

## Stored Procedures (الإجراءات المخزنة)

**Developers Name:**

**Mohammed Al-Awi**

**Mohammed Al-Shabibi**

**Ahmed Al-Subhi**

**Instructors Name:**

**Karim**

**Fatma Al-Mamari**

## المحتويات

مقدمة

ما هو الإجراء المخزن (Stored Procedure)

الفرق بين PROCEDURE و FUNCTION

مثال عملي: حساب مكافأة الموظف ((Calculate Bonus

مثال عملي ثاني: تغذية الحيوان ((FeedAnimal

استخدامات في شركات كبرى

## المقدمة

### Stored Procedures – الإجراءات المخزنة

#### ما الهدف من المحاضرة؟

هذه المحاضرة تهدف إلى جعلك تفهم:

- ما هي الإجراءات المخزنة Stored Procedures؟
- لماذا نستخدمها؟
- كيف ننشئها خطوة بخطوة؟
- ما الفرق بينها وبين الدوال (Functions)؟
- كيف تُستخدم في أنظمة واقعية مثل الموارد البشرية أو إدارة الحيوانات؟
- كيف تعمل في شركات كبيرة مثل Facebook و Instagram؟

#### ماذا سنتعلم بنهاية المحاضرة؟

- إنشاء إجراء مخزن بنفسك.
- تعديل الإجراء وتشغيله واختباره.
- فهم الأخطاء الشائعة وكيفية اصلاحها.
- تطبيق على أنظمة حقيقية بطريقة عملية.

## ما هو Stored Procedure؟

بالإنجليزية:

Stored Procedure: is a set of SQL statements that are saved in the database and can be executed together as a single unit.

It works like a function in programming: you call it when needed, and it runs a predefined process.

Think of it like a mini-program saved inside the database

بالعربية:

الإجراء المخزن هو مجموعة من أوامر SQL يتم حفظها داخل قاعدة البيانات، وتُنفَّذ كلها دفعة واحدة عند استدعائها.

بدل ما تكتب نفس الكود كل مرة، تخزنه باسم، و تستدعيه متى ما احتجت.

مثال بسيط :

\* مثل زر "حساب الراتب" في نظام الموظفين. تضغطه → يُحسب الراتب، يُضاف البونص، وتُطبع النتيجة.

\* هذا الزر في الحقيقة هو "Stored Procedure".

الفائدة	الشرح بالعربية
إعادة الاستخدام (Reusable)	يمكن استدعاؤه عدة مرات بدون تكرار الكود
الأمان (Security)	يخفي تفاصيل البيانات ويعطي صلاحية محددة
الأداء العالي (Performance)	يتم تنفيذه من الخادم مباشرة مما يجعله أسرع
تقليل الأخطاء (Less Errors)	لأن الكود مكتوب مرة واحدة ومدقق مسبقاً
التنظيم (Modular Logic)	يسهل تقسيم النظام إلى وظائف واضحة ومنفصلة

## الفرق بين FUNCTION و PROCEDURE

العنصر	Stored Procedure	Function (الدالة)
الهدف	تنفيذ عملية كاملة أو أكثر	إرجاع نتيجة واحدة فقط
التركيب	يمكن أن يحتوي على منطق شرطي، حلقات، تحديثات	غالبًا بسيطة ومخصصة لحساب أو تحويل قيمة
يعيد قيمة	ممكن (ليس إجباري)	نعم، يجب أن يعيد قيمة دائمًا
يستخدم داخل SELECT	لا يمكن استخدامه مباشرة في SELECT	نعم يمكن استخدامه داخل SELECT
يمكن تعديل البيانات	نعم مثل UPDATE و INSERT	لا، غالبًا لا يمكنه التعديل فيه
يدعم PRINT	نعم، يمكن طباعة رسائل	لا يدعم الطباعة

```
CREATE FUNCTION GetBonus (@salary FLOAT)
RETURNS FLOAT
AS
BEGIN
    RETURN @salary * 0.10;
END;
```

مثال Function:

```
SELECT Name, dbo.GetBonus(Salary) AS Bonus FROM Employee;
```

الاستخدام:

```
CREATE PROCEDURE CalculateBonus @EmpID INT
AS
BEGIN
    RETURN @salary * 0.10;
END;
```

مثال Procedure:

```
EXEC CalculateBonus @EmpID = 1;
```

الخلاصة:

\* إذا كنت تحتاج تنفيذ عملية كاملة (مثل حساب الراتب، تحديث سجل) → استخدم Procedure.

