

/\*

Hồ Phúc Lâm

Lop DHTMDT17

Ôn tập trigger - BÀI TẬP TUẦN 7

31/07/2024

\*/

/\*

Trigger chỉ có thể được kích hoạt một cách tự động bởi một trong các lệnh Insert, Update, Delete.

☑ Khi một trigger được kích hoạt:

o Dữ liệu được insert hoặc update sẽ được chứa trong bảng Inserted

o Dữ liệu bị delete được chứa trong bảng Deleted.

Inserted và Deleted là 2 bảng tạm chỉ có giá trị bên trong trigger, thường dùng để kiểm tra dữ liệu trước khi commit hay roll back

☑ Có thể áp dụng trigger cho View

Loại trigger:

o INSTEAD OF trigger: bỏ qua các hành động kích hoạt trigger (các thao tác insert, delete, update), thay vào đó nó sẽ thực hiện các câu lệnh bên trong trigger. Thường dùng trong View với các chức năng sau:

☑ Cập nhật nhiều bảng trong view cùng một lúc

☑ Tăng điều kiện ràng buộc trên các thuộc tính so với CHECK

☑ Đánh giá trạng thái của bảng trước và sau khi cập nhật dữ liệu.

☑ Cho phép từ chối một phần các câu lệnh và thực thi phần còn lại

o AFTER trigger (FOR): được thực thi sau khi các câu lệnh SQL thực thi thành công. AFTER trigger là trigger mặc định khi dùng từ FOR.

☑ Không dùng trong view

\*/

```

use Northwind
Go

/*
1.Tạo một Instead of trigger thực hiện trên view.
*/
--Tạo mới 2 bảng MEmployees và MDepartment theo cấu trúc sau:
create table MDepartment(
    DepartmentID int not null primary key,
    Name nvarchar(50),
    GroupName nvarchar(50)
)
Go

create table MEmployees(
    EmployeeID int not null primary key,
    FirstName nvarchar(50),
    MiddleName nvarchar(50),
    LastName nvarchar(50),
    DepartmentID int foreign key references MDepartment(DepartmentID)
)
Go

--Tạo một view tên EmpDepart_view bao gồm các field: EmployeeID,
--FirstName, MiddleName, LastName, DepartmentID, Name, GroupName, dựa
--trên 2 bảng MEmployees và MDepartment.
create view EmpDepart_view
as
    select EmployeeID, FirstName, MiddleName, LastName, md.DepartmentID, Name, GroupName
    from MDepartment md
    join MEmployees me on me.DepartmentID = md.DepartmentID
Go
--xem lại view

```

```
select * from EmpDepart_view
Go
```

```
--Tạo một trigger tên InsteadOf_Trigger thực hiện trên view
--EmpDepart_view, dùng để chèn dữ liệu vào các bảng MEmployees và
--MDepartment mỗi khi chèn một record mới thông qua view EmpDepart_view.
```

```
create trigger InsteadOf_Trigger
on EmpDepart_view
instead of insert
as
begin
    insert into MEmployees(EmployeeID, FirstName, MiddleName, LastName,DepartmentID)
    select EmployeeID, FirstName, MiddleName, LastName,DepartmentID
    from inserted

    insert into MDepartment(DepartmentID, Name, GroupName)
    select DepartmentID, Name, GroupName
    from inserted
end
Go
```

```
--Kiểm tra trigger
insert EmpDepart_view values(1, 'Ho', 'Phuc', 'Lam', 12, 'Project Manager', 'IT');
insert EmpDepart_view values(2, 'Nguyen', 'Hoang', 'Huy', 13, 'Marketing', 'Sales')
Go
```

```
--Xem lại các bảng
select * from EmpDepart_view
select * from MEmployees
select * from MDepartment
Go
```

```

/*
2.Tạo một trigger thực hiện trên bảng MSalesOrders có chức năng thiết lập độ ưu
tiên của khách hàng (CustPriority) khi người dùng thực hiện các thao tác Insert,
Update trên bảng MSalesOrders theo điều kiện như sau:
--Nếu năm ký đơn đặt hàng là 1996 thì độ ưu tiên của khách hàng (CustPriority) là 1
--Nếu năm ký đơn đặt hàng là 1997 thì độ ưu tiên của khách hàng (CustPriority) là 2
--Nếu năm ký đơn đặt hàng là 1998 trở đi thì độ ưu tiên của khách hàng (CustPriority) là 3
*/
create table MCustomers(
    CustomerID nchar(5) not null primary key,
    CustPriority int
)
Go

create table MSalesOrders(
    SalesOrderID int not null primary key,
    OrderDate date,
    CustomerID nchar(5) foreign key references
    MCustomers (CustomerID)
)
Go

--Chèn dữ liệu cho bảng MCustomers, lấy dữ liệu từ bảng Customers, nhưng
--chỉ lấy CustomerID bắt đầu bằng ký tự A và B, cột custpriority cho giá trị null.
insert into MCustomers(CustomerID, CustPriority)
    select CustomerID, null
    from Customers
    where CustomerID like 'A%' or CustomerID like 'B%'
Go

--Chèn dữ liệu cho bảng MSalesOrders, lấy dữ liệu từ bảng Orders
--chỉ lấy những hóa đơn của khách hàng có trong bảng MCustomers
insert into MSalesOrders (SalesOrderID, OrderDate, CustomerID )

