

المحاضرة التاسعة

لغة التحكم بالتدفق والمؤشرات

Control of Flow Language and Cursors

مقدمة: في الكثير من الأحيان نجد أنه من الضروري ألا يتم تنفيذ استعلام ما إلا عند تحقق شرط معين، وقد يكون أحد السيناريوهات المحتملة أيضاً الرغبة في عدم تنفيذ استعلام ما لم يعد استعلام سابق نتيجة متوقعة، أو أن نحتاج إلى تنفيذ استعلامات مختلفة بحسب قيمة معامل محدد مثلاً .

تتيح لغة Transact SQL بعض الكلمات المفتاحية التي تسمح بالتحكم بتدفق التنفيذ لعبارات T-SQL. من دون تعليمات التحكم بالتدفق فإن تعليمات T-SQL ستفذ بالترتيب حسب ورودها أما بوجود تعليمات التحكم بالتدفق فإن التعليمات سترتبط مع بعضها البعض بالطريقة ذاتها كما في لغات البرمجة حيث نجد التعبيرات الشرطية IF.. ELSE و CASE WHEN .. وتعليمات التكرار طالما أن شرط معين محقق كحلقة WHILE.

الكلمات المفتاحية

الكلمة المفتاحية	الشرح
BEGIN ... END	تعريف كتلة من التعبيرات A Statement Block
IF ... ELSE	لتعريف التنفيذ الشرطي (أي تنفيذ كتلة تعليمات عند تحقق شرط منطقي ما) والتنفيذ الاختياري لكتلة أخرى عند عدم تحقق الشرط
CASE	تعيد تعبيراً واحداً من مجموعة من التعبيرات بناء على شرط مقارنة
WHILE	تنفيذ مجموعة من التعليمات طالما أن شرطاً محدداً ما زال

محققاً	
الخروج من حلقة التكرار While الداخلية	BREAK
إعادة تنفيذ حلقة التكرار While	CONTINUE
الخروج غير المشروط	RETURN
متابعة التنفيذ عند التعليمة التالية للعلامة التي اسمها <i>label</i>	GOTO label
انتظار مؤقت للتنفيذ	WAITFOR

1. التعليمة الشرطية **IF...ELSE**
وتستخدم لتنفيذ كتلة تعليمات بناءً على تحقق شرط منطقي ما.

IF *Boolean_expression*

{ *sql_statement* | *statement_block* }

[**ELSE**

{ *sql_statement* | *statement_block* }]

ينفذ الجزء ELSE في حال كان الشرط المنطقي *Boolean_expression* غير محقق (أي يعطي قيمة منطقية FALSE). وهذا الجزء اختياري وليس بالضرورة موجود.

مثال:

في قاعدة المعطيات Pubs نريد تحديد كون وسطي أسعار الكتب أكبر أو أصغر من \$15 مع طباعة العبارة المناسبة

USE pubs

IF (SELECT AVG(price) FROM titles) < \$15

PRINT 'Average title price is less than \$15.'

ELSE

```

IF (SELECT AVG(price) FROM titles ) > $15

    PRINT 'Average title price is more than $15.'

ELSE

    PRINT 'Average title price is equal to $15.'

```

مثال: في قاعدة المعطيات Pubs نريد طباعة بيانات عن الكتب بحسب مجالات السعر: أصغر من 8، بين 8 و 15، وأكبر من 15\$ مع رسالة توضيحية تبين مجال السعر.

Use Pubs

Go

Declare @msg **VarChar**(100)

If (**Select Count**(*) **From** Titles **Where** Price <8) >0

Begin

Set @msg = 'There are several books with prices less than \$ 8. These Books are:'

Print @msg

Select title **From** titles **where** price<8

End

If (**Select Count**(*) **From** Titles **Where** Price **Between** 8 and 15) >0

Begin

Set @msg = 'There are several books with bearable prices. These Books are:'

Print @msg

Select title **From** titles **Where** price **Between** 8 and 15

End

If (**Select Count**(*) **From** Titles **Where** Price > 15) >0

Begin

Set @msg = 'Many books are very expensive. These Books are:'

Print @msg

Select title **From** titles **Where** price > 15

End

Else

Begin

```
Set @msg = 'this msg appears if the condition above is not True'  
Print @msg  
End
```

➤ في حال أبدلنا السعر في آخر تعليمة **IF** لتكون:

```
If (Select Count(*) From Titles Where Price > 25) > 0  
نجد أن الشرط سيكون تقيمه False وبالتالي سيذهب مباشرة إلى ELSE ويطبع  
الرسالة.
```

2. التعليمة CASE

تعيد إحدى القيم الممكنة من مجموعة قيم وذلك بناء على شرط ما.

