



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات ٢

رقم الجلسة (٦)

عنوان الجلسة

التوابع المعرفة من قبل المستخدم

المؤشرات Cursors



جدول المحتويات

Contents

رقم الصفحة	العنوان
٣	التوابع المعرفة من قبل المستخدم
٤	أنماط التوابع المعرفة من قبل المستخدم
٥	التوابع المعرفة من قبل المستخدم والتي تعيد نمط جدول
٧	التوابع المعرفة من قبل المستخدم ضمن السياق
٨	المؤشرات CURSORS

الغاية من الجلسة:

- ✓ تمرين الطالب على بناء و تعديل التوابع و استدعاؤها و التعرف على أنواعها المختلفة و حالات استخدامها
- ✓ تعريف الطالب بالمؤشرات و أهمية استخدامها ضمن الاجرائيات المخزنة

التوابع المعرفة من قبل المستخدم User Defined Functions

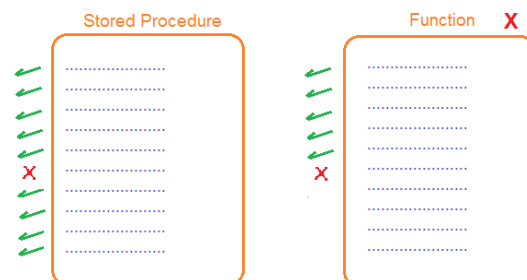
التوابع هي أغراض ضمن قاعدة المعطيات و تقسم في SQL SERVER إلى:

- توابع النظام System Defined Functions.
- التوابع المعرفة من قبل المستخدم User Defined Functions سنتناولها في هذا الدرس.

التوابع تتكون من سلسلة من التعليمات التي تخزن من اجل استخدام لاحق. يتم إنشاء التوابع بالتعليمة CREATE FUNCTION، ويتم تعديل التوابع بالتعليمة ALTER FUNCTION، أما حذف التوابع فيتم بالتعليمة DROP FUNCTION.

الفروقات بين التوابع المعرفة من قبل المستخدم UDF و الاجرائيات المخزنة SP:

التوابع	الاجرائيات المخزنة
يعيد قيمة وحيدة بشكل دائم	يمكن أن لا تعيد أي قيمة أو أن تعيد عدة قيم
يمكن أن يعيد قيمة من نمط جدول	لا يمكن أن يعيد قيمة من نمط جدول
لا نستطيع استدعاء اجرائية مخزنة ضمنه	يمكن استدعاء تابع ضمن الاجرائية
يمكن استخدام التابع ضمن SELECT / where / having	لا يمكن استخدام الاجرائية ضمن SELECT / where / having
لا يمكن استخدام ادارة الأخطاء ضمنه بالتالي وجود خطأ سيتهي تنفيذ التابع حكما	يمكن استخدام ادارة الأخطاء try-catch ضمنه مما يضمن تنفيذ الاجرائية حتى مع وجود خطأ



بعض النقاط المتعلقة بالتتابع:

- ✓ من أجل إنشاء، تعديل، حذف تابع يجب أن تملك صلاحية CREATE FUNCTION.
- ✓ قبل أن تستطيع استخدام تابع في سلسلة من التعليمات يجب أن تعطى الصلاحية لذلك.
- ✓ يمكن أن تدخل التتابع في تعريف الجداول وذلك عن طريق إعطاء قيمة افتراضية DEFAULT أو فرض قيد CHECK CONSTRAINT أو الأعمدة المحسوبة COMPUTED COLUMNS.
- ✓ تأخذ التتابع معاملا Parameter أو أكثر وتعيد قيمة عددية أو جدولا. كما يمكن للتتابع أن تكون بدون معاملات.
- ✓ التتابع التي تعيد قيمة عددية يمكن أن تستخدم في أي تعبير Expression أو أي مكان يستخدم قيمة من نفس النمط التي يعيدها التابع.

أنماط التتابع المعرفة من قبل المستخدم :

- التتابع التي تعيد قيمة عددية Scalar functions.
 - التتابع المعرفة ضمن السياق والتي تعيد جدولا Inline Table-Valued Functions:
 - التتابع التي تحوي أكثر من تعليمية والتي تعيد جدولا Multi-Statement Table-Valued Functions
- من الملاحظ أنه لا يمكن أن تحوي التتابع على تعليمة CREATE. أي أنه لا يمكن إنشاء أغراض جديدة في قاعدة البيانات من ضمن التابع.

