CHƯƠNG 9. TRIGGER AND USER DEFINED FUNCTION (UDF)

A. TRIGGER.

Trigger là 1 loại Stored Procedures đặc biệt được thực hiện 1 cách tự động khi user thực hiện việc cập nhật (insert, update, delete) dữ liệu trên table. Trigger nhằm mục đích đảm bảo sự an toàn về ràng buộc toàn vẹn dữ liệu . Mỗi table có thể có nhiều trigger tương ứng với các hành động insert, delete, update trên table.

Ta gõ vào tên trigger thay cho TRIGGER NAME, và gõ vào các câu lệnh sau từ khóa AS.

- Để kiểm tra cú pháp của các lệnh trong Trigger, ta click nút lệnh **Check Syntax**
- Để xóa trigger, ta chọn tên trigger, sau đó click nút **Delete**

I. CÚ PHÁP:

```
CREATE TRIGGER trigger_name ON { table | view}
  {FOR BEFORE | AFTER | INSTEAD OF }
{ [DELETE] [,] [INSERT] [,] [UPDATE] }
     [NOT FOR REPLICATION]
     AS
        sql_statement [...n]
  }
 hoặc là
  {FOR { [INSERT] [,] [UPDATE] }
     [NOT FOR REPLICATION]
     AS
     {   IF UPDATE (column)
       [{AND | OR} UPDATE (column)]
          [...n]
       | IF (COLUMNS_UPDATED() {bitwise_operator} updated_bitmask)
          { comparison_operator} column_bitmask [...n]
     }
       sql_statement [ ...n]
  }
}
```

Các tham số:

- *trigger_name* : là tên của trigger. Tên trigger phải tuân thủ quy tắc như danh hiệu, và là duy nhất trong cơ sở dữ liệu.
- *table* , view : là tên của table hoặc view mà trên đó trigger được thực hiện để kiểm tra

- BEFORE : Trigger sẽ hoạt động trước khi thao tác lệnh xảy ra
- AFTER : Trigger sẽ hoạt động sau khi thao tác lệnh xảy ra
- INSTEAD OF : Trigger sẽ thực thi thay thế cho cho các thao tác INSERT, UPDATE hoặc DELETE

- { [DELETE] [**,**] [INSERT] [**,**] [UPDATE] } | { [INSERT] [**,**] [UPDATE]} : là các từ khóa cho biết trigger sẽ tự động hoạt động theo lệnh nào.
- NOT FOR REPLICATION : trigger sẽ không hoạt động khi tiến trình nhân bản có thay đổi dữ liệu trên table có trigger.

* Một số table đặc biệt được dùng trong lệnh CREATE TRIGGER: **deleted** and **inserted** là các table logic. Chúng có cấu trúc tương tự như table mà trigger đang hoạt động, và các table này lưu giữ giá trị cũ hay giá trị mới của các mẫu tin đã được thay đổi bởi hành động của user.

- IF UPDATE (*column*) : kiểm tra cột nào đang chịu sự tác động của lệnh INSERT hoặc UPDATE, UPDATE(column) không được sử dụng với lệnh DELETE. Ta có thể kiểm tra nhiều cột
- IF (COLUMNS_UPDATED()) **:** kiểm tra cột nào đang chịu sự tác động của lệnh INSERT hoặc UPDATE. COLUMNS_UPDATED() trả về 1 trị kiểu varbinary để cho ta biết các cột nào trong table đã được chèn hay được hiệu chỉnh dữ liệu.

Ví dụ:

1 Dùng trigger để nhắc nhở:

Ví dụ này sẽ in 1 thông báo đến client khi có 1 ai đó đang thêm hay thay đổi dữ liệu trong table Nhanvien.

```
USE QLVT
IF EXISTS (SELECT name FROM sysobjects
      WHERE name = 'reminder' AND xtype = 'TR')
   DROP TRIGGER reminder
GO

CREATE TRIGGER reminder ON NhanVien
AFTER INSERT, UPDATE
AS RAISERROR (‘Khong duoc them nhan vien moi hoac hieu chinh’, 16, 10)
GO
```

II. TRIGGER KIỂM TRA CÁC THAO TÁC CẬP NHẬT DỮ LIỆU:

1. **Trigger kiểm tra thao tác thêm mẫu tin** :Trigger loại này dùng để kiểm tra mẫu tin thêm vào phải tuân thủ các ràng buộc về khoá chính, và khóa ngoại.