لهذه التعليمة نمطين:

- النمط البسيط **Simple CASE**: وفيه تتم مقارنة تعبير ما مع مجموعة من التعابير المحتملة لتحديد النتيجة.

- نمط الاكتشاف **Searched CASE**: وفيه يتم اختبار مجموعة من الشروط المنطقية لتحديد النتيجة.

يمكن تضمين جزء ELSE في كلا النمطين.

النمط البسيط:

```
CASE input_expression  
  WHEN when_expression THEN result_expression  
  [ ...n ]  
  [  
    ELSE else_result_expression  
  ]  
END
```

نمط الاكتشاف:

CASE

WHEN *Boolean_expression* **THEN** *result_expression*

[...*n*]

[

ELSE *else_result_expression*

]

END

: input_expression

وهو التعبير المراد مقارنته مع مجموعة القيم الموجودة في النمط البسيط (تعبير المقارنة)

: WHEN when_expression

وهو تعبير بسيط لمقارنة قيمته مع قيمة تعبير المقارنة. نمط هذا التعبير ونمط تعبير المقارنة يجب أن يتطابقا أو أن يكون هناك تحويل ممكن مضمن بينهما.

: THEN result_expression

وهو ناتج التعليمة في حال طابق تعبير المقارنة التعبير البسيط المحدد في فقرة **WHEN** الموافقة.

n : للدلالة على أن الجزء **WHEN...THEN** يمكن أن تتكرر أكثر من مرة.

: ELSE else_result_expression

وهو نتيجة التعليمة في حال كانت قيمة المقارنة مختلفة عن جميع التعابير البسيطة المحددة في الفقرات **WHEN** السابقة له. إذا لم يتم تحديد هذا الجزء وكانت قيمة المقارنة مغايرة لجميع الأنماط البسيطة المحددة فإن ناتج التعليمة هو **NULL**.

مثال: يظهر المثال التالي كيفية استخدام تعليمة **case** من أجل تعديل القيمة الراجعة من حقل. تظهر التعليمة نوع، عنوان، سعر الكتب.

USE pubs

```

GO
SELECT title, Category =
CASE type
WHEN 'popular_comp' THEN 'Popular Computing'
WHEN 'mod_cook' THEN 'Modern Cooking'
WHEN 'business' THEN 'Business'
WHEN 'psychology' THEN 'Psychology'
WHEN 'trad_cook' THEN 'Traditional Cooking'
ELSE 'Not yet categorized'
END,
CAST(title AS varchar(25)) AS 'Shortened Title',
price AS Price
FROM titles

```

مثال:

يمكننا نمط الاستكشاف لتعليمة case من استبدال قيم حقل في النتيجة النهائية.
اكتب تعليمة استعلام عنوان الكتاب، تعيد من أجل كل كتاب "very Reasonable" إذا كان
سعر الكتاب أقل من عشرة، "Coffee Table Title" إذا كان سعر الكتاب بين 10 و 20
(أصغر تماماً من 20) ، "not yet priced" إذا كان السعر null. و إلا تعيد
."Expensive Book"

```

USE pubs
GO
SELECT title, 'Price Category' =
CASE
WHEN price IS NULL THEN 'Not yet priced'
WHEN price < 10 THEN 'Very Reasonable Title'
WHEN price >= 10 and price < 20 THEN 'Coffee Table Title'
ELSE 'Expensive book!'
END,
CAST(title AS varchar(20)) AS 'Shortened Title'
FROM titles
ORDER BY price

```

المثال السابق بعد اجراء تعديل بسيط:

```

SELECT title,
CASE
    WHEN price IS NULL THEN 'Not yet priced'
    when type = 'mod_cook' THEN 'Modern Book'
    WHEN price < 10 THEN 'Very Reasonable Title'
    WHEN price >= 10 and price < 20 THEN 'Coffee Table Title'
    WHEN price =(select price from titles where title_id = 'BU1032')
        THEN 'data base book'
    ELSE 'Expensive book!'
END NewColomn, price , type

FROM titles
ORDER BY price

```

3. حلقة التكرار WHILE

وتستخدم لتنفيذ تعليمة SQL أو كتلة تعليمات طالما أن شرطاً منطقياً مازال محققاً.

