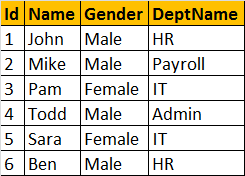
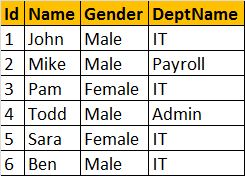
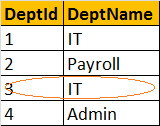
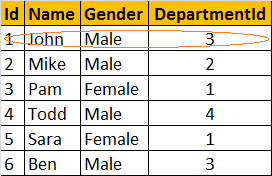
**Suggested SQL Server Videos before watching this Video**  
[Part 43 - DML triggers](http://csharp-video-tutorials.blogspot.com/2012/09/dml-triggers-part-43.html)  
[Part 44 - DML After Update Trigger](http://csharp-video-tutorials.blogspot.com/2012/09/after-update-trigger-part-44.html)  
[Part 45 - Instead of Insert Trigger](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-insert-trigger-part-45.html)   
  
   
  
   
  
   
  
**In this video we will learn about, INSTEAD OF UPDATE** trigger. An INSTEAD OF UPDATE triggers gets fired instead of an update event, on a table or a view. For example, let's say we have, an INSTEAD OF UPDATE trigger on a view or a table, and then when you try to update a row with in that view or table, instead of the UPDATE, the trigger gets fired automatically. INSTEAD OF UPDATE TRIGGERS, are of immense help, to correctly update a view, that is based on multiple tables.  
  
**Let's create the required Employee and Department tables**, that we will be using for this demo.  
  
**SQL Script to create tblEmployee table:**  
CREATE TABLE tblEmployee  
(  
  Id int Primary Key,  
  Name nvarchar(30),  
  Gender nvarchar(10),  
  DepartmentId int  
)  
  
**SQL Script to create tblDepartment table**  
CREATE TABLE tblDepartment  
(  
 DeptId int Primary Key,  
 DeptName nvarchar(20)  
)  
  
**Insert data into tblDepartment table**  
Insert into tblDepartment values (1,'IT')  
Insert into tblDepartment values (2,'Payroll')  
Insert into tblDepartment values (3,'HR')  
Insert into tblDepartment values (4,'Admin')  
  
**Insert data into tblEmployee table**  
Insert into tblEmployee values (1,'John', 'Male', 3)  
Insert into tblEmployee values (2,'Mike', 'Male', 2)  
Insert into tblEmployee values (3,'Pam', 'Female', 1)  
Insert into tblEmployee values (4,'Todd', 'Male', 4)  
Insert into tblEmployee values (5,'Sara', 'Female', 1)  
Insert into tblEmployee values (6,'Ben', 'Male', 3)  
  
**Since, we now have the required tables**, let's create a view based on these tables. The view should return Employee Id, Name, Gender and DepartmentName columns. So, the view is obviously based on multiple tables.  
**Script to create the view:**  
Create view vWEmployeeDetails  
as  
Select Id, Name, Gender, DeptName  
from tblEmployee   
join tblDepartment  
on tblEmployee.DepartmentId = tblDepartment.DeptId  
  
**When you execute, Select \* from vWEmployeeDetails**, the data from the view, should be as shown below  
   
  
**In Part 45, we tried to insert a row into the view**, and we got an error stating - 'View or function vWEmployeeDetails is not updatable because the modification affects multiple base tables.'  
  
**Now, let's try to update the view**, in such a way that, it affects, both the underlying tables, and see, if we get the same error. The following UPDATE statement changes **Name column** from **tblEmployee** and **DeptName column** from **tblDepartment**. So, when we execute this query, we get the same error.  
Update vWEmployeeDetails   
set Name = 'Johny', DeptName = 'IT'  
where Id = 1  
  
**Now, let's try to change, just the department of John from HR to IT**. The following UPDATE query, affects only one table, tblDepartment. So, the query should succeed. But, before executing the query, please note that, employees **JOHN** and **BEN** are in **HR**department.  
Update vWEmployeeDetails   
set DeptName = 'IT'  
where Id = 1  
  
**After executing the query**, select the data from the view, and notice that **BEN'sDeptName** is also changed to **IT**. We intended to just change **JOHN's DeptName**. So, the UPDATE didn't work as expected. This is because, the UPDATE query, updated the **DeptName from HR to IT**, in tblDepartment table. For the UPDATE to work correctly, we should change the **DeptId** of **JOHN** from 3 to 1.   
  
**Incorrectly Updated View**  
   
  
**Record with Id = 3, has the DeptName changed from 'HR' to 'IT'**  
   
  
**We should have actually updated, JOHN's DepartmentId from 3 to 1**  
  
  
  
**So, the conclusion is that, if a view is based on multiple tables**, and if you update the view, the UPDATE may not always work as expected. To correctly update the underlying base tables, thru a view, INSTEAD OF UPDATE TRIGGER can be used.  
  
**Before, we create the trigger, let's update the DeptName to HR for record with Id = 3.**

Update tblDepartment set DeptName = 'HR' where DeptId = 3  
  
**Script to create INSTEAD OF UPDATE trigger:**  
Create Trigger tr\_vWEmployeeDetails\_InsteadOfUpdate  
on vWEmployeeDetails  
instead of update  
as  
Begin  
 -- if EmployeeId is updated  
 if(Update(Id))  
 Begin  
 Raiserror('Id cannot be changed', 16, 1)  
 Return  
 End  
   
 -- If DeptName is updated  
 if(Update(DeptName))   
 Begin  
 Declare @DeptId int  
  
 Select @DeptId = DeptId  
 from tblDepartment  
 join inserted  
 on inserted.DeptName = tblDepartment.DeptName  
   
 if(@DeptId is NULL )  
 Begin  
 Raiserror('Invalid Department Name', 16, 1)  
 Return  
 End  
   
 Update tblEmployee set DepartmentId = @DeptId  
 from inserted  
 join tblEmployee  
 on tblEmployee.Id = inserted.id  
 End  
   
 -- If gender is updated  
 if(Update(Gender))  
 Begin  
 Update tblEmployee set Gender = inserted.Gender  
 from inserted  
 join tblEmployee  
 on tblEmployee.Id = inserted.id  
 End  
   
 -- If Name is updated  
 if(Update(Name))  
 Begin  
 Update tblEmployee set Name = inserted.Name  
 from inserted  
 join tblEmployee  
 on tblEmployee.Id = inserted.id  
 End  
End  
  
**Now, let's try to update JOHN's Department to IT.**  
Update vWEmployeeDetails   
set DeptName = 'IT'  
where Id = 1  
  
**The UPDATE query works as expected.** The INSTEAD OF UPDATE trigger, correctly updates, JOHN's DepartmentId to 1, in tblEmployee table.  
  
**Now, let's try to update Name, Gender and DeptName.** The UPDATE query, works as expected, without raising the error - 'View or function vWEmployeeDetails is not updatable because the modification affects multiple base tables.'  
Update vWEmployeeDetails   
set Name = 'Johny', Gender = 'Female', DeptName = 'IT'   
where Id = 1  
  
**Update**() function used in the trigger, returns true, even if you update with the same value. For this reason, I recomend to compare values between inserted and deleted tables, rather than relying on Update() function. The Update() function does not operate on a per row basis, but across all rows.