

```

--Tạo một CSDL để test thử trigger
Create Database demo;
use demo;
Create table class(
    ID_class int primary key,
    nameClass nvarchar(30)
);
Create table Student(
    ID_student int primary key,
    fullName nvarchar(30),
    Age int,
    class_id int,
    CONSTRAINT Ma FOREIGN KEY (class_id) REFERENCES class(Id_class)
);
--Tạo một Insert trigger nhằm đảm bảo giá trị của cột tuổi khi chèn vào là luôn luôn lớn hơn
16

```

```

select * from class
select * from Student
Go

```

```

Create TRIGGER CheckAgeOnInsert
on Student
For insert
As
    Begin
        If exists (select * from inserted where age<16)
        begin
            print N'tuổi không thể nhỏ hơn 16';
            rollback transaction;
        end
    end
Go
--trigger không được gọi trực tiếp bằng truy vấn mà tạo ra khi có lệnh gọi
Insert into Student values (N'Phan Thị Lệ Thuyền',15,1,0)
Insert into Student values (N'Phan Thị Lệ Thuyền',19,1,0)
Go
-- chúng ta cũng có thể tạo Trigger cho thao tác update dữ liệu
Create TRIGGER CheckAgeOnUpdate
on Student
For update
As
    Begin
        If exists (select * from inserted I inner join deleted D
            on I.id = D.id where D.Age>I.Age)
        begin
            print N'tuổi mới không thể nhỏ hơn tuổi trước đó';
            rollback transaction;
        end
    end
Go
-- Chèn theo dữ liệu vào
Insert into Student values (N'Trần Văn Hoàng',20,1,0)
Insert into Student values (N'Lâm Vĩ Dạ',22,1,0)
-- Trigger update chỉ xảy ra khi có sự kiện update xảy ra, nên chúng ta update dữ liệu
Update Student SET Age= 18 where fullName like 'Tran Van Hoang';
Update Student SET Age= 21 where fullName like 'Trần Văn Hoàng';

```

```

Go
Create TRIGGER deleteStudent1
on Student
For Delete
As
    Begin
        declare @Id int;
        Select @Id=id from deleted;
        rollback transaction;
        update Student set Deleted=1 where id=@id;

    end
Go
-- xóa nhiều sinh viên
Alter TRIGGER deleteStudent
on Student
For Delete
As
    Begin
        update Student set Deleted=1 where id in (select ID from deleted)
        rollback transaction;
    end
go
select * from class
select * from Student
go

```

---

```

use demo
select * from class
select * from Student
go
---STORE PROCE
-- Tạo một thủ tục gồm 2 việc: chèn mới một sinh viên và update lại sinh viên tuổi 20 thành
19
create procedure In_make_student
as
begin
insert into Student (fullName, age, classID, Deleted) values ('Phan văn A', 23, 1,0);
update Student set Age = 21 where age = 20;
end
go
-- thực thi procedure
Exec In_make_student

-- Muốn sửa store prore thì thay create thành alter
Go
alter procedure up_make_student
as
begin
insert into Student (fullName, age, classID, Deleted) values ('Phan Thanh DUY', 23, 1,0);
update Student set Age = 25 where age = 21;
end
go
select * from Student

```

```

-- Tham số đầu vào
alter procedure up_make_student @productAge int
-- Tạo input cho nó
as
begin
insert into Student (fullName, age, classID, Deleted) values ('Phan Thanh DUY', 23, 1,0);
update Student set Age = 25 where age = @productAge;
end
go
-- thực thi lệnh
Exec up_make_student 23, ''

-- chúng ta cũng có thể tạo nhiều tham số khác nhau
go
alter procedure up_make_student
-- Tạo input cho nó
@productAge int,
@name varchar(30)
as
begin
insert into Student (fullName, age, classID, Deleted) values (@name, 23, 1,0);
update Student set Age = 25 where age = @productAge;
end
go
Exec up_make_student 23, 'Ho Thanh Nghi'
select * from Student

```