



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات 2

رقم الجلسة (4)

عنوان الجلسة

الاجرائيات المخزنة



العام الدراسي: 2023-2024

الفصل الدراسي : الأول



جدول المحتويات

Contents

رقم الصفحة	العنوان
3	إنشاء إجرائية مخزنة :Creating Stored Procedure
4	مثال لاستدعاء إجرائية:
5	أمثلة عن الاجرائيات
6	استعراض إجرائية مخزنة Viewing

الغاية من الجلسة:

في نهاية هذه الجلسة سوف يكون الطالب قادر على :

- ✓ إنشاء إجرائية مخزنة Creating Stored Procedure.
- ✓ تنفيذ الإجراءيات المخزنة Executing.
- ✓ تعديل وإعادة تسمية الإجراءيات المخزنة Modifying and Renaming.
- ✓ إعادة ترجمة الإجراءيات المخزنة Recompiling.
- ✓ استعراض إجرائية مخزنة Viewing.

:Transact- SQL

هي لغة برمجة تشكل الواجهة الرئيسية بين أي نظام يتعامل مع قواعد البيانات وقاعدة البيانات. عند استخدام TSQL هناك طريقتين لتخزين وتنفيذ البرامج:

- يمكن تخزين البرامج محليا Locally. ومن ثم إنشاء برامج تقوم بإرسال البرامج المخزنة محليا إلى مخدم قواعد البيانات للتنفيذ من ثم معالجة النتائج.
- يمكن تخزين البرامج على مخدم قواعد البيانات Stored Procedures، ومن ثم إنشاء برامج تنفذ هذه البرامج المخزنة ومن ثم معالجة النتائج.

فوائد تخزين الإجراءيات على مخدم قواعد البيانات:

- تسمح الإجراءيات المخزنة بالبرمجة الكتلية Modular Programming: يتم إنشاء الإجراءية مرة واحدة، تخزن على مخدم قواعد المعطيات، لينم استدعاؤها عدد من المرات.
- تنفيذ أسرع للإجراءيات Faster Execution: يتم تهجئة Parsing الإجراءية، تحسن الأداء Optimization، وتنفيذ نسخة من الإجراءية موجودة في الذاكرة In Memory بعد تنفيذها لأول مرة.
- تخفيف الضغط على الشبكة Reduce Network Traffics: إذا كان عملية تتطلب تنفيذ مئات من التعليمات على مخدم قواعد البيانات، فإن الإجراءيات المخزنة تساعد بتنفيذها بالاسم.
- يمكن فرض قيود على التنفيذ Execution Security Mechanism: يمكن منح صلاحيات تنفيذ للإجراءيات المخزنة.

الإجراءيات المخزنة في مخدم قواعد المعطيات:

- لها وسطاء متحولات دخل parameters، وتعيد مجموعة من القيم على شكل وسطاء متحولات خرج Output Parameters إلى البرنامج الذي قام بتنفيذ الإجراءية.
- تحوي على مجموعة من التعليمات Statements التي تقوم بمجموعة من العمليات Operations على قاعدة البيانات، كما يمكن استدعاء إجرائية مخزنة من قبل إجرائية مخزنة أخرى.
- تعيد الإجراءيات المخزنة عند التنفيذ متحول حالة Status Parameter لتعلم البرنامج المُستدعي عن نجاح أو فشل العملية.

إنشاء إجرائية مخزنة Creating Stored Procedure:

يمكن إنشاء إجرائية مخزنة باستخدام التعليمة CREATE PROCEDURE. قبل إنشاء إجرائية مخزنة يجب أخذ بعين الاعتبار النقاط التالية:

1. يجب ان يملك صلاحية مالك قاعدة بيانات DBOWNER، كل من يريد أن ينشأ إجرائية مخزنة.
2. يجب تحديد اسم الإجراءية المخزنة. الإجراءية المخزنة هي غرض من أغراض قاعدة البيانات التي يتبع اسمها قواعد تسمية المتحولات.
3. عند إنشاء إجرائية يجب تحديد النقاط التالية:

A. اسم الإجراءية Procedure Name

B. معاملات الدخل Input Parameters

C. معاملات الخرج Output Parameters

القيمة المعادة إلى الإجراءية المستدعية للإجراءية الحالية والتي تدل على فشل أو نجاح الإجراءية. كما يجب تحديد رسالة الخطأ المرافقة للقيمة المعادة.

قواعد لبرمجة الإجراءات المخزنة:

- لا يمكن تضمين تعليمات الإنشاء CREATE التالية في إجرائية مخزنة :

- CREATE PROCEDURE
- CREATE TRIGGER
- CREATE VIEW

- العدد الأعظمي للوسطاء في إجرائية مخزنة 2100 وسيط.
- الحد الأعظمي للمتحولات المحلية ضمن إجرائية محدود فقط بسعة الذاكرة المتوفرة.
- الحجم الأعظمي لإجرائية مخزنة 128MB.

تحديد وسطاء إجرائية

تتخاطب الإجرائية مع البرنامج الذي يستدعيها عبر مجموعة وسطاء Parameters. حيث تمكن الوسطاء البرنامج من تمرير قيم إلى الإجرائية Input Parameters كما تمكنه من الحصول على قيم من الإجرائية Output Parameters.