Ví dụ: Tạo trigger Test_ThemSV để kiểm tra khi ta thêm 1 sinh viên mới thì mã lớp của sinh viên đó phải có trước trong table Lop. Nếu mã lớp này chưa có trong table Lop, thì báo lỗi và bỏ qua việc thêm sinh viên đó.

```
CREATE TRIGGER Test_ThemSV  ON dbo.Sinhvien
FOR INSERT
AS
      Declare @Loi int=1
      If exists( Select * from Lop , inserted
                  where Lop.malop=inserted.malop)
        Set @Loi=0
      If @Loi=1
       raiserror( 'Khong the them sinh vien vi ma lop chua ton tai ben table
                  LOP',16,1)
```

2. **Trigger kiểm tra thao tác xóa mẫu tin:** Trigger loại này thường được dùng để kiểm tra ràng buộc về khóa ngoại; Ví dụ như khi ta xóa 1 lớp thì phải đảm bảo các sinh viên của lớp phải được xóa trước.

Ví dụ: Tạo trigger XoaLop để kiểm tra khi ta xoá 1 lớp thì các sinh viên của lớp đó phải được xóa trước. Trong trường hợp vẫn còn sinh viên thuộc lớp thì xem như thao tác xóa không hoàn thành.

```
CREATE TRIGGER XoaLop ON [Lop]
FOR DELETE
AS
if  (select count(*) from sinhvien sv, deleted where sv.malop=deleted.malop) > 0
 begin
    raiserror( 'Khong the xoa lop nay vi sinh vien van con trong co so du lieu',16,1)
with nowait
/*   print  'Khong the xoa lop nay vi sinh vien van con trong co so du lieu' */
 end
```

3. **Trigger kiểm tra thao tác hiệu chỉnh dữ liệu:** Viết 1 trigger kiểm tra việc hiệu chỉnh mã sinh viên trên table Sinhvien. Nếu mã sinh viên ta đang thay đổi đã được nhập điểm thì báo lỗi và không cho phép thực hiện việc hiệu chỉnh đó.

```
CREATE TRIGGER Upd_Masv  ON [dbo].[SINHVIEN]
FOR  UPDATE
AS
    if @@rowcount = 0
            return
    if update(masv)
      if exists(Select * from Diem d, deleted sv Where d.masv=sv.masv)
```

```
        begin
         raiserror ('Khong the sua ma sinh vien vi sinh vien nay da duoc nhap
                                                          diem',16,10) with nowait
          rollback transaction
        end
        else   print 'Da hieu chinh ma sinh vien '
    return
```

Bài tập

1. Tạo Trigger TR_AfterInsert_CTPN để cập nhật số lượng tồn trong bảng VATTU khi ta lập 1 phiếu nhập

```
ALTER TRIGGER  [dbo].[TR_AfterInsert_CTPN]
   ON  [dbo].[CTPN]
   AFTER INSERT
AS
BEGIN

    UPDATE VATTU
        SET SOLUONGTON= SOLUONGTON + (SELECT SOLUONG FROM inserted)
    WHERE MAVT = (SELECT MAVT FROM inserted)

END
```

2. Tạo Trigger CapNhat_SLTon_for_delete để cập nhật số lượng tồn trong bảng VATTU khi ta xóa 1 dòng trong bảng CT_PN

3. Tạo Trigger TR_AfterUpdate_CTPN để cập nhật số lượng tồn trong bảng VATTU khi ta thay đổi field số lượng của 1 dòng trong bảng CTPN

```
CREATE TRIGGER  TR_AfterUpdate_CTPN
   ON  CTPN    AFTER  UPDATE
AS
BEGIN
IF (UPDATE(SOLUONG))
BEGIN
       UPDATE VATTU
     SET SOLUONGTON= SOLUONGTON -(SELECT SOLUONG FROM deleted) +(SELECT SOLUONG
                                                          FROM inserted)
               WHERE MAVT = (SELECT MAVT FROM inserted)
END
-- Trường hợp hiệu chỉnh field ???  sẽ ảnh hưởng tới số lượng tồn
END
```

4. TƯƠNG TỰ CHO CÁC THAO TÁC INSERT, UPDATE, DELETE TREN TABLE CTPX