\* إذا كنت تحتاج تحسب رقم وترجعه فقط → استخدم Function.



مثال عملي: إجراء لحساب المكافأة باستخدام (Calculate Bonus) PROCEDURE

```
--| 1. إنشاء قاعدة البيانات
CREATE DATABASE HRSystem;
GO

-- 2. استخدام قاعدة البيانات
USE HRSystem;

-- 3. إنشاء جدول الموظفين
CREATE TABLE Employee (
    ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Position VARCHAR(100),
    Salary FLOAT
);
GO

-- 4. إدخال بيانات تجريبية
INSERT INTO Employee VALUES (1, 'Ahmed AlBalushi', 'Manager', 1200);
INSERT INTO Employee VALUES (2, 'Fatma AlHinai', 'Accountant', 800);
INSERT INTO Employee VALUES (3, 'Salim AlZadjali', 'Developer', 950);
INSERT INTO Employee VALUES (4, 'Aisha AlBusaidi', 'Intern', NULL);
GO

-- 5. إنشاء الإجراء لحساب البونص
CREATE PROCEDURE CalculateBonus
    @EmpID INT
AS
BEGIN
    DECLARE @Salary FLOAT, @Bonus FLOAT;

    SELECT @Salary = Salary FROM Employee WHERE ID = @EmpID;

    IF @Salary IS NOT NULL
    BEGIN
        SET @Bonus = @Salary * 0.10;
        PRINT 'Employee Bonus: ' + CAST(@Bonus AS VARCHAR);
    END
    ELSE
    BEGIN
        PRINT 'Employee not found or salary is NULL';
    END
END;
GO
```

```

45
46 -- اختبار الإجراء مع رقم موظف موجود
47 EXEC CalculateBonus @EmpID = 2;
48
49
50 -- اختبار الإجراء مع رقم موظف غير موجود
51 EXEC CalculateBonus @EmpID = 99;
52 --
53
54 -- في الراتب NULL اختبار موظف لديه
55 EXEC CalculateBonus @EmpID = 4;
56 --
57

```

اختبار :

```

%
Messages
Employee Bonus: 80
Employee not found or salary is NULL
Employee not found or salary is NULL

Completion time: 2025-05-27T15:11:10.4716164+04:00

```

## شرح كل سطر في الإجراء CalculateBonus

السطر	الشرح بالعربية
CREATE PROCEDURE CalculateBonus	CalculateBonus إنشاء إجراء مخزن جديد اسمه.
EmpID INT	هذا متغير إدخال (باراميتر) يمثل رقم الموظف.
AS	بداية محتوى الإجراء.
BEGIN	بداية تنفيذ مجموعة الأوامر.
DECLARE @Salary FLOAT, @Bonus FLOAT;	تعريف متغيرين داخليين لحفظ الراتب والمكافأة.
SELECT @Salary = Salary FROM Employee WHERE ID = @EmpID;	باستخدام Employee جلب راتب الموظف من جدول رقم الموظف.
IF @Salary IS NOT NULL	التأكد أن الموظف موجود وراتبه غير فارغ.
BEGIN ... END	في حالة تحقق الشرط سيتم تنفيذ ما بداخل هذا الجزء.
SET @Bonus = @Salary * 0.10;	حساب البونص بنسبة 10% من الراتب.
PRINT 'Employee Bonus: ' + CAST ...	(طباعة نتيجة البونص (تحويل الرقم إلى نص).
ELSE	(في حالة لم يكن هناك راتب (أو الموظف غير موجود).
PRINT 'Employee not found ...'	طباعة رسالة توضح عدم وجود الموظف أو عدم وجود راتب.