مثال:

في المثال التالي سوف نقوم بإنشاء تابعا يولد قيمة تمثل حجم مكعب. وسوف نقوم باستخدام هذا التابع من أجل حساب قيمة محسوبة في جدول في عمود. اسمه BrickVolume

```
create function CubicVolume
(
    @length as int,
    @width int,
    @height as int ) -- input parameters
returns int
begin
    return (@length * @width * @height)
end

-- 1. first call of the function
select dbo.CubicVolume(1,2,3)

-- 2. second call of the function
```

```
declare @x int
select @x = dbo.cubicVolume(1,2,3)
print @x
```

-- 3 create the table that uses the function CubicVolume()

create table Bricks

```
(
    BrickPartNumber int primary key,
    BrickColor varchar(20),
    BrickHeight int,
    BrickLength int,
    BrickWidth int,
    BrickVolume as
    (
        dbo.CubicVolume(BrickHeight,BrickLength,BrickWidth)
    )
)
```

insert into Bricks(BrickPartNumber,BrickColor,BrickHeight,BrickWidth,
BrickLength)

values (1,'dark red',1,2,3)

select * from Bricks

التوابع المعرفة من قبل المستخدم والتي تعيد نمط جدول

التوابع المعرفة من قبل المستخدم والتي تعيد جدولاً يمكن أن تكون بديلاً ممتازاً عن المناظير Views. يمكن أن تستخدم التوابع التي تعيد جدولاً في أي مكان يمكن استخدام جدول أو منظار. بينما تحوي المناظير على تعليمية اختيار واحدة، يمكن للتوابع أن تحوي تعليمات أخرى تمكننا من تنفيذ آلية عمل في التابع.

التوابع المعرفة من قبل المستخدم والتي تعيد جدولا يمكن أن تكون بديلا عن الإجراءات التي تعيد نتيجة واحدة على شكل جدول **Single Result Set**، في هذه الحالة يمكن للتوابع أن تستخدم في فقرة **From** من تعليمة الاختيار **Select** بينما الإجراءات المخزنة لا يمكن أن تستخدم في هذه الفقرة.

ملاحظة:

تعرف التوابع التي تعيد جدولا متحولا داخليا من نمط جدول. يمكن استخدام هذا المتحول الداخلي لعمليات الإضافة والحذف والتعديل. ناتج العمليات السابقة سوف يكون خرج التابع.

مثال

من قاعدة البيانات **Northwind**، نريد قائمة بأرقام وأسماء الشاحنين **Shippers** وأرقام الطلبات وتاريخ الشحن وكلفة الشحن وذلك فقط للطلبات التي يزيد كلفة شحنها عن مقدار محدد. علما أن عمود كلفة الشحن موجود في جدول الطلبات واسمه **orders.Freight**.

```

use Northwind
go
CREATE FUNCTION LargeOrderShippers ( @FreightParm money )
    RETURNS @OrderShipperTab TABLE
        ( ShipperID int,
          ShipperName nvarchar(80),
          OrderID int,
          ShippedDate datetime,
          Freight money
        )
AS
BEGIN
    INSERT @OrderShipperTab
        SELECT S.ShipperID, S.CompanyName,
            O.OrderID, O.ShippedDate, O.Freight
        FROM Shippers AS S INNER JOIN Orders AS O
            ON S.ShipperID = O.ShipVia
        WHERE O.Freight > @FreightParm
    RETURN
END
-- Call the Function in From STATEMENT
select * from LargeOrderShippers($200)

```

التوابع المعرفة من قبل المستخدم ضمن السياق

التوابع المعرفة من قبل المستخدم ضمن السياق هي مجموعة جزئية من التوابع بشكل عام. تعيد هذه التوابع جدولاً وتتألف من تعليمة اختيار واحدة (استعلام واحد). تستخدم هذه التوابع كبديل عن المناظير.

مثال:

إذا طلب منا قائمة بالكتب و أسماء المؤلفين و الكميات المباعة من أجل نوع محدد من الكتب الأعمال مثلاً . إن أحد الخيارات هو إنشاء منظار View على الشكل التالي:

```
CREATE VIEW book_info_v AS
select t.title,t.price,CONCAT(a.au_fname,' ',a.au_lname) as
author_name,sum(s.qty) sales_quantity
from titles t join titleauthor ti on t.title_id = ti.title_id
join authors a on a.au_id= ti.au_id
join sales s on t.title_id = s.title_id
where t.type = 'business'
group by t.title,t.price,CONCAT(a.au_fname,' ',a.au_lname)
```

نلاحظ في هذا المثال أننا قمنا ببناء المنظار بشرط محدد و هو نوع الكتب أعمال و يمكن جعل الاستعلام أكثر ديناميكية ببناء تابع يتم تمرير نوع الكتاب من خلال بارامتر دخل إلى التابع كالتالي :

```
CREATE FUNCTION book_info
( @type nvarchar(12) )
RETURNS table
AS
RETURN (
select t.title,t.price,CONCAT(a.au_fname,' ',a.au_lname) as
author_name,sum(s.qty) sales_quantity
from titles t join titleauthor ti on t.title_id = ti.title_id
join authors a on a.au_id= ti.au_id
join sales s on t.title_id = s.title_id
where t.type = @type
group by t.title,t.price,CONCAT(a.au_fname,' ',a.au_lname)
)
```