ففي SQL Server يستخدم التعبير WHILE مع BEGIN وEND.

```

WHILE Boolean_expression
{ sql_statement | statement_block }
[ BREAK ]
{ sql_statement | statement_block }
[ CONTINUE ]

{statements}

```

Boolean_expression: وهو تعبير منطقي (أي يعيد إما صح أو خطأ). يمكن للتعبير

المنطقي أن يحوي تعليمة Select وفي هذه الحالة يجب وضع تعليمة الانتقال بين أقواس.

{ *sql_statement* / *statement_block* } ويمكن أن تتضمن أي تعليمة SQL أو كتلة

تعليمات SQL. كتلة تعليمات هي مجموعة من التعليمات الموضوعة بين BEGIN و

END.

BREAK: وتؤدي إلى الخروج من تعليمة **WHILE** الداخلية (في حالة وجود أكثر من حلقة **WHILE** منسوجة ضمن بعضها البعض). الكلمة المفتاحية **END** تحدد نهاية الحلقة.

CONTINUE: وتؤدي إلى العودة إلى بداية الحلقة مجدداً وإهمال كافة التعليمات اللاحقة لها.

مثال نريد أن نضاعف أسعار جميع الكتب طالما أن وسطي السعر أقل من \$60 وأعلى سعر لا يتجاوز \$160.

USE pubs

GO

WHILE (SELECT AVG(price) FROM titles) < \$60

BEGIN

UPDATE titles

SET price = price * 2

SELECT MAX(price) FROM titles

IF (SELECT MAX(price) FROM titles) > \$160

BREAK

ELSE

CONTINUE

END

AVG(price)	AVG(price)<60	MAX(price)	MAX(price)>160	price	
14.7662	True	22.95	False	Price*2	Continue
29.5325	True	45.90	False	Price*2	Continue
59.065	True	91.80	False	Price*2	Continue
118.13	False	-	-	-	

Stop while

➤ في حال غيرنا الشروط ليكون وسطي السعر أقل من \$120 وأعلى سعر لا يتجاوز \$90.

WHILE (SELECT AVG(price) FROM titles) < \$120

BEGIN


```

UPDATE titles
  SET price = price * 2
SELECT MAX(price) FROM titles
IF (SELECT MAX(price) FROM titles) > $90
  BREAK
ELSE
  CONTINUE
END

```

AVG(price)	AVG(price)<120	MAX(price)	MAX(price)>90	price
14.7662	True	22.95	False	Price*2
29.5325	True	45.90	False	Price*2
59.065	True	91.80	True	—

Continue
Continue
Break

Go Out
While

4. التعليمة WAITFOR

وتستخدم لإيقاف تنفيذ التعليمة التالية له مدة محددة من الزمن أو بدء التنفيذ في زمن محدد.

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

DELAY : تنفيذ التعليمة التالية بعد مرور الزمن المحدد بالقيمة time (هذا المتحول يجب أن يكون من النمط DateTime).

TIME : تنفيذ التعليمة التالية بحلول التوقيت المحدد بالقيمة time (وهو متحول يجب أن يكون من النمط DateTime).

مثال تقوم التعليمة التالية بإيقاف التنفيذ لمدة ثانيتين ثم تقوم بعرض قائمة بأرقام الموظفين في قاعدة البيانات Northwind.

WAITFOR DELAY '00:00:02'

SELECT EmployeeID **FROM** Northwind.dbo.Employees

مثال عند الساعة العاشرة والنصف و59 ثانية ليلاً أنشئ تقريراً بأرقام وأسماء الموظفين من قاعدة البيانات Northwind.

