1. Create a trigger which doesnt allow sal update if current sal is more than proposed

ALTER TRIGGER trgInsteadOfUpdate ON emp

INSTEAD OF UPDATE

AS

DECLARE @empno int;

DECLARE @sal int;

DECLARE @cursal int;

SELECT @cursal=i.sal FROM deleted i

SELECT @sal=i.sal FROM inserted i;

SELECT @empno=i.empno FROM inserted i;

BEGIN

IF(@sal<@cursal)

BEGIN

RAISERROR ('Cannot update salary since it is less than current salary',16,1);

END

ELSE

BEGIN

UPDATE emp

SET sal=@sal

WHERE empno=@empno

PRINT 'Record Updated-Instead of Update Trigger'

END

END

GO

UPDATE emp

SET sal=700

WHERE empno=7369

2. Create a trigger which does not allow delete on emp on Tuesday.

CREATE TRIGGER trgInsteadOfDeleteOnTuesday ON emp

INSTEAD OF DELETE

AS

DECLARE @empno int

DECLARE @day VARCHAR(20)

SET @day=DATENAME(WEEKDAY,GETDATE())

SELECT @empno=d.empno from deleted d;

if(@day='Tuesday')

BEGIN

RAISERROR('Cannot delete on Tuesday',16,1);

END

ELSE

BEGIN

DELETE FROM emp

WHERE empno=@empno;

PRINT 'Record Deleted-Instead of Delete Trigger'

END

GO

DELETE From emp

WHERE empno=2979

3. Insert,Delete and Update trigger combined

ALTER TRIGGER trgAfterInsert ON Employee\_Test

FOR INSERT,DELETE,UPDATE

AS

DECLARE @empid int;

DECLARE @empname varchar(100);

DECLARE @empsal decimal(10,2);

DECLARE @audit\_action varchar(100);

DECLARE @insertcount int;

DECLARE @deletecount int;

SELECT @empid=i.Emp\_ID FROM inserted i;

SELECT @empname=i.Emp\_Name FROM inserted i;

SELECT @empsal=i.Emp\_Sal FROM inserted i;

SELECT @insertcount=COUNT(\*) FROM inserted i;

SELECT @deletecount=COUNT(\*) FROM deleted i;

IF (@insertcount=1 AND @deletecount=0)

BEGIN

SELECT @empid=i.Emp\_ID FROM inserted i;

SELECT @empname=i.Emp\_Name FROM inserted i;

SELECT @empsal=i.Emp\_Sal FROM inserted i;

SET @audit\_action='Inserted Record--After Insert Trigger';

END

ELSE IF (@deletecount=1 AND @insertcount=0)

BEGIN

SELECT @empid=i.Emp\_ID FROM deleted i;

SELECT @empname=i.Emp\_Name FROM deleted i;

SELECT @empsal=i.Emp\_Sal FROM deleted i;

SET @audit\_action='Deleted Record--After Delete Trigger';

END

ELSE

BEGIN

SELECT @empid=i.Emp\_ID FROM inserted i;

SELECT @empname=i.Emp\_Name FROM inserted i;

SELECT @empsal=i.Emp\_Sal FROM inserted i;

SET @audit\_action='Updated Record--After Update Trigger';

END

INSERT INTO Employee\_Test\_Audit

VALUES(@empid,@empname,@empsal,@audit\_action,getdate());

IF (@insertcount=1 AND @deletecount=0)

PRINT 'AFTER INSERT trigger fired'

ELSE IF (@deletecount=1 AND @insertcount=0)

PRINT 'AFTER DELETE trigger fired'

ELSE

PRINT 'AFTER UPDATE trigger fired'

GO

INSERT INTO Employee\_Test VALUES('Levy',5000);

INSERT INTO Employee\_Test VALUES('Levy1',5000);

DELETE FROM Employee\_Test

WHERE Emp\_ID=13

UPDATE Employee\_Test

SET Emp\_name='LEVY'

WHERE Emp\_ID=12

4.Transaction to change the location

CREATE PROCEDURE changeLoc

AS

BEGIN

BEGIN TRANSACTION

UPDATE dept SET loc='Bangalore' WHERE deptno=10

COMMIT TRANSACTION

End

EXEC changeLoc

5. Drop a trigger

DROP TRIGGER trgInsteadOfUpdate

6. Disable and Enable Trigger

DISABLE TRIGGER trgAfterInsert on Employee\_Test

ENABLE TRIGGER trgAfterInsert on Employee\_Test

7. NOCOUNT

SET NOCOUNT ON is a SET function which stops the number of rows affected message in a query.It reduces the network traffic hence improving the performance.

SET COUNT ON-Count will not be returned

SET COUNT OFF-Returns the count.