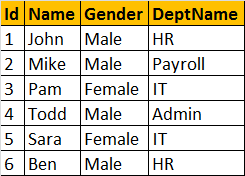
**Suggested SQL Server Videos before watching this Video**  
[Part 45 - Instead of Insert Trigger](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-insert-trigger-part-45.html)  
[Part 46 - Instead of Update Trigger](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-update-triggers-part-46.html)   
  
   
  
   
  
   
  
**In this video we will learn about, INSTEAD OF DELETE trigger**. An INSTEAD OF DELETE trigger gets fired instead of the DELETE event, on a table or a view. For example, let's say we have, an INSTEAD OF DELETE trigger on a view or a table, and then when you try to update a row from that view or table, instead of the actual DELETE event, the trigger gets fired automatically. INSTEAD OF DELETE TRIGGERS, are used, to delete records from 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**](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-insert-trigger-part-45.html), 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'. Along, the same lines, in [**Part 46**](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-update-triggers-part-46.html), when we tried to update a view that is based on multiple tables, we got the same error. To get the error, the UPDATE should affect both the base tables. If the update affects only one base table, we don't get the error, but the UPDATE does not work correctly, if the **DeptName**column is updated.  
  
**Now, let's try to delete a row from the view, and we get the same error.**  
Delete from vWEmployeeDetails where Id = 1  
  
**Script to create INSTEAD OF DELETE trigger:**  
Create Trigger tr\_vWEmployeeDetails\_InsteadOfDelete  
on vWEmployeeDetails  
instead of delete  
as  
Begin  
 Delete tblEmployee   
 from tblEmployee  
 join deleted  
 on tblEmployee.Id = deleted.Id  
   
 --Subquery  
 --Delete from tblEmployee   
 --where Id in (Select Id from deleted)  
End  
  
**Notice that, the trigger tr\_vWEmployeeDetails\_InsteadOfDelete**, makes use of DELETED table. DELETED table contains all the rows, that we tried to DELETE from the view. So, we are joining the DELETED table with tblEmployee, to delete the rows. You can also use sub-queries to do the same. In most cases JOINs are faster than SUB-QUERIEs. However, in cases, where you only need a subset of records from a table that you are joining with, sub-queries can be faster.  
  
**Upon executing the following DELETE statement**, the row gets DELETED as expected from tblEmployee table  
Delete from vWEmployeeDetails where Id = 1 

|  |  |
| --- | --- |
| **Trigger** | **INSERTED or DELETED?** |
| Instead of Insert | DELETED table is always empty and the INSERTED table contains the newly inserted data. |
| Instead of Delete | INSERTED table is always empty and the DELETED table contains the rows deleted |
| Instead of Update | DELETED table contains OLD data (before update), and inserted table contains NEW data(Updated data) |