

**1) Write an example for INSTEAD OF TRIGGERS for each category (INSERT, UPDATE, DELETE). Do some meaning actions when it is triggers? Like updating the audit table, setting up the updated time (Do not just display something from database).**

In a database, a trigger is a set of Structured Query Language (SQL) statements that automatically "fires off" an action when a specific operation, such as changing data in a table, occurs. A trigger consists of an event (an INSERT, DELETE, or UPDATE statement issued against an associated table) and an action (the related procedure). Triggers are used to preserve data integrity by checking on or changing data in a consistent manner.

## **Instead of Insert**

```
CREATE TABLE Student
(
  Id int Primary Key,
  Name nvarchar(30),
  Gender nvarchar(10),
  SectionId int
);
```

```
CREATE TABLE Section
(
  SecId int Primary Key,
  SecName nvarchar(20)
);
```

```
Insert into Section(SecId,SecName) values (1,'Sec1')
Insert into Section(SecId,SecName) values (2,'Sec2')
Insert into Section(SecId,SecName) values (3,'Sec3')
Insert into Section(SecId,SecName) values (4,'Sec4')
```

```
Insert into Student(Id,Name,Gender,SectionId) values (1,'Shashank1', 'Male', 3)
Insert into Student(Id,Name,Gender,SectionId) values (2,'Shashank2', 'Male', 2)
Insert into Student(Id,Name,Gender,SectionId) values (3,'Shashank3', 'Male', 1)
Insert into Student(Id,Name,Gender,SectionId) values (4,'Shashank4', 'Male', 4)
Insert into Student(Id,Name,Gender,SectionId) values (5,'Shashank5', 'Male', 1)
Insert into Student(Id,Name,Gender,SectionId) values (6,'Shashank6', 'Male', 3)
```

```
Create view ViewStudentDetails
```

```

as
Select Id, Name, Gender, SecName
from Student
join Section
on Student.SectionId = Section.SecId
Create trigger TR_ViewStudentDetails_InsteadOfInsert
on ViewStudentDetails
Instead Of Insert
as
Begin
Declare @SecId int

Select @SecId = SecId
from Section
join inserted
on inserted.SecName = Section.SecName

Insert into Student(Id, Name, Gender, SectionId)
Select Id, Name, Gender, @SecId
from inserted
End

```

Instead of Insert trigger is use whenever the foreign key constraint of one table may get violated because of entry in any other table.

So, in instead of Insert trigger we first store the value in a declared variable and then update the value simultaneously in both tables which helps in not violating the constraint..

## **Instead of Update**

The code for table creation and inserting values is the same.

### **Syntax for Instead of Update**

```

CREATE TRIGGER InsteadOfUPDATETriggerExample
on
viewStudentDetails
INSTEAD OF UPDATE

BEGIN

```

```
-- variable declaration
-- logic for update trigger
END
```

Instead Of Update Trigger in SQL Server is usually used to correctly update the records from a view that is based on multiple tables.

## **Instead of Delete**

```
CREATE TRIGGER InsteadOfDELETETriggerExample
on
viewStudentDetails
INSTEAD OF DELETE

BEGIN
-- variable declaration
-- logic for delete trigger
END
```

Instead Of Delete Trigger in SQL Server is usually used to correctly delete the records from a view that is based on multiple tables and foreign key constraint is getting violated.

Basically triggers are methods that are used to insert, update or delete the records from tables when the referential integrity is being violated. In the trigger the developer has to provide the appropriate actions that should be taken when an event or so fires.