USE northwind

go

BEGIN

WAITFOR TIME '22:30:59'

SELECT

employeeId,

firstName + ' ' + lastname **as** ' full name'

FROM

employees

ORDER BY employeeid

END

5. **Goto**: تعليمة القفز

مثال:

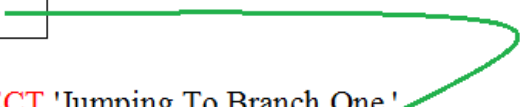
```

DECLARE @Counter INT;
SET @Counter = 1;
WHILE @Counter < 10
BEGIN
    SELECT @Counter
    SET @Counter = @Counter + 1
    IF @Counter = 4 GOTO Branch_One --Jumps to the first branch.
    IF @Counter = 5 GOTO Branch_Two --This will never execute.
END
Branch_One:
    SELECT 'Jumping To Branch One.'
GOTO Branch_Three; --This will prevent Branch_Two from executing.
Branch_Two:
    SELECT 'Jumping To Branch Two.'
Branch_Three:
    SELECT 'Jumping To Branch Three.'

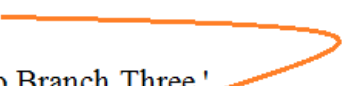
```

@Counter	@Counter<10
1	True
2	True
3	True
4	True

SELECT 'Jumping To Branch One.'



GOTO Branch Three



SELECT 'Jumping To Branch Three.'

المؤشرات CURSORS

الهدف من المؤشرات:

تسمح المؤشرات بالمرور على نتائج تعليمة Select سطرًا سطرًا، ومن أجل كل سطر من الأسطر يمكن إجراء معالجة محددة.

تعريف المؤشرات:

الصيغة النظامية لتعريف مؤشر وفق معيار SQL- 92 هي على الشكل التالي:

```
DECLARE cursor_name [ INSENSITIVE ] [ SCROLL ] CURSOR  
FOR select_statement  
[ FOR { READ ONLY | UPDATE [ OF column_name [ ,...n ] ] } ]
```

الصيغة الممددة لتعريف مؤشر هي على الشكل التالي:

```
DECLARE cursor_name CURSOR  
[ LOCAL | GLOBAL ]  
[ FORWARD_ONLY | SCROLL ]  
[ STATIC | KEYSET | DYNAMIC | FAST_FORWARD ]  
[ READ_ONLY | SCROLL_LOCKS | OPTIMISTIC ]  
[ TYPE_WARNING ]  
FOR select_statement  
[ FOR UPDATE [ OF column_name [ ,...n ] ] ]
```

سوف نستخدم الصيغة الممددة لتعريف مؤشر لذلك سوف نقوم بشرح البنية الثانية:

<i>cursor_name</i>	اسم المؤشر
--------------------	------------

LOCAL	المؤشر داخلي لإجرائية أو كتلة statement block، عند انتهاء تنفيذ الإجرائية أو كتلة التعليمات ينتهي تعريف المؤشر. لكن إذا أسند المؤشر الداخلي (المعرف ضمن إجرائية) إلى متحول معرف output يبقى تعريف المؤشر ويمكن استخدامه خارج الإجرائية.
GLOBAL	المؤشر عام ويمكن استخدامه في جميع الإجرائيات بعد تعريفه.
FORWARD_ONLY	يمكن المرور على نتائج عملية Select باتجاه واحد Next (من السجل الأول إلى السجل الأخير).
SCROLL	يعني أن التجوال على البيانات المعرف عليها المؤشر يمكن أن يكون من الأول إلى الأخير ومن الأخير إلى الأول: (الاحتمالات الممكنة) : Fetch next : إحضار السجل التالي Fetch prior: إحضار السجل السابق Fetch first: إحضار السجل الأول Fetch last: إحضار السجل الأخير
STATIC	يقوم DBMS بإنشاء نسخة عن البيانات المراد التجوال عليها في قاعدة البيانات المؤقتة TEMPDB (نتيجة لذلك لا تؤثر عمليات الحذف والإضافة ضمن الجداول على بيانات المؤشر). يمكن التجوال على بيانات المؤشر في هذه الحالة كالتجوال في حالة الخيار SCROLL.
KEYSET	يقوم مخدم قواعد البيانات بإنشاء جدول في قاعدة البيانات TEMPDB يحوي فقط مفاتيح السجلات التي سوف تستخدم كبيانات للمؤشر. إذا تم تغيير البيانات التي لا تشكل مفتاحاً للسجلات التي سوف يمر عليها المؤشر، سوف يظهر هذا التغيير عند استحضار

