BÀI TẬP TUẦN 7

Mục tiêu:

Sinh viên hiểu được khái niệm, vai trò của trigger trong CSDL, phân biệt được các loại trigger, hiện thực được các loại trigger trên CSDL NorthWind. SV có thể tuỳ chọn viết After trigger hoặc Instead of trigger. SV phải tự cho mẫu dữ liệu để kiểm chứng Trigger

- ***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:*
  - *Dữ liệu được **insert** hoặc **update** sẽ được chứa trong bảng **Inserted***
  - *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:*
  - ***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*
  - ***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*

1. Tạo một **Instead of trigger** thực hiện trên **view**. Thực hiện theo các bước sau:
   - 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)
)
```

```
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)
)
```

- Tạo một **view** tên **EmpDepart_view** bao gồm các field: EmployeeID, FirstName, MiddleName, LastName, DepartmentID, Name, GroupName, dựa trên 2 bảng **MEmployees** và **MDepartment.**
- 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.**

Dữ liệu test:

insert **EmpDepart_view** values(1, 'Nguyen','Hoang','Huy',
11,'Marketing','Sales')

Kết quả:

	DepartmentID	Name	groupName
1	11	Marketing	Sales

	EmployeeID	Firtname	MiddleName	LastName	DepartmentID
1	1	Nguyen	Hoang	Huy	11

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

Các bước thực hiện:

- Tạo bảng **MCustomers** và **MSalesOrders** theo cấu trúc sau:

```
create table MCustomers
 (
        CustomerID nchar(5) not null primary key,
        CustPriority int
```

```
)

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

- 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.
- Chèn dữ liệu cho bảng **MSalesOrders**, lấy dữ liệu từ bảng **Orders,**

  chỉ lấy những hóa đơn của khách hàng có trong bảng **MCustomers**.

- Viết trigger, lấy dữ liệu từ bảng inserted và deleted
- Viết câu lệnh kiểm tra việc thực thi của trigger vừa tạo bằng cách chèn thêm hoặc cập nhật một record trên bảng **MSalesOrders**

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 <=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
)

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

- Chèn dữ liệu cho bảng **MSuppliers**, lấy dữ liệu từ bảng **Suppliers, cột** NumOfProduct gán giá trị NULL, bảng **MProducts** lấy từ bảng **Products**
- Viết trigger theo yêu cầu trên và viết câu lệnh hiện thực trigger

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'.
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.
6. Viết trigger cho thao tác Insert, Update, Delete trên bảng Order Details. Khi có mẫu tin được chèn vào hoặc hiệu chỉnh hoặc xoá thì cập nhật lại cột TongTriGia trong bảng Orders với TongTriGia = Tổng tiền của Quantity * UnitPrice. Lưu ý: nếu bảng Orders chưa có cột TongTriGia thì bổ sung vào trước khi kiểm chứng trigger.
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.
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 0.1