

# CƠ SỞ DỮ LIỆU

## BÀI THỰC HÀNH SỐ 5

### 1 Nội dung thực hành

- Xây dựng các câu RBTV bằng Trigger.
- Thực hiện các bài tập về RBTV của CSDL Quanlibanhang.

### 2 Hướng dẫn

#### 2.1 Giới thiệu Trigger:

- Trigger là một kỹ thuật lập trình đặc biệt trong SQL, dùng để thực hiện các loại ràng buộc toàn vẹn có tính phức tạp, nó sẽ có hiệu lực khi chúng ta thay đổi dữ liệu trên *một bảng dữ liệu* cụ thể, hoặc các lệnh xử lý làm thay đổi dữ liệu của các lệnh: **insert, update, delete**. Trong trigger có thể chứa các lệnh truy vấn từ các bảng khác hoặc bao gồm những lệnh truy vấn phức tạp.
- Trigger *chạy một cách tự động*: chúng được kích hoạt ngay tức thì khi có sự thay đổi dữ liệu trên bảng dữ liệu.

#### 2.2 Cú pháp:

##### 2.2.1 Lệnh tạo trigger:

```
Create Trigger trigger_name on table_name
For [insert, update, delete]
As
Begin
    Khai báo các xử lý
End
```

**Lưu ý:** trigger sau khi đã tạo, để chỉnh sửa nội dung các khai báo xử lý thì ta thay từ khóa **Create** bằng từ khóa **Alter** => sau đó cho thực thi lại trigger (bôi đen nhấn F5) thì nội dung đó mới có hiệu lực.

### 2.2.2 Lệnh xóa trigger:

**Drop Trigger** trigger\_name

**Lưu ý:** trường hợp chỉ muốn ngưng kiểm tra trigger thì ta sử dụng lệnh sau:

**Alter table** table\_name **Disable/Enable Trigger** trigger\_name

**Hoặc Alter table** table\_name **Disable/Enable Trigger** ALL

## 2.3 Một số thao tác liên quan đến trigger:

### 2.3.1 Thao tác dữ liệu:

- Khi người dùng thực hiện thao tác **insert** dữ liệu vào trong các quan hệ thì hệ thống sẽ tự động tạo ra bảng tạm với tên **INSERTED** có cấu trúc tương tự như bảng dữ liệu mà ta muốn insert dữ liệu vào.
- Khi người dùng thực hiện thao tác **delete** dữ liệu các quan hệ thì hệ thống sẽ tự động tạo ra bảng tạm với tên **DELETED** có cấu trúc tương tự như bảng dữ liệu mà ta muốn delete dữ liệu.
- Khi người dùng thực hiện thao tác update dữ liệu các quan hệ thì hệ thống sẽ thực hiện 2 thao tác sau:

    • Tạo ra bảng **DELETED** để chứa dữ liệu cũ

    • Tạo ra bảng **INSERTED** để chứa dữ liệu mới

### 2.3.2 Thao tác lập trình:

- Khai báo biến: **DECLARE**
- Xuất thông tin: **PRINT, RAISE**
- Cấu trúc điều kiện: **IF**
- Cấu trúc lặp: **FOR**
- Con trỏ: **CURSOR**
- Hủy cập nhật dữ liệu vào hệ thống bộ nhớ: **ROLLBACK TRAN** hoặc **ROLLBACK TRANSACTION**

## 3 Bài tập

**Bài 1.** Sinh viên làm quen với việc cài đặt trigger cho các RBTV cơ bản sau:

- a) Khi thêm một khách hàng thì hiển thị thông báo “Khách hàng đã thêm thành công”
- b) Khi thêm 1 hoá đơn thì hiển thị thông báo “Hoá đơn của khách hàng <Họ tên khách hàng> đã được thêm thành công”
- c) Khi xoá một CTHD thì hiển thị thông báo “CTHD của <SoHD> đã bị xoá”
- d) Khi chỉnh sửa tên sản phẩm của một sản phẩm thì cho biết sản phẩm đó của nước nào sản xuất

