This video is a continuation of [Part - 43](http://csharp-video-tutorials.blogspot.com/2012/09/dml-triggers-part-43.html), Please watch Part 43, before watching this video.   
  
   
  
   
  
   
  
**Triggers make use of 2 special tables**, INSERTED and DELETED. The inserted table contains the updated data and the deleted table contains the old data. The After trigger for UPDATE event, makes use of both inserted and deleted tables.   
  
**Create AFTER UPDATE trigger script:**  
Create trigger tr\_tblEmployee\_ForUpdate  
on tblEmployee  
for Update  
as  
Begin  
 Select \* from deleted  
 Select \* from inserted   
End  
  
**Now, execute this query:**  
Update tblEmployee set Name = 'Tods', Salary = 2000,   
Gender = 'Female' where Id = 4  
  
**Immediately after the UPDATE statement execution**, the AFTER UPDATE trigger gets fired, and you should see the contenets of INSERTED and DELETED tables.  
  
**The following AFTER UPDATE trigger, audits employee information upon UPDATE**, and stores the audit data in tblEmployeeAudit table.  
Alter trigger tr\_tblEmployee\_ForUpdate  
on tblEmployee  
for Update  
as  
Begin  
      -- Declare variables to hold old and updated data  
      Declare @Id int  
      Declare @OldName nvarchar(20), @NewName nvarchar(20)  
      Declare @OldSalary int, @NewSalary int  
      Declare @OldGender nvarchar(20), @NewGender nvarchar(20)  
      Declare @OldDeptId int, @NewDeptId int  
       
      -- Variable to build the audit string  
      Declare @AuditString nvarchar(1000)  
        
      -- Load the updated records into temporary table  
      Select \*  
      into #TempTable  
      from inserted  
       
      -- Loop thru the records in temp table  
      While(Exists(Select Id from #TempTable))  
      Begin  
            --Initialize the audit string to empty string  
            Set @AuditString = ''  
             
            -- Select first row data from temp table  
            Select Top 1 @Id = Id, @NewName = Name,   
            @NewGender = Gender, @NewSalary = Salary,  
            @NewDeptId = DepartmentId  
            from #TempTable  
             
            -- Select the corresponding row from deleted table  
            Select @OldName = Name, @OldGender = Gender,   
            @OldSalary = Salary, @OldDeptId = DepartmentId  
            from deleted where Id = @Id  
   
     -- Build the audit string dynamically             
            Set @AuditString = 'Employee with Id = ' + Cast(@Id as nvarchar(4)) + ' changed'  
            if(@OldName <> @NewName)  
                  Set @AuditString = @AuditString + ' NAME from ' + @OldName + ' to ' + @NewName  
                   
            if(@OldGender <> @NewGender)  
                  Set @AuditString = @AuditString + ' GENDER from ' + @OldGender + ' to '+ @NewGender  
                   
            if(@OldSalary <> @NewSalary)  
                  Set @AuditString = @AuditString + ' SALARY from ' + Cast(@OldSalary as nvarchar(10))+ ' to ' + Cast(@NewSalary as nvarchar(10))  
                    
     if(@OldDeptId <> @NewDeptId)  
                  Set @AuditString = @AuditString + ' DepartmentId from ' + Cast(@OldDeptId as nvarchar(10))+ ' to ' + Cast(@NewDeptId as nvarchar(10))  
             
            insert into tblEmployeeAudit values(@AuditString)  
              
            -- Delete the row from temp table, so we can move to the next row  
            Delete from #TempTable where Id = @Id  
      End  
End