	<p>الأسطر.</p> <p>إذا تم حذف (أو تعديل) سجلات مفاتيحها الأولية مخزنة في الجدول في قاعدة البيانات TEMPDB، فإن ذلك سوف يضع قيمة لمتحول عام :</p> <p>@@FETCH_STATUS = -2</p>
DYNAMIC	تظهر جميع التعديلات التي تمت على البيانات أثناء التجوال على البيانات.
READ_ONLY	يمنع هذا الخيار أي تعديل على السجلات المستخدمة من قبل المؤشر (التعديل باستخدام المؤشر نفسه) لكنه لا يمنع التعديل الخارجي عليها.
FAST_FORWARD	تعني هذه التعليمات: FORWARD_ONLY, READ_ONLY
SCROLL_LOCKS	وضع قفل على السجل الحالي الذي يؤشر عليه المؤشر، الأمر الذي يسمح بتعديل السجل الحالي أو حذفه بنجاح.
OPTIMISTIC	لا يضع هذا الخيار أي قفل على السجل الذي يؤشر عليه المؤشر، وبالتالي لا يمكن ضمان نجاح عمليات التعديل والحذف للسجل الحالي.
TYPE_WARNING	يظهر مخدم قواعد البيانات رسالة خطأ إذا تم تحويل المؤشر إلى نمط آخر.
<i>select_statement</i>	تعليمات الاختيار التي تحدد البيانات التي سوف نستخدمها بالمؤشر <i>cursor_name</i>
FOR UPDATE	تحديد الأعمدة التي يسمح المؤشر بتعديلها عبر المؤشر نفسه.

[OF <i>column_name</i>	
[..., <i>n</i>]]	

ملاحظات: بعد تعريف أي مؤشر :

1. يجب فتح المؤشر قبل استخدامه:

Open *cursor_name*

2. يتم إحضار السجل التالي باستخدام التعليمة :

Fetch next from *crusor_name*

3. يتم إحضار السجل السابق باستخدام التعليمة:

Fetch prior from *cursor_name*

4. يتم إحضار السجل الأول باستخدام التعليمة:

Fetch first from *cursor_name*

5. يتم إحضار السجل الأخير باستخدام التعليمة:

Fetch last from *cursor_name*

6. عند إحضار سجل باستخدام إحدى التعليمات السابقة يجب دوماً فحص قيمة المتحول

@@FETCH_STATUS ، فإذا كانت -2 هذا يعني أن السجل لا يمكن إحضاره

إما لأنه قد عدل أو لأنه قد حذف.

7. تنتهي عملية إحضار السجلات عندما تصبح قيمة المتحول

@@FETCH_STATUS < > 0.

(0 يعني نجاح عملية الإحضار، 1- فشل أو أننا خرجنا خارج المؤشر،

2- يعني السجل محذوف)

8. عند الانتهاء من استخدام المؤشر يجب تحرير المؤشر باستخدام التعليمات التالية:

Close *cursor_name*

Deallocate *cursor_name*

مثال سوف نقوم في المثال التالي بتعريف مؤشر يحضر اسم الموظف وكنيته ومن ثم يطبع الاسم الكامل للموظف من قاعدة البيانات northwind.

```
declare @fname varchar(20), @lname varchar(20), @full
varchar(255)
declare cur cursor for
select firstname,      lastname
from employees
order by firstname, lastname
open cur
fetch next from cur into @fname, @lname
while @@fetch_status = 0
begin
    if @@fetch_status <> -2
    begin
        set @full = @fname + ' ' + @lname
        print @full
    end
    fetch next from cur into @fname, @lname
end
close cur
deallocate cur
```


firstname	lastname	@full
Andrew	Fuller	Andrew Fuller
Anne	Dodsworth	Anne Dodsworth
Janet	Leverling	Janet Leverling
Laura	Callahan	Laura Callahan
Margaret	Peacock	Margaret Peacock
Michael	Suyama	Michael Suyama
Nancy	Davolio	Nancy Davolio
Robert	King	Robert King
Steven	Buchanan	Steven Buchanan