**Hướng dẫn câu a:**

```
CREATE TRIGGER trg_Baila ON KHACHHANG
FOR INSERT
AS
BEGIN
    Print 'Khách hàng đã thêm thành công'
END
```

**Kiểm tra lại trigger có hoạt động?**

```
INSERT INTO KHACHHANG(MAKH) VALUES('KH50')
```

**Hướng dẫn câu b:**

```
CREATE TRIGGER trg_Kiemtra_NhapHD ON HOADON
FOR INSERT
AS
BEGIN
    -- Dùng từ khoá để khai báo biến
    DECLARE @MAKH CHAR(4)
    DECLARE @HOTEN VARCHAR(40)
    -----
    SELECT @MAKH=MAKH FROM INSERTED
    SELECT @HOTEN=HOTEN
    FROM KHACHHANG
    WHERE MAKH=@MAKH
```

```
PRINT 'Hoa đơn của khách hàng '+@HOTEN+' đã được
them thanh cong'
END
```

**Bài 2.** Sinh viên hoàn thành Phần I bài tập Quanlibanhang (từ câu 11 đến câu 15)

**Hướng dẫn câu 11:** Ngày mua hàng (NGHD) của một khách hàng thành viên sẽ lớn hơn hoặc bằng ngày khách hàng đó đăng ký thành viên (NGDK).

Các bước để giải một câu RBTV phức tạp:

- Xác định bối cảnh: KHACHHANG, HOADON
- Xác định bảng tâm ảnh hưởng:
  - + Dấu “+” cho biết khi ta thao tác dữ liệu sẽ bị vi phạm
  - + Dấu “-” cho biết khi ta thao tác dữ liệu không bị vi phạm

	Thêm	Xóa	Sửa
KHACHHANG	-	-	+(NGDK)
HOADON	+	-	+(NGHD)

- Thực hiện cài đặt trigger tại những ô có dấu “+”

**-- Khi thêm 1 HD sẽ vi phạm RBTV**

**-- Cách 1:**

```
CREATE TRIGGER trg01_ins_HOADON ON HOADON
```

```
FOR INSERT
```

```
AS
```

```
BEGIN
```

```
-- Khai báo biến để chứa dữ liệu từ bảng INSERTED (bảng
nay có cấu trúc tương tự bảng HOADON) khi ta thêm dữ
liệu vào bảng HOADON
```

```
DECLARE @MAKH varchar(4);
```

```
DECLARE @NGHD smalldatetime;
```

```
SELECT @MAKH=MAKH,@NGHD=NGHD FROM INSERTED
```

```
-- Khai báo biến để chứa dữ liệu từ bảng KHACHHANG
```

```
DECLARE @NGDK smalldatetime;
```

```
SELECT @NGDK=NGDK
FROM KHACHHANG
WHERE MAKH=@MAKH

IF (@NGDK>@NGHD)
BEGIN
    PRINT 'ERROR: NGDK phai < NGHD'
    ROLLBACK TRAN
END
ELSE
BEGIN
    PRINT 'HOA DON DA THEM THANH CONG'
END
END

-----
-- Cách 2:
CREATE TRIGGER trg01_ins_HOADON ON HOADON
FOR INSERT
AS
BEGIN
    IF (SELECT COUNT(*) FROM KHACHHANG K,INSERTED I
        WHERE K.MAKH=I.MAKH AND K.NGDK>I.NGHD)>0
    BEGIN
        PRINT 'ERROR: NGDK phai < NGHD'
        ROLLBACK TRAN
    END
    ELSE
    BEGIN
        PRINT 'HOA DON DA THEM THANH CONG'
    END
END
END
```

**Cài đặt tương tự cho các dấu cộng còn lại.**

**Bài 3.** Sinh viên hoàn thành bài tập Quanligiaovu.