تحديد الأنماط Specifying a Data Type

يتم تحديد نمط لكل وسيط لإجرائية. تماما كما نعرف نمط اسم حقل في جدول. عند استدعاء إجرائية يجب أن تكون القيم التي تمرر إلى الوسطاء تحترم نمط وحجم الوسيط. مثال إذا كان نمط وسيط هو tinyint فإن الوسيط يجب أن يأخذ قيم طبيعية تخزن على 1 Byte.

مثال لاستدعاء إجرائية:

عند استدعاء إجرائية يمكن أن تمرر القيم إلى الإجرائية بطريقتين. مثال لنفرض أننا أنشأنا إجرائية لها الوسطاء التالية بالترتيب : @first, @second, @third. كما أنه اسم الإجرائية هو sum_proc. يمكن تمرير قيم إلى الإجرائية بإحدى الطريقتين التاليتين:

a. Without Specifying Parameter Names:

```
EXECUTE sum_proc 7, 1, 3
```

b. Specifying Parameters' Names:

```
EXECUTE sum_proc @second = 1, @first = 7, @third = 3
```

في الحالة الأولى تكون قيم وسطاء الدخل : @first = 7, @second = 1, @third = 3

تحديد اتجاه الوسطاء Specifying the Direction of a Parameter

يمكن للإجرائيات المخزنة أن تتلقى قيم عبر الوسطاء من قبل البرنامج المستدعي لها.

مثال 1:

مثال: في قاعدة البيانات Pubs، اكتب إجرائية تقوم بإيجاد سعر كتاب ما title. إذا لم يتم تحديد إي كتاب فإن الإجرائية تعيد معلومات سعر واسم ونوع جميع الكتب.

```
DROP PROCEDURE my_test_out
GO
CREATE PROCEDURE my_test_out
    @bookName varchar(80) = NULL -- This is the input parameter.
AS
BEGIN
    IF @bookName IS NULL
    BEGIN
        SELECT Title, type , price
```

```
FROM titles
```

```
END
ELSE
SELECT price
FROM titles
WHERE title = @bookName
END
```

```
exec my_test_out 'Life Without Fear' ;
declare @out int;
exec @out = my_test_out 'Life Without Fear' ;
print @out ;
```

مثال 2:

من قاعدة البيانات northwind المطلوب بناء اجرائية تقوم بعرض اسم المنتج و الكمية المشتراة من قبل زبون معين يتم تمرير معرف الزبون كبارامتر دخل و يكون خرج الاجرائية اسم المنتج الذي قام بشراؤه الزبون من كافة الطلبيات و الكمية المشتراة

```
use northwind
GO
CREATE PROCEDURE CustOrderHist @CustomerID nchar(5)
AS
SELECT ProductName, Total=SUM(Quantity)
FROM Products P, [Order Details] OD, Orders O, Customers C
WHERE C.CustomerID = @CustomerID
AND C.CustomerID = O.CustomerID AND O.OrderID = OD.OrderID AND
OD.ProductID = P.ProductID
GROUP BY ProductName
يمكن التنفيذ من خلال التعليمة
exec CustOrderHist BOLID
```

مثال 3:

من قاعدة البيانات northwind المطلوب بناء اجرائية تقوم بعرض اسم المنتج و سعره و كميته و مقدار الخصم عليه قيمة المبلغ المطلوب دفعه عن المنتج و يتضمن جداء الكمية ضرب سعر القطعة بعد الخصم و ذلك من أجل رقم طلبية يتم تمريره للإجرائية كبارامتر دخل

```
use northwind
GO
CREATE PROCEDURE CustOrdersDetail @OrderID int
AS
SELECT ProductName,Od.UnitPrice, Quantity
FROM Products P, [Order Details] Od
WHERE Od.ProductID = P.ProductID and Od.OrderID = @OrderID
يمكن التنفيذ من خلال التعليمة
exec CustOrdersDetail 10250
```

تعديل وإعادة تسمية الإجراءات المخزنة.
لحذف إجرائية مخزنة نستخدم التعليمة DROP.

مثال:

```
DROP PROCEDURE get_sales_for_title
```

لإعادة تسمية إجرائية تحذف الإجرائية ثم تنشأ من جديد. هذا الأمر يؤدي إلى حذف جميع الصلاحيات التي أعطيت للإجرائية.
يمكن استخدام التعليمة ALTER PROCEDURE لتعديل إجرائية دون الحاجة إلى استخدام الأمر DROP PROCEDURE.
إذا أردت أن تشفر الإجرائية التي قمت بإنشائها، بحيث لا يستطيع أحد رؤية محتوى الإجرائية فإن ذلك ممكنا عبر استخدام التعليمة WITH ENCRYPTION.

مثال:

```
-- EX 03
USE NORTHWIND
GO
CREATE PROCEDURE MyProc WITH ENCRYPTION
AS
BEGIN
    SELECT * FROM EMPLOYEES
END
```

استعراض إجرائية مخزنة Viewing
لعرض إجرائية مخزنة نستخدم التعليمة التالية: sp_helptext.

```
sp_helptext get_sales_for_title
```