```

```
select OrderID, OrderDate, CustomerID
from Orders
where CustomerID in (select CustomerID from MCustomers)
```

Go

```
--Viết trigger, lấy dữ liệu từ bảng inserted và deleted
create trigger tr_IsUpdateInsertOfMSalesOrders
on MSalesOrders
after insert, update
as
begin
    update MCustomers
        set CustPriority = case
            when year(i.OrderDate) = 1996 then 1
            when year(i.OrderDate) = 1997 then 2
            when year(i.OrderDate) >= 1998 then 3
        end
    from MCustomers mc
    join inserted i on i.CustomerID = mc.CustomerID
end
```

Go

```
--Kiem tra thuc thi
insert into MCustomers (CustomerID, CustPriority) values ('AIUH', null)
Go
select * from MCustomers
Go
insert into MSalesOrders (SalesOrderID, OrderDate, CustomerID) values ('12', '1996-07-28', 'AIUH')
Go
select * from MCustomers
Go
```

```
/*
```

3. Viết một trigger thực hiện trên bảng MProducts sao cho khi người dùng thực hiện chèn thêm một sản phẩm mới vào bảng MProducts thì chương trình cập nhật số sản phẩm trong cột NumOfProduct của bảng MSuppliers. Nếu tổng số sản phẩm của nhà cung cấp tương ứng  $\leq 5$  thì cho phép chèn thêm, ngược lại thì hiển thị thông báo “Nhà cung cấp đã đủ sản phẩm” và hủy giao tác. Các bước thực hiện:

```
*/
```

--Tạo mới 2 bảng MProducts và MSuppliers theo cấu trúc sau:

```
create table MSuppliers(  
    SupplierID int not null primary key,  
    CompanyName nvarchar(40),  
    NumOfProduct int  
)  
Go  
Create table MProducts(  
    ProductID int not null primary key,  
    ProductName nvarchar(40),  
    SupplierID int foreign key references MSuppliers(SupplierID)  
)  
Go
```

--Chèn dữ liệu cho bảng MSuppliers, lấy dữ liệu từ bảng Suppliers, cột  
--NumOfProduct gán giá trị NULL, bảng MProducts lấy từ bảng Products

```
insert into MSuppliers(SupplierID, CompanyName, NumOfProduct)  
    select SupplierID, CompanyName, null  
    from Suppliers  
Go  
insert into MProducts(ProductID, ProductName, SupplierID)  
    select ProductID, ProductName, SupplierID  
    from Products  
Go
```

```

--Viết trigger theo yêu cầu trên và viết câu lệnh hiện thực trigger
create trigger tr_InsertMProduct
on MProduct
after insert
as
begin
    declare @SupID int, @CountProduct int;
    select @SupID = SuppliersID from inserted
    select @CountProduct = count(*) from MProducts where SupplierID = @SupID

    if @CountProduct > 5
        begin
            raiserror('Nha cung cap da du san pham', 16, 1);
            rollback;
        end
    else
        begin
            update MSuppliers
            set NumOfProduct = @CountProduct
            where SupplierID = @SupID
        end
    end
end
Go

--thuc thi trigger - chen du lieu vao mproducts
insert into MProducts (ProductID, ProductName, SupplierID) values (501, 'Iphone 15PRO', 1);
Go

--xem lai cac bang
select * from Products
select * from Suppliers
select * from MProducts