--call the function

```
SELECT * FROM book_info('business')
```

ملاحظات:

- لا يوجد حاجة إلى تعريف اسم المتحول الذي سنستخدمه داخليا في جسم التابع كما في حالة التوابع التي تعيد جدولاً ولها جسماً.
- لا يوجد حاجة إلى تعريف بيئة الجدول المعاد من التابع.
- إن وسطاء التابع يمكن أن تستخدم في فلترة البيانات المعادة من قبل التابع.

إعادة كتابة الإجراءات المخزنة كتوابع

- في الحالة العامة: إذا كانت الإجراءية تعيد قيمة **Scalar Value** يجب إعادة كتابة الإجراءية كتابع يعيد القيمة المحسوبة من قبل الإجراءية. وإذا كانت الإجراءية تعيد جدولاً وحيداً **Single Result Set** وأردنا استخدام هذا الجدول في فقرة **From** من تعليمة **Select** فإنه يجب إعادة كتابة الإجراءية كتابع يعيد جدولاً.

المؤشرات **CURSORS**

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

الصيغة النظامية لتعريف مؤشر هي على الشكل التالي:

```
DECLARE cursor_name  
FOR select_statement
```

ملاحظات: بعد تعريف أي مؤشر :

1. يجب فتح المؤشر قبل استخدامه:
Open cursor_name
2. يتم إحضار السجل التالي باستخدام التعليمة:
Fetch next from cursor_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، فهذا يعني أن السجل لا يمكن إحضاره إما لأنه قد عدل أو لأنه قد حذف.



٧. تنتهي عملية إحضار السجلات عندما تصبح قيمة المتحول < @FETCH_STATUS > 0 (0 يعني نجاح عملية الإحضار، ١- فشل أو أننا خرجنا خارج المؤشر، ٢- يعني السجل محذوف)

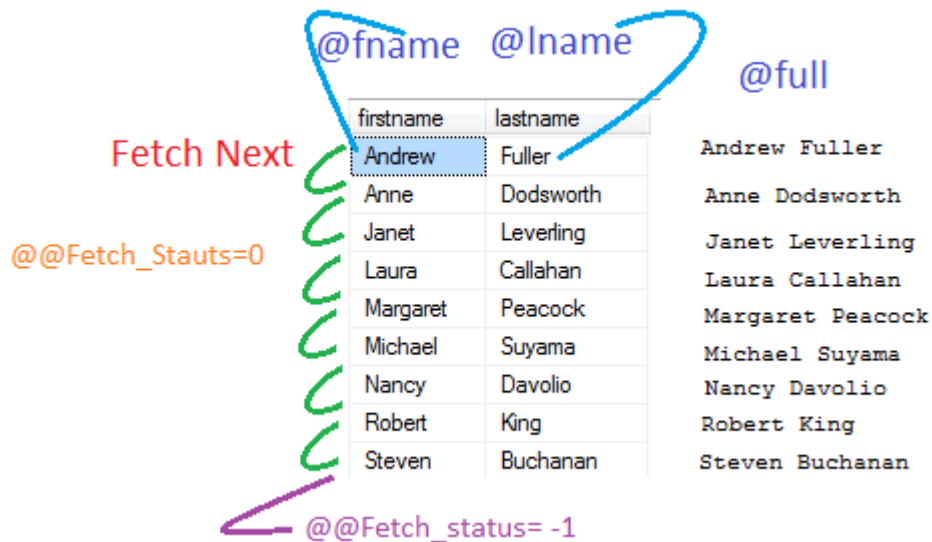
٨. عند الانتهاء من استخدام المؤشر يجب تحرير المؤشر باستخدام التعليمات التالية:
Close cursor_name

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

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



مثال ٢ :

عرض اسم المنتج مع السعر الفردي من جدول المنتجات في قاعدة البيانات Northwind

```

DECLARE
    @product_name VARCHAR(MAX),
    @list_price DECIMAL;
DECLARE cursor_product CURSOR
FOR SELECT productName,UnitPrice FROM products;
OPEN cursor_product;
FETCH NEXT FROM cursor_product INTO
    @product_name,
    @list_price;
WHILE @@FETCH_STATUS = 0
    BEGIN
        PRINT @product_name + ' ' + CAST(@list_price AS varchar);
        FETCH NEXT FROM cursor_product INTO
            @product_name,
            @list_price;
    END;
CLOSE cursor_product;
DEALLOCATE cursor_product;
    
```