**شرح التريجر (Trigger) في SQL**

**التريجر (Trigger) في SQL هو أداة بتشتغل تلقائيًا لما يحصل حدث معين على جدول في قاعدة البيانات. ممكن تشوفه كأنه "زر تلقائي" بيشتغل لما حد يضيف (INSERT)، يعدل (UPDATE)، أو يحذف (DELETE) بيانات في الجدول.**

**فايدة التريجر:**

* **ضمان صحة البيانات: التحقق من البيانات المُدخلة.**
* **تسجيل الأحداث: تسجيل أي تعديل أو حذف أو إضافة على البيانات (Logging).**
* **تنفيذ مهام تلقائية: زي تحديث جداول مرتبطة أو القيام بحسابات إضافية.**

**أنواع التريجر:**

**1. حسب نوع الحدث:**

* **INSERT: لما يتم إضافة صف جديد للجدول**
* **UPDATE: لما يتم تعديل البيانات الموجودة في الجدول.**
* **DELETE: لما يتم حذف صف من الجدول.**

**2. حسب التوقيت:**

* **BEFORE: يتم تنفيذ التريجر قبل تنفيذ العملية.**
* **AFTER: يتم تنفيذ التريجر بعد تنفيذ العملية.**

**أمثلة عملية:**

**1. تسجيل الإضافات تلقائيًا (AFTER INSERT):**

**السيناريو:**

**لما يتم إضافة موظف جديد في جدول employees، يتم تسجيل هذه العملية تلقائيًا في جدول audit\_log.**

**-- إنشاء جدول الموظفين**

**CREATE TABLE employees (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(100),**

**position VARCHAR(50)**

**);**

**-- إنشاء جدول تسجيل العمليات**

**CREATE TABLE audit\_log (**

**log\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**action VARCHAR(50),**

**action\_date DATETIME**

**);**

**-- إنشاء التريجر**

**DELIMITER //**

**CREATE TRIGGER after\_employee\_insert**

**AFTER INSERT ON employees**

**FOR EACH ROW**

**BEGIN**

**INSERT INTO audit\_log (action, action\_date)**

**VALUES (CONCAT('Added employee: ', NEW.name), NOW());**

**END;**

**//**

**DELIMITER ;**

**اللي بيحصل:**

* **عند إضافة موظف جديد:**
* **INSERT INTO employees (name, position)**
* **VALUES ('محمد', 'مبرمج');**
* **التريجر يتفعل تلقائيًا ويسجل العملية في جدول audit\_log بالشكل التالي:**
* **Added employee: محمد | 2025-01-10 14:30:00**

**2. التأكد من صحة البيانات (BEFORE INSERT):**

**السيناريو:**

**التأكد أن اسم الموظف يبدأ بحرف كابيتال قبل إضافته في جدول employees.**

**DELIMITER //**

**CREATE TRIGGER before\_employee\_insert**

**BEFORE INSERT ON employees**

**FOR EACH ROW**

**BEGIN**

**IF LEFT(NEW.name, 1) NOT BETWEEN 'A' AND 'Z' THEN**

**SIGNAL SQLSTATE '45000'**

**SET MESSAGE\_TEXT = 'اسم الموظف لازم يبدأ بحرف كابيتال!';**

**END IF;**

**END;**

**//**

**DELIMITER ;**

**اللي بيحصل:**

* **لو الاسم مش بيبدأ بحرف كابيتال، العملية بتفشل وبيظهر رسالة خطأ.**

**3. تسجيل تغييرات الرواتب (AFTER UPDATE):**

**السيناريو:**

**لما يتم تعديل راتب موظف، يتم تسجيل الراتب القديم والجديد في جدول salary\_audit.**

**-- إنشاء جدول تسجيل تغييرات الرواتب**

**CREATE TABLE salary\_audit (**

**audit\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**employee\_id INT,**

**old\_salary DECIMAL(10, 2),**

**new\_salary DECIMAL(10, 2),**

**change\_date DATETIME**

**);**

**DELIMITER //**

**CREATE TRIGGER after\_salary\_update**

**AFTER UPDATE ON employees**

**FOR EACH ROW**

**BEGIN**

**IF OLD.salary != NEW.salary THEN**

**INSERT INTO salary\_audit (employee\_id, old\_salary, new\_salary, change\_date)**

**VALUES (NEW.id, OLD.salary, NEW.salary, NOW());**

**END IF;**

**END;**

**//**

**DELIMITER ;**

**اللي بيحصل:**

* **عند تعديل راتب موظف، التريجر يسجل البيانات القديمة والجديدة في جدول salary\_audit.**

**إدارة التريجرات:**

**حذف التريجر:**

**لو محتاج تلغي تريجر معين:**

**DROP TRIGGER IF EXISTS after\_employee\_insert;**

**نصائح عند استخدام التريجر:**

1. **استخدام بحكمة: التريجرات ممكن تضيف حمل إضافي على قاعدة البيانات، خصوصًا في الجداول الكبيرة.**
2. **اختبر التريجرات بدقة: تأكد أنها بتشتغل زي ما أنت متوقع بدون مشاكل.**
3. **توثيق التريجرات: دايمًا سجل كل التريجرات المستخدمة في المشروع لتسهيل إدارة الكود لاحقًا.**

**خاتمة:**

**التريجر أداة قوية جدًا في SQL، بتساعدك على تنفيذ مهام تلقائية وضمان صحة البيانات. لكن لازم تتعامل معاها بحذر لتجنب أي مشاكل في الأداء أو التعقيد.**