## مثال عملي 2 : إجراء اطعام الحيوانات في نظام ملجأ الحيوانات PROCEDURE

```
-- إنشاء قاعدة بيانات الحيوانات
CREATE DATABASE AnimalShelterDB;

-- استخدام قاعدة البيانات
USE AnimalShelterDB;
GO

-- إنشاء جدول الحيوانات
CREATE TABLE Animal (
    AnimalID INT PRIMARY KEY,
    Name VARCHAR(50),
    Type VARCHAR(50),
    LastFed DATETIME
);
GO

-- إدخال بيانات الحيوانات
INSERT INTO Animal VALUES (1, 'Bello', 'Dog', '2024-05-20');
INSERT INTO Animal VALUES (2, 'Kitty', 'Cat', '2024-05-24');
INSERT INTO Animal VALUES (3, 'Falcon', 'Bird', NULL);
GO

-- إنشاء إجراء لتحديث وقت الإطعام
CREATE PROCEDURE FeedAnimal
    @AnimalID INT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Animal WHERE AnimalID = @AnimalID)
    BEGIN
        UPDATE Animal SET LastFed = GETDATE() WHERE AnimalID = @AnimalID;
        PRINT 'Animal has been fed.';
    END
    ELSE
    BEGIN
        PRINT 'Animal ID not found.';
    END
END;
GO
```



الاختبار

```

41 -- اختبار الإجراء - تغذية حيوان موجود
42 EXEC FeedAnimal @AnimalID = 3;
43
44
45 -- اختبار رقم حيوان غير موجود
46 EXEC FeedAnimal @AnimalID = 99;
47
48

```

%

Messages

(1 row affected)  
Animal has been fed.  
Animal ID not found.

## شرح كل سطر في الإجراء FeedAnimal

السطر	الشرح بالعربية
CREATE PROCEDURE FeedAnimal	إنشاء إجراء مخزن جديد باسم FeedAnimal.
@AnimalID INT	متغير إدخال يمثل رقم الحيوان.
AS	بداية محتوى الإجراء.
BEGIN	بداية تنفيذ الأوامر.
IF EXISTS (SELECT 1 FROM Animal WHERE AnimalID = @AnimalID)	التحقق من وجود الحيوان في الجدول.
BEGIN	بداية الجزء الذي يتم تنفيذه إذا كان الحيوان موجود.
UPDATE Animal SET LastFed = GETDATE() WHERE AnimalID = @AnimalID;	تحديث تاريخ آخر إطعام للحيوان إلى الوقت الحالي.
PRINT 'Animal has been fed.';	طباعة رسالة تؤكد أن الحيوان تم إطعامه.
END	نهاية كتلة IF.
ELSE	في حالة عدم وجود الحيوان.
PRINT 'Animal ID not found.';	طباعة رسالة خطأ تشير إلى أن رقم الحيوان غير موجود.

## أمثلة حقيقية من الشركات العالمية (Real World Usage)

لماذا هذا الجزء مهم؟

لأنك كمبرمج قد تتساءل:

"أين أستخدم الإجراءات المخزنة؟ هل فعلاً تُستخدم في الشركات؟"

الإجابة: نعم! وتُستخدم كثيراً في كل شركة لديها قواعد بيانات كبيرة ومعقدة.

### أمثلة من الواقع:

#### 1. Facebook (فيسبوك)

- تستخدم إجراءات مخزنة لتحليل تقارير المستخدمين.
- عند الإبلاغ عن محتوى، يتم تشغيل Procedure للتحقق وتسجيل الإبلاغ وتحديث الحالة.
- تُستخدم أيضاً لتوليد تقارير النشاط لكل مستخدم.

#### 2. Instagram (إنستغرام)

- عند رفع صورة، يتم تشغيل إجراءات متعددة:
  - حفظ البيانات.
  - إرسال إشعارات للمتابعين.
  - تسجيل الوقت والموقع.
- تُستخدم الإجراءات لتسجيل وتحديث التعليقات والردود.

### 3. Amazon (أمازون)

- تحديث حالة الطلب يتم عبر Stored Procedure:
  - يتم تغيير الحالة من "جاري التحضير" → "تم الشحن".
  - تنفذ خصم المخزون تلقائيًا.
  - ترسل إشعارًا بالبريد.

### 4. البنوك

- تستخدمها لتطبيق الفوائد البنكية شهريًا تلقائيًا.
- حساب الرسوم عند السحب.
- إرسال تنبيه عند حدوث تحويل مشبوه.

### 5. المستشفيات

- عند تسجيل فحص طبي:
  - يتم تسجيل الوقت.
  - يتم ربط الفحص بالمريض والطبيب.
  - إرسال تنبيه للمريض أو الطبيب.

### خلاصة:

كل شركة تستخدم قواعد بيانات كبيرة تعتمد على الإجراءات المخزنة لأتمتة المهام، وتوفير الوقت، وتقليل الأخطاء.