TRIGGER

-a store procedure

-automatically activated when inserting, deleting

or updating (delete -->insert) a row on a table.

-two considered tables: inserted, deleted

Example 1: the year old of student must be under 24

```
                Insert          Delete          Update
Student           +               -             +(Birthday)
create function GetYearOld(@birthday datetime)
returns int
as begin
     return year(getdate())-year(@birthday)
end




create trigger Example1_trigger on Student
for insert,update
as begin

     if(exists(select * from inserted I
     where dbo.getyearold(i.birthday)>24))
     begin
         raiserror('Loi....',16,1)
         rollback tran
     end

end
go
```

Example 2: BeginYear of a class must be less than its EndYear

```
            iNSERT  DELETE   UPDATE
CLASS       +         -        +(BeginYear,EndYear)
create trigger Example2_trigger on Class
for insert,update
as begin

    if(exists(select * from inserted i where
         i.beginyear > i.endyear  ))
    begin
         raiserror('Loi......',16,1)
         rollback tran
    end
end
go
```

Example 3: the credit of a subject must be less than 5

```
                 Insert   Delete   update
SUBJECT          +          -        +(credit)

create trigger Example3_trigger on subject
for insert,update
as begin
    if(exists (select * from inserted where credits>5))
    begin
         raiserror('Loi....',16,1)
         rollback tran
    end
end
go
```

Example 4: a teacher assigned teaching a subject

must have ability teaching the subject

```
         Insert Delete Update
course   +        -       +(TeacherID, SubjectID)
Ability  -        +        +(TeacherID, SubjectID)
```

```sql
create trigger Example4_1 on course
for insert,update
as begin
    if(exists( select * from inserted i where
                i.subjectID not in (select a.subjectID
                            from ability a
                where a.teacherID=i.teacherID)))
    begin
        raiserror('Loi....',16,1)
        rollback tran
    end
end
create trigger Example4_2 on Ability
for delete,update
as begin
    if(exists( select * from deleted d
                where d.subjectID in (
                        select c.subjectID
                        from course c
                        where c.teacherID=d.teacherID)))
    begin
        raiserror('Loi....',16,1)
        rollback tran
    end
end
go
```

Example 5: the number of subjects that a teacher can teach must be more than 3.

	Insert	Delete	Update
Ability	–	+	+(TeachID)

```sql
create function GetAbility(@teacherID varchar(50))
returns int
as begin
    declare @n int
    select @n=count(*)  from ability
    where teacherID=@teacherID
    return @n
end

create trigger Example5_trigger on ability
for delete, update
as begin
    if(exists( select * from deleted d where
    dbo.GetAbility(d.teacherID)<3))
    begin
        raiserror('Loi....',16,1)
```

```
            rollback tran
        end
end
go
```

Example 6: the number of subjects that a teacher can teach must be less than 3.

```
            Insert    Delete  Update
ability   +        -       + (teacherID)
create function GetAbility(@teacherID varchar(50))
returns int
as begin
    declare @n int
    select @n=count(*)  from ability
    where teacherID=@teacherID
    return @n
end

create trigger Example6_trigger on ability
for insert, update
as begin
    if(exists( select * from inserted i
            where dbo.getGetAbility(i.teacherID)>=3))
    begin
        raiserror('Loi....',16,1)
        rollback tran
    end
end
```

1: the number of the male students in a class must be less than 10

```
            Insert delete update
Student    +       -       +(ClassID)
create function getMaleNumber (@classID varchar(20))
returns int
as begin
    declare @n int
    select @n=count(*) from student where classID=@classID
    and gender='Male'
    return @n
end
create trigger exercise_trigger on student
for insert, update
as begin
    if(exists( select * from inserted i
            where dbo.getMaleNumber(i.classID)>=10))
    begin
        raiserror('Loi....',16,1)
        rollback tran
```

```
        end
end
```

2: the perfect of a class must belong to the class

```
            insert   Delete  update
Class          +         -          +(Perfect)
Student        -         +          +(ClassID)
create trigger exercise2_1_trigger on class
for insert,update
as begin
    if(exists( select * from inserted i
              where i.perfect not in (select s.ID
              from student s where s.classID=i.classID))
    begin
        raiserror('Loi....',16,1)
        rollback tran
    end
end
create trigger exercise2_1_trigger on Student
for delete, update
as begin
    if(exists( select * from deleted i
              where i.ID in (select perfect from class)


    begin
        raiserror('Loi....',16,1)
        rollback tran
    end
end
```