# المؤشرات
# CURSORS

**الهدف من المؤشرات:**

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

**قم بكتابة اجراء يقوم بطباعة رقم المنتج واسم المنتج باستخدام المؤشرات**

```
create procedure [dbo].[display_prod]
as
begin
declare @prodnr int
declare @name varchar(30)
declare cur cursor  for select product_nr,name from product
open cur
fetch from cur into @prodnr,@name
while @@FETCH_STATUS=0
begin
print cast(@prodnr as varchar(30))+' '+@name+char(10)+char(13)
fetch from cur into @prodnr,@name
end
close cur
end
```

**قم بكتابة اجراء يقوم بزيادة سعر المنتج الذي رقمه اكبر من الرقم الممر كبارامتر a وذلك بمقدار نسبة مئوية نمررها له بالبارامتر b للمنتجات التي مجموع مبيعاتها اكبر من 10 ، وبنسبة مئوية مقدارها c للمنتجات التي مجموع مبيعاتها اقل من 10.**

**نقوم بإنشاء تابع يقوم بحساب مجموع مبيعات منتج معين.**

```
create function get_item_sold(@prodnr int ) returns int
as
begin
declare @res int
select @res=sum(quantity) from line where product_nr=@prodnr
return @res
end
```

**نكتب الاجراء المطلوب:**

```
create procedure updateprod(@a int,@b int ,@c int ) as
begin
declare @prodnr int
declare cur cursor for select product_nr from product
where product_nr>@a
open cur
fetch from cur into @prodnr
while (@@fetch_status=0)
begin
declare @totalsold int
```

```
select @totalsold=dbo.get_item_sold(@prodnr)
if @totalsold>10
update product set cost=cost+cost*@b/100 where product_nr=@prodnr
else
update product set cost=cost+cost*@c/100 where product_nr=@prodnr

fetch from cur into @prodnr
end
close cur
end
```

**وظيفة : قم بكتابة اجراء يقوم بتسديد مبلغ مقداره 10% من قيمة الديون المتبقية على الزبون ، اذا كان الزبون قد سدد أكثر من نصف المبالغ المترتبة عليه واستخدم المؤشر لمسح الزبائن واحدا تلو الاخر.**

**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:**

| اسم المؤشر | *cursor_name* |
|---|---|
| المؤشـــر داخلي لإجرائية أو كتلة statement block، عند انتهاء تنفيذ الإجرائية أو كتلة التعليمات ينتهي تعريف المؤشـر. لكن إذا أسـند المؤشـر الداخلي ( المعرف ضـمن إجرائية ) إلى متحول معرف output يبقى تعريف المؤشر ويمكن استخدامه خارج الإجرائية. | LOCAL |
| المؤشر عام ويمكن استخدامه في جميع الإجرائيات بعد تعريفه. | GLOBAL |
| يمكن المرورعلى نتائج عملية Select باتجاه واحد Next (من السجل الأول إلى السجل الأخير). | FORWARD_ONLY |

| | |
|---|---|
| SCROLL | يعني أن التجوال على البيانات المعرف عليها المؤشر يمكن أن يكون من الأول إلى الأخير ومن الأخير إلى الأول: ( الاحتمالات الممكنة ) :<br>Fetch next : إحضار السجل التالي<br>Fetch prior: إحضار السجل السابق<br>Fetch first: إحضار السجل الأول<br>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.<br>إذا تم تغيير البيانات التي لا تشكل مفتاحاً للسجلات التي سوف يمر عليها المؤشر، سوف يظهر هذا التغيير عند استحضار الأسطر.<br>إذا تم حذف (أو تعديل ) سجلات مفاتيحها الأولية مخزنة في الجدول في قاعدة البيانات TEMPDB، فإن ذلك سوف يضع قيمة لمتحول عام :<br>@@FETCH_SATUTS = -2 |
| 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 التعليمة:<br>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. |
| *select_statement* | تعليمة الاختيار التي تحدد البيانات التي سوف نستخدمها بالمؤشر *cursor_name* |
| FOR UPDATE<br>[ OF *column_name* [ ,...*n* ] ] | تحديد الأعمدة التي يسمح المؤشر بتعديلها عبر المؤشر نفسه. |

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

Close *cursor_name*

Deallocate *cursor_name*

## مثال 9

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

```
use northwind
go
declare @fname varchar(20)
declare @lname varchar(20)
declare @msg 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 @msg = @fname + ' ' + @lname
            print @msg
    end
    fetch next from cur into @fname, @lname
end
close cur
deallocate cur
```

## مثال 10:

**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.**

```
use pubs
go
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
```