```

```
select * from MSuppliers
```

```
--Xem ma supplierid =1 trong bang msupplier
```

```
select * from MSuppliers where SupplierID = 1;
```

```
Go
```

```
--xem san tat ca san pham co ma supplierid=1
```

```
select * from MSuppliers ms, MProducts mp where ms.SupplierID =mp.SupplierID and mp.SupplierID = 1
```

```
Go
```

```
/*
```

```
4.Viết trigger cho thao tác Insert của bảng Products. Khi có thao tác chèn vào bảng Products thì đưa ra một thông báo là 'Có <n> mẫu tin được chèn'.
```

```
*/
```

```
create trigger tr_InsertOfProducts
```

```
on Products
```

```
after insert
```

```
as
```

```
begin
```

```
declare @numCount int
```

```
select @numCount = count(*) from inserted
```

```
end
```

```
Go
```

```
--Kiem tra
```

```
insert into Products ( ProductName, SupplierID) values ( 'SamSung 20A', 1), ( 'SamSung 22B', 2);
```

```
Go
```

```
select * from Products
```

```
Go
```

```
drop trigger tr_InsertOfProducts
```

```
Go
```



```

--cách khác sử dụng biến rowcount
create trigger tr_InsertOfProducts
on Products
after insert
as
begin
    declare @numCount int
    set @numCount = @@ROWCOUNT
    print(N'Có ' + convert(nvarchar,@numCount) + N' mẫu tin được chèn!!!')
end
Go

```

/\*  
 5. Viết trigger cho thao tác Insert trên bảng Orders. Sau khi có mẫu tin được chèn vào bảng Orders thì mẫu tin đó cũng được chèn vào bảng Orders\_Backup. Lưu ý: nếu chưa có bảng Orders\_Backup thì tạo Orders\_Backup (có cấu trúc hoàn toàn giống như bảng Orders) trước khi kiểm chứng trigger.  
 \*/

```

--xem bảng orders
select * from Orders
exec sp_help Orders
Go

```

```

--Tạo bảng Orders_Backup
create table Order_Backup(
    OrderID    int not null primary key,
    CustomerID nchar(10),
    EmployeeID int,
    OrderDate  datetime,
    RequiredDate  datetime,
    ShippedDate  datetime,
    ShipVia     int,

```

```

    Freight    money,
    ShipName   nvarchar(80),
    ShipAddress nvarchar(120),
    ShipCity   nvarchar(30),
    ShipRegion nvarchar(30),
    ShipPostalCode nvarchar(20),
    ShipCountry nvarchar(30),
)
Go

--viết trigger
create trigger tr_InsertOrdersBackup
on orders
after insert
as
begin
    insert into Order_Backup select * from inserted
end
Go

--kiểm tra thực thi
insert into Orders(CustomerID, EmployeeID, OrderDate) values ('ALFKI', 1, 2024-02-08)
Go
select * from Customers
select * from Employees
Go

--xem lại kết quả bảng order-backup
select * from Orders
select * from Order_Backup
Go

/*

