

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:

اسم المؤشر	cursor_name
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.	LOCAL
المؤشر عام ويمكن استخدامه في جميع الإجرائيات بعد تعريفه.	GLOBAL
يمكن المرور على نتائج عملية Select باتجاه واحد Next (من السجل الأول إلى السجل الأخير).	FORWARD_ONLY

SCROLL	<p>يعني أن التجوال على البيانات المعرف عليها المؤشر يمكن أن يكون من الأول إلى الأخير ومن الأخير إلى الأول: (الاحتمالات الممكنة) :</p> <p>Fetch next : إحضار السجل التالي</p> <p>Fetch prior: إحضار السجل السابق</p> <p>Fetch first: إحضار السجل الأول</p> <p>Fetch last: إحضار السجل الأخير</p>
STATIC	<p>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</p>
KEYSET	<p>يقوم مخدم قواعد البيانات بإنشاء جدول في قاعدة البيانات TEMPDB يحوي فقط مفاتيح السجلات التي سوف تستخدم كبيانات للمؤشر.</p> <p>إذا تم تغيير البيانات التي لا تشكل مفتاحاً للسجلات التي سوف يمر عليها المؤشر، سوف يظهر هذا التغيير عند استحضار 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	<p>تظهر جميع التعديلات التي تمت على البيانات أثناء التجوال على البيانات.</p>
READ_ONLY	<p>يمنع هذا الخيار أي تعديل على السجلات المستخدمة من قبل المؤشر (التعديل باستخدام المؤشر نفسه) لكنه لا يمنع التعديل الخارجي عليها.</p>
FAST_FORWARD	<p>تعني هذه التعليمة:</p> <p>FORWARD_ONLY, READ_ONLY</p>
SCROLL_LOCKS	<p>وضع قفل على السجل الحالي الذي يؤشر عليه المؤشر، الأمر الذي يسمح بتعديل السجل الحالي أو حذفه بنجاح.</p>
OPTIMISTIC	<p>لا يضع هذا الخيار أي قفل على السجل الذي يؤشر عليه المؤشر، وبالتالي لا يمكن ضمان نجاح عمليات التعديل والحذف للسجل الحالي.</p>

TYPE_WARNING	يظهر مخدّم قواعد البيانات رسالة خطأ إذا تم تحويل المؤشر إلى نمط آخر.
select_statement	تعلّمة الاختيار التي تحدد البيانات التي سوف نستخدمها بالمؤشر <i>cursor_name</i>
FOR UPDATE [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 التعلّمة :

Fetch next from *crusor_name*

3. يتم إحضار السجل السابق باستخدام التعلّمة:

Fetch prior from *cursor_name*

4. يتم إحضار السجل الأول باستخدام التعلّمة:

Fetch first from *cursor_name*

5. يتم إحضار السجل الأخير باستخدام التعلّمة:

Fetch last from *cursor_name*

6. عند إحضار سجل باستخدام إحدى التعلّمات 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 <> 0 @@FETCH_STATUS.

8. عند الانتهاء من استخدام المؤشر يجب تحرير المؤشر باستخدام التعلّمات 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

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
```