

SQL Server Triggers

1. Trigger ما هو الـ.

تخيل أن لديك جهازًا إلكترونيًا (مثل الإنذار) يتم تشغيله تلقائيًا عندما يحدث شيء معين، مثل فتح الباب أو كسر الزجاج. الـ يعمل بنفس الطريقة. هو جزء من الكود يتم تشغيله تلقائيًا عندما يتم إدخال بيانات جديدة أو SQL Server Trigger في تعديلها أو حذفها من جدول في قاعدة البيانات.

2. لماذا نستخدم الـ Trigger؟

عندما نريد أن يحدث شيء معين بشكل تلقائي عندما يحدث حدث في قاعدة البيانات. مثلًا Trigger نستخدم الـ

عندما يتم إضافة بيانات إلى جدول معين، يمكننا تسجيل هذه العملية في جدول آخر.

عندما يتم تعديل البيانات، يمكننا التحقق إذا كانت التعديلات صحيحة.

عندما يتم حذف بيانات، يمكننا منع الحذف أو تنفيذ منطق آخر بدلاً من الحذف.

3. أنواع الـ Triggers:

AFTER Trigger: Delete (تعديل بيانات)، أو Update (إضافة بيانات)، Insert يحدث بعد أن تتم عملية معينة مثل (حذف بيانات).

INSTEAD OF Trigger: يقوم هذا النوع يعمل بدلاً من الحدث. مثلاً، بدلاً من تنفيذ عملية الحذف، يمكننا جعل الـ يعمل شيء آخر.

4. مثال عملي لتوضيح الفكرة.

لنفترض أنك تدير متجرًا، وكلما أضفت منتجًا جديدًا، ترغب في تسجيل هذا الحدث في سجل خاص بالتعديلات.

مثال:

CREATE TABLE Products (

```
ProductID INT PRIMARY KEY,  
  
ProductName NVARCHAR(100),  
  
Price DECIMAL(10, 2)  
  
);
```

إنشاء جدول لتسجيل التعديلات 2.

```
CREATE TABLE ProductLog (  
  
    LogID INT PRIMARY KEY IDENTITY,  
  
    ProductID INT,  
  
    Action NVARCHAR(50),  
  
    LogDate DATETIME  
  
);
```

لتسجيل أي عملية إضافة للمنتجات Trigger إنشاء 3.

```
CREATE TRIGGER trgAfterInsert  
  
ON Products  
  
AFTER INSERT  
  
AS  
  
BEGIN  
  
    -- ProductLog نسجل عملية الإضافة في جدول --  
  
    INSERT INTO ProductLog (ProductID, Action, LogDate)  
  
    SELECT ProductID, 'Inserted', GETDATE()  
  
    FROM Inserted
```

END

كيف يعمل المثال؟

Products: عندما تضيف منتجًا جديدًا إلى جدول

sql

INSERT INTO Products (ProductID, ProductName, Price)

VALUES (1, 'Laptop', 1200.50);

الذي أنشأته سيتم تشغيله تلقائيًا **Trigger** الـ

يُظهر أن منتجًا جديدًا قد تم إضافته **ProductLog** بإضافة سجل إلى جدول **Trigger** سيقوم الـ