```

6. Vi t trigger cho thao t c Insert, Update, Delete tr n b ng Order Details. Khi c  m u tin đ c ch n v o ho c hi u ch nh ho c xo  th  c p nh t l i c t TongTriGia trong b ng Orders v i TongTriGia = T ng ti n c a Quantity \* UnitPrice. L u  : n u b ng Orders ch a c t TongTriGia th  b  sung v o tr oc khi ki m ch ng trigger.

```
*/
--xem l i c c b ng c n th c hi n
select * from [Order Details]
select * from Orders

--Th m c t TongTriGia v o table Orders
alter table orders add TongTriGia money
Go
select * from Orders
Go

--t o trigger
create trigger tr_InsertUpdateDeleteOrderDetails
on [dbo].[Order Details]
after insert, update, delete
as
begin
    update o
    set TongTriGia = ISNULL((
        select sum(od.Quantity * od.UnitPrice)
        from [Order Details] od
        where od.OrderID =o.OrderID
    ),0)
    from Orders o
    where o.OrderID in (select distinct OrderID from inserted) or o.OrderID in (select distinct
OrderID from deleted)
end
Go
```

```

--kiểm tra thực thi
--Chèn một mẫu tin mới vào bảng Order Details
insert into [Order Details] (OrderID, ProductID, Quantity, UnitPrice)
values (10248, 77, 5, 10.5);
Go

-- Cập nhật số lượng hoặc giá của mẫu tin trong Order Details
update [Order Details] set Quantity = 6 where OrderID = 10248 and ProductID = 77;
Go

-- Xóa một mẫu tin từ bảng Order Details
delete from [Order Details] where OrderID = 10248 and ProductID = 77;
Go

-- Kiểm tra dữ liệu trong bảng Orders để xem cột TongTriGia đã được cập nhật
select * from Orders
Go

/*
7.Viết trigger cho thao tác Insert, Update để kiểm tra ràng buộc liên thuộc tính
liên quan giữa Giá Bán (UnitPrice) trong [Order Details] và Giá Gốc (UnitPrice)
trong bảng Products như sau: Giá Bán trong [Order Details] luôn luôn lớn hơn
Giá Gốc trong Products, nếu vi phạm thì thông báo và không cho phép Insert hay
Update.
*/
use Northwind
Go
create trigger tr_CheckInsertOrderDetails
on [dbo].[Order Details]
after insert, update
as
begin

```

```

        if exists(select 1 from inserted i, Products p
                  where i.ProductID = p.ProductID
                  and i.UnitPrice <= p.UnitPrice)
        begin
            rollback;
            print(N'Lỗi Giá bán nhỏ hơn Giá mua gốc!!!')
        end
    end
end
Go

```

```

--Kiểm tra thực thi
--xem lại các bảng
select * from Products where ProductID = '1'
select * from [Order Details] where OrderID = '10248'
--chèn dữ liệu sai
insert into [Order Details] (OrderID, ProductID,UnitPrice) values (10248, 1, 22);
--chèn dữ liệu đúng
insert into [Order Details] (OrderID, ProductID,UnitPrice) values (10248, 1, 24);
--xóa dòng dữ liệu
delete from [Order Details] where OrderID = '10248' and ProductID = '1'
--xóa trigger
drop trigger tr_CheckInsertOrderDetails
Go

```

/\*8.Viết trigger cho thao tác Insert, Update, Delete trên bảng [Order Details]. Khi có mẫu tin chèn vào bảng [Order Details] thì tự động cập nhật chiết khấu (Discount), chiết khấu được tính theo quy định sau:

Nếu số lượng (Quantity) <=5 thì chiết khấu là 0.05

Nếu số lượng từ 6 đến 10 thì chiết khấu 0.07

Nếu số lượng từ 11 đến 20 thì chiết khấu là 0.09, ngược lại thì chiết khấu là 0.\*/

```

create trigger tr_8_UpdateOrderDetails
on [Order Details]
after insert, update, delete

```

```

as
begin
    update od
    set od.Discount = case
        when od.Quantity <= 5 then 0.05
        when od.Quantity between 6 and 10 then 0.07
        when od.Quantity between 11 and 20 then 0.09
        else 0
    end
    from [Order Details] od, inserted i
    where i.OrderID = od.OrderID
end
Go

--kiểm tra thực thi
--xem bảng
select * from [Order Details] where OrderID = '10248' and ProductID = '1'
Go
--xóa
delete from [Order Details] where OrderID = '10248' and ProductID = '1'
Go
--chèn
insert into [Order Details] (OrderID, ProductID, UnitPrice, Quantity)
values (10248, 1, 24,5)
Go
--cập nhật
update od set od.Quantity = '25' from [Order Details] od where od.OrderID = '10248'
Go

```