNumber - @SecondNumber
    Set    @FirstNumber    = @FirstNumber - @SecondNumber
End

Else
Begin
    Declare @TempNumber    Int
    Set    @TempNumber     = @FirstNumber
    Set    @FirstNumber    = @SecondNumber
    Set    @SecondNumber   = @TempNumber
End

Select    @FirstNumber    As FirstNumber,
          @SecondNumber   As SecondNumber
```



# دستور While

برای تعریف حلقه بکار می رود.  
قالب کلی دستور به شکل زیر است :

عبارت شرطی حضور در حلقه While

[Begin]

کد دستورات مورد نظر

[End]

مثال :

```
Declare    @Number    Int
Set        @Number    = 100
```

```
While      @Number > 0
Begin
  Print    @Number
  Set      @Number    = @Number    - 1
End
```

# دستور While

مثال :

مطلوبست اعداد زوج بین دو عدد

```
Declare @FirstNumber Int ,
        @SecondNumber Int
Select
    @FirstNumber = 1 ,
    @SecondNumber = 100
Set @FirstNumber = @FirstNumber + @FirstNumber % 2

While @FirstNumber <= @SecondNumber
Begin
    Print @FirstNumber
    Set @FirstNumber = @FirstNumber + 2
End
```

# Sub Query

Sub Query حاوی یک دستور Select است که فقط یک مقدار را بر می گرداند به عبارت دیگر Result Set حاصل از یک Sub Query جدولی است تنها دارای یک سطر و یک ستون .  
از نتیجه Sub Query می توان در دستورات Select و یا Where استفاده نمود .  
مثال :

Use [Lab-Inventory]

Select

```
(Select Title From Item
Where Item.ID = Inventory.Item_ID) As Item ,
(Select Title From Color
Where Color.ID = Inventory.Item_ID) As Color ,
Quantity
```

From

Inventory

# Nested Query

شامل یک یا چند دستور Select است که در دستور From قرار می گیرد  
هنگامیکه بخواهیم مقادیری خاص از یک Result Set به عنوان محدوده بازبینی اطلاعات قرار گیرند ، از  
Nested Query استفاده می کنیم .

مثال :

```
Use [Lab-EasyShop]
Select
    Person.FirstName ,
    Person.LastName ,
    ShopOrder.Date
From
    (Select *      From Customer
     Where ID      Between 1 And 2      ) As Person,
    ShopOrder
Where
    Person.ID = ShopOrder.Customer_ID
```

# تابع *INITCAP*

- In oracle..  
select initcap('abcd') from name;  
returns  
Abcd
- In MS SQL  
select upper(left(colName, 1)) + substring(colName, 2,  
len(colName)) as colName

# Like Operator

اپراتور مقایسه ای Like برای مقایسه مقدار یک رشته با الگویی خاص به کار می رود :

## «الگوی مورد نظر» Like «رشته مورد نظر»

Character	Description	Example
%	همه چیز	Title LIKE '%com%' acomputers , computers : True FirstName LIKE 'aha%' ahang , ahani hangar : True sahab , jahan : False
_ (underscore)	همه چیز با طول یک کاراکتر	FirstName LIKE '_ean' Dean, Sean, Bean : True Alean , Roean : False
[ ]	یکی از اعضای مجموعه	LastName LIKE '[CL]arsen' Carsen & Larsen : True Parsen : False CLarsen: ?
[-]	یکی از اعضای کران مشخص	LastName LIKE '[C-P]arsen' Carsen, Larsen, Karsen, Parsen : True Yarsen : False



# Like Operator

[^]	غیر از یکی از اعضای مجموعه یا کران مشخص	<p>LastName LIKE 'de[^l]%'</p> <p>همه فامیلهایی که با de شروع می شوند اما حرف بعدی آنها l نیست .</p> <p>debtor , dekka , demote : True</p> <p>delta , delvar : False</p>
[%]	%	<p>Operation LIKE '5[%]'</p> <p>5% : True</p> <p>25% : False</p>
[[]]	[	<p>Operation LIKE 'N[[]'</p> <p>N[ : True</p> <p>N[m] : False</p>
[_]	_	<p>Operation LIKE '[_]n'</p> <p>_n : True</p> <p>m_n : False</p>
]	]	<p>Operation LIKE '5]'</p> <p>5] : True</p> <p>25] : False</p>

# Like Operator

برای درک بهتر به چند مثال زیر توجه کنید :

۱- سال شمسی بین ۱۳۳۰ تا ۱۳۹۹

۲- ۵ رقمی بودن شماره دانشجویی

۳- ستون Scode اگر با حرف A شروع شد با ارقام ۱ الی ۵ خاتمه یابد،  
اگر با حرف B شروع شد با ارقام ۶ الی ۹ خاتمه یابد .

آزمایشگاه پایگاه داده

حمیدرضا رمضانی



# Like Operator

برای درک بهتر به چند مثال زیر توجه کنید :

۱- سال شمسی بین ۱۳۳۰ تا ۱۳۹۹

Year Like '[1][3][3-9][0-9]'

۲- ۵ رقمی بودن شماره دانشجویی

StudentNumber Like '[1-9][0-9][0-9][0-9][0-9]'

۳- رشته Scode اگر با حرف A شروع شد با ارقام ۱ الی ۵ خاتمه یابد ، اگر با حرف B شروع شد با ارقام ۶ الی ۹ خاتمه یابد .

Scode Like 'A%[1-5]' Or Scode Like 'B%[6-9]'

# توابع رشته ای در SQL Server

Ascii	مقدار کد Ascii مربوط به کاراکتر مورد نظر را برمی گرداند
Char	یک کد Ascii مربوط را به یک کاراکتر معادل تبدیل می کند
Unicode	مقدار کد Unicode مربوط به کاراکتر مورد نظر را برمی گرداند
NChar	یک کد Unicode مربوط را به یک کاراکتر معادل تبدیل می کند
CharIndex	در یک رشته محل شروع یک عبارت را اعلام می کند
Left	تعداد مشخص کاراکتر از سمت چپ رشته را بر می گرداند
Right	تعداد مشخص کاراکتر از سمت راست رشته را بر می گرداند
Substring	تعداد مشخص کاراکتر از محل مشخص شده در رشته را بر می گرداند
Lower	رشته را به حروف کوچک تبدیل می کند
Upper	رشته را به حروف بزرگ تبدیل می کند
Replace	در کل رشته مورد نظر عبارت خاصی را با عبارت دیگری جایگزین می کند
LTrim	کل فضا های خالی سمت چپ یک رشته را حذف می کند
RTrim	کل فضا های خالی سمت راست یک رشته را حذف می کند
Space	به تعداد مورد نظر ، فضای خالی بر می گرداند
Reverse	یک رشته را بصورت برعکس بر می گرداند
Len	طول یک رشته را بر می گرداند.
Str	یک مقدار عددی را به یک رشته تبدیل می کند.
Replicate	یک رشته را به تعداد مشخص تکرار می کند .