

TRRIGERS

ما هو Trigger؟

الـ trigger هو مجموع من أكواد برمجية يتم تنفيذها عند حدوث حدث معين.

Definition: a trigger defines an action the database should take when some database related event occurs.

يتم اضافة Trigger كالتالي

```
Create TRIGGER [TRIGGER NAME]
on [TABLE NAME]
For [INSERT | UPDATE | DELETE ]
AS
BEGIN
    --Query to execute
End
```

مثال: حفظ الاضافة والتعديل على جدول الموظفين

1- نقوم باضافة جدول الى قاعدة بيانات الشركة (Company)

```
CREATE TABLE logTbl (
    id int IDENTITY(1,1) NOT NULL,
    action nvarchar (max) NULL,
    primary key (id)
);
```

2- بعد اضافة الجدول نقوم باضافة TRIGGER كالتالي

```
Create TRIGGER tblEmployee_ForInsert
on employee
For INSERT
AS
BEGIN
    DECLARE @SSN nvarchar(9)
    DECLARE @FNAME nvarchar(20)
    SELECT @SSN=SSN,@FNAME=FNAME from inserted
    insert into logTbl
    values ('New emplyess with SSN= '+ @SSN + ' with name : '+@FNAME+' is added at
'+ cast(Getdate() as nvarchar(20)))
End
```

H.W قم بتعديل ال Trigger ليتم حفظ جميع معلومات الموظف

3- بعد ذلك نقوم باضافة Trigger لتتبع الحذف

```
CREATE TRIGGER tblEmployee_ForDelete
on employee
For Delete
AS
BEGIN
    DECLARE @SSN nvarchar(9)
    SELECT @SSN=SSN from deleted
    insert into logTbl
    values ('An emplyess with SSN= '+ @SSN + ' is Deleted at '+ cast(Getdate() as
nvarchar(20)))
END
```

H.W قم بتعديل ال Trigger الخاص بالحذف ليتم حفظ جميع معلومات الموظف قبل الحذف

-4 - ولإضافة Trigger خاص بالتحديث نضيف الكود التالي

```
CREATE trigger tblEmployee_ForUpdate
on employee
for Update
as
Begin
    -- Declare variables to hold Old_ and New updated data

    Declare @SSN nvarchar(9)
    Declare @Old_FNAME nvarchar(8), @New_FNAME nvarchar(8)
    Declare @Old_MINIT nvarchar(2), @New_MINIT nvarchar(2)
    Declare @Old_LNAME nvarchar(8), @New_LNAME nvarchar(8)
    Declare @Old_BDATE date, @New_BDATE date
    Declare @Old_ADDRESS nvarchar(27), @New_ADDRESS nvarchar(27)
    Declare @Old_SEX nvarchar(1), @New_SEX nvarchar(1)
    Declare @Old_SALARY numeric(7, 0), @New_SALARY numeric(7, 0)
    Declare @Old_SUPERSSN nvarchar(9), @New_SUPERSSN nvarchar(9)
    Declare @Old_DNO numeric(1, 0), @New_DNO numeric(1, 0)

    -- Variable to build the string
    Declare @stingToAdd nvarchar(1000)

    -- Load the updated records into temporary table
    Select *
    into #TempTbl
    from inserted

    -- Loop throw the records in temp table
    While(Exists(Select SSN from #TempTbl))
    Begin
        --Initialize the string to empty
        Set @stingToAdd = ''

        -- Select first row data from temp table
        SELECT TOP 1
            @SSN=SSN
            ,@New_FNAME=FNAME
            ,@New_MINIT=MINIT
            ,@New_LNAME=LNAME
            ,@New_BDATE=BDATE
            ,@New_ADDRESS=ADDRESS
            ,@New_SEX=SEX
            ,@New_SALARY=SALARY
            ,@New_SUPERSSN=SUPERSSN
            ,@New_DNO=DNO
        from #TempTbl

        -- Select the corresponding row from deleted table
        SELECT
            @Old_FNAME=FNAME
            ,@Old_MINIT=MINIT
            ,@Old_LNAME=LNAME
            ,@Old_BDATE=BDATE
            ,@Old_ADDRESS=ADDRESS
            ,@Old_SEX=SEX
            ,@Old_SALARY=SALARY
            ,@Old_SUPERSSN=SUPERSSN
            ,@Old_DNO=DNO
```

```

from deleted
where SSN = @SSN

    -- Build the audit string dynamically
    Set @stingToAdd = 'Employee with SSN = ' + @SSN + ' changed'
if(@Old_FNAME <> @New_FNAME)
    Set @stingToAdd = @stingToAdd + ' FNAME from ' + @Old_FNAME + ' to ' + @New_FNAME

if(@Old_MINIT <> @New_MINIT)
    Set @stingToAdd = @stingToAdd + ' ,MINIT from ' + @Old_MINIT + ' to ' + @New_MINIT

if(@Old_LNAME <> @New_LNAME)
    Set @stingToAdd = @stingToAdd + ' ,LNAME from ' + @Old_LNAME + ' to ' + @New_LNAME

if(@Old_BDATE <> @New_BDATE)
    Set @stingToAdd = @stingToAdd + ' ,BDATE from ' + Cast(@Old_BDATE as nvarchar(20))+ '
to ' + Cast(@New_BDATE as nvarchar(20))

if(@Old_ADDRESS <> @New_ADDRESS)
    Set @stingToAdd = @stingToAdd + ' ,ADDRESS from ' + @Old_ADDRESS + ' to ' + @New_ADDRESS

if(@Old_SEX <> @New_SEX)
    Set @stingToAdd = @stingToAdd + ' ,SEX from ' + @Old_SEX + ' to ' + @New_SEX

if(@Old_SALARY <> @New_SALARY)
    Set @stingToAdd = @stingToAdd + ' ,SALARY from ' + Cast(@Old_SALARY as nvarchar(10)) + '
to ' + Cast(@New_SALARY as nvarchar(10))

if(@Old_SUPERSSN <> @New_SUPERSSN)
    Set @stingToAdd = @stingToAdd + ' ,SUPERSSN from ' + @Old_SUPERSSN + ' to ' +
@New_SUPERSSN

if(@Old_DNO <> @New_DNO)
    Set @stingToAdd = @stingToAdd + ' DNO from ' + Cast(@Old_DNO as nvarchar(1)) + ' to ' +
Cast(@New_DNO as nvarchar(1))

insert into logTbl values(@stingToAdd)

    -- Delete the row from temp table, so we can move to the next row
Delete from #TempTbl where SSN = @SSN
End
End