مثال

سوف نقوم في هذا المثال باستخدام المؤشرات من أجل بناء تقرير. التقرير المراد بناؤه هو على الشكل التالي:

السطر الأول يحوي اسم المؤلف:

مجموعة الأسطر التالية هي مجموعة الكتب التي ألفها هذا المؤلف.

نكرر العملية السابقة من أجل جميع المؤلفين في قاعدة البيانات pubs.

```
DECLARE @au_id varchar(11), @au_fname varchar(20),
          @au_lname varchar(40),      @message varchar(80),
          @title varchar(80)
```

```
PRINT '----- Authors report -----'
```

```
DECLARE authors_cursor CURSOR FOR
SELECT au_id, au_fname, au_lname
FROM authors
ORDER BY au_id
OPEN authors_cursor
```

```

FETCH NEXT FROM authors_cursor
    INTO @au_id, @au_fname, @au_lname
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT ''
    SELECT @message = '----- Books by Author: ' +
        @au_fname + ' ' + @au_lname
    PRINT @message
    -- Declare an inner cursor based
    -- on au_id from the outer cursor.
    DECLARE titles_cursor CURSOR FOR
    SELECT t.title
    FROM titleauthor ta, titles t
    WHERE ta.title_id = t.title_id AND
        ta.au_id = @au_id -- Variable value from the outer cursor
    OPEN titles_cursor
    FETCH NEXT FROM titles_cursor INTO @title
    IF @@FETCH_STATUS <> 0
        PRINT '        <<No Books>>'
    WHILE @@FETCH_STATUS = 0
        BEGIN
            SELECT @message = '        ' + @title
            PRINT @message
            FETCH NEXT FROM titles_cursor INTO @title
        END
    CLOSE titles_cursor
    DEALLOCATE titles_cursor
    -- Get the next author.
    FETCH NEXT FROM authors_cursor
    INTO @au_id, @au_fname, @au_lname

```

END

CLOSE authors_cursor

DEALLOCATE authors_cursor

au_id	au_fname	au_lname
172-32-1176	Johnson	White
213-46-8915	Marjorie	Green
238-95-7766	Cheryl	Carson
267-41-2394	Michael	O'Leary
274-80-9391	Dean	Straight
341-22-1782	Meander	Smith
409-56-7008	Abraham	Bennet
427-17-2319	Ann	Dull
472-27-2349	Burt	Girglesby
486-29-1786	Charlene	Locksley
527-72-3246	Morningstar	Greene
648-92-1872	Reginald	Blotch-Halls
672-71-3249	Akiko	Yokomoto

au_id	title_id	au_ord	royaltyper
172-32-1176	PS3333	1	100
213-46-8915	BU1032	2	40
213-46-8915	BU2075	1	100
238-95-7766	PC1035	1	100
267-41-2394	BU1111	2	40
267-41-2394	TC7777	2	30
274-80-9391	BU7832	1	100
409-56-7008	BU1032	1	60
427-17-2319	PC8888	1	50
472-27-2349	TC7777	3	30
486-29-1786	PC9999	1	100

title_id	title
BU1032	The Busy Executive's Database Guide
BU1111	Cooking with Computers: Sumptuous Balance Sheets
BU2075	You Can Combat Computer Stress!
BU7832	Straight Talk About Computers
MC2222	Silicon Valley Gastronomic Treats
MC3021	The Gourmet Microwave
MC3026	The Psychology of Computer Cooking
PC1035	But Is It User Friendly?
PC8888	Secrets of Silicon Valley
PC9999	Net Etiquette
PS1372	Computer Phobic AND Non-Phobic Individuals: Beha...
PS2091	Is Anger the Enemy?
PS2106	Life Without Fear
PS3333	Prolonged Data Deprivation: Four Case Studies

-انتهت المحاضرة-