```

5- لتجربة ال Trigger التي تم اضافتها نقوم بعمل تحديث او حذف او اضافة على قاعدة البيانات ثم نستدعي جدول ال LogTbl لمعاينة التغيرات

```

Insert into employee values('Mohammed','K','Karem','123212238','2-5-1985','44 alzahra
kut','M','25000','123456789','4')

```

```

UPDATE employee
SET FNAME = 'hadi',MINIT = 'm',LNAME = 'Mohammed',BDATE = '1-1-2000',ADDRESS = '591
atlantic st,apt b',SEX = 'M',SALARY = 23000 ,SUPERSSN = '123456789' ,DNO = 5
WHERE ssn='123212238'

```

```

Delete from employee where SSN='123212238'

```

والنتيجة ستكون كالآتي

```
10
11
12 Select * from logTbl
13
14
15
```

% <

Results Messages

id	action
19	New employess with SSN= 123212238 and name is Mohammedis added at Apr 22 2016 2:20PM
20	Employee with SSN = 333445555 changed FNAME from FRANKLIN to hadi ,MINIT from T to m ,LNAME from WONG to Mohammed ,...
21	An employess with SSN= 123212238 is Deleted at Apr 22 2016 2:20PM

Stored Procedures

الإجراءات المُخزّنة: هي جملة استعلام مخزنة في قاعدة البيانات يمكن أن تقوم بمجموعة من الأعمال، كإضافة سجل ثم تعديل البيانات في سجل آخر، والقيام بالحسابات على مجموعة أخرى من السجلات وذلك اعتباراً من أمر واحد من المستخدم. يمكن انشاء Procedures مرة واحدة ثم استدعائه اكثر من مرة وذلك لتسهيل العمل على قاعدة البيانات واصلاح الاخطاء والولوج الى قاعدة البيانات بطريقة سلسلة وبكفاءة. يمكن انشاء Procedures عن طريق الامر التالي

```
CREATE PROCEDURE <Procedure_Name>
-- Add the parameters for the stored procedure here
<@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1>,
<@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2>
AS
BEGIN
-- Insert statements for procedure here
SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
```

مثال لانشاء Procedures لعملية حذف قيد من جدول الموظفين

```
CREATE PROCEDURE DeleteFromEmployee
@SSN nvarchar(9)
AS
BEGIN
Delete from employee where SSN=@SSN
END
```

ولاستدعاء ال Procedure نكتب اليعاز التالي

```
EXEC DeleteFromEmployee '111111111';
```

مثال : لانشاء Procedure ليقوم بعملية الاضافة والتحديث والحذف :

```
CREATE procedure CreatEmployee
@Operation nvarchar(9),
@SSN nvarchar(9)
,@FNAME nvarchar(8)=null
,@MINIT nvarchar(2)=null
,@LNAME nvarchar(8)=null
,@BDATE date=null
,@ADDRESS nvarchar(27)=null
,@SEX nvarchar(1)=null
,@SALARY numeric(7, 0)=null
,@SUPERSSN nvarchar(9)=null
,@DNO numeric(1, 0)=null
AS
BEGIN
if @Operation='insert'
BEGIN
INSERT INTO employee
(FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
VALUES
(@FNAME, @MINIT, @LNAME, @SSN, @BDATE, @ADDRESS, @SEX, @SALARY, @SUPERSSN, @DNO)
```

```

    END
ELSE IF @Operation='update'
    BEGIN
        UPDATE employee
        SET FNAME = @FNAME,MINIT= @MINIT,LNAME = @LNAME,BDATE
=@BDATE,ADDRESS = @ADDRESS,SEX = @SEX,SALARY = @SALARY,SUPERSSN = @SUPERSSN ,DNO = @DNO
        WHERE SSN=@SSN
    END
ELSE IF @Operation = 'delete'
    BEGIN
        Delete from employee where SSN=@SSN
    END

ELSE IF @Operation <>'insert' OR @Operation<>'update' OR @Operation<>'delete'
    BEGIN
        Raiserror('Please chose one operation (insert, update, or delete)', 16, 1)
        Return
    END
END

```