**TASK #01: Create**

1. **After Insert**
2. **After Update**
3. **After Delete**

**Triggers on Shippers Table in Northwind Database. Record query and their respective results in output.**

**Audit table:**

create table Shippers\_Audit(

ShipperID int not null,

CompanyName nvarchar(40) not null,

Phone nvarchar(24),

ActionPerformed nvarchar(20))

**Queries:**

**Insert:**

create trigger tgrAfterInsert on Shippers after insert

as declare @id int, @cname varchar(40),@phone varchar(24),@action varchar(50);

select @id=i.ShipperID from inserted i;

select @cname=i.CompanyName from inserted i;

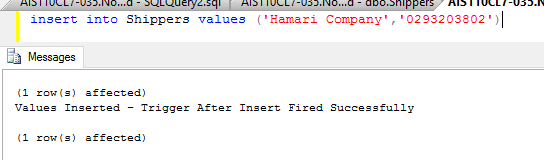
select @phone=i.Phone from inserted i;

set @action='Insert';

insert into Shippers\_Audit values(@id,@cname,@phone,@action);

print 'Values Inserted - Trigger After Insert Fired Successfully'

**Output:**



**Update:**

create trigger tgrAfterUpdate on Shippers after update

as declare @id int, @cname varchar(40),@phone varchar(24),@action varchar(50);

select @id=i.ShipperID from inserted i;

select @cname=i.CompanyName from inserted i;

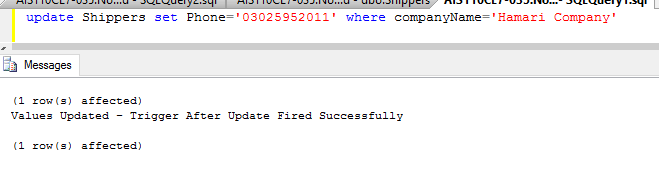
select @phone=i.Phone from inserted i;

set @action='Update';

insert into Shippers\_Audit values(@id,@cname,@phone,@action);

print 'Values Updated - Trigger After Update Fired Successfully'

**Output:**



**Delete:**

create trigger tgrAfterDelete on Shippers after delete

as declare @id int, @cname varchar(40),@phone varchar(24),@action varchar(50);

select @id=i.ShipperID from deleted i;

select @cname=i.CompanyName from deleted i;

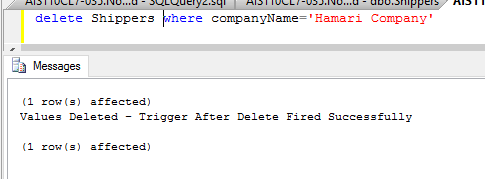
select @phone=i.Phone from deleted i;

set @action='Delete';

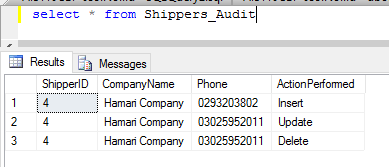
insert into Shippers\_Audit values(@id,@cname,@phone,@action);

print 'Values Deleted - Trigger After Delete Fired Successfully'

**Output:**



**Shippers Audit Table:**



**Task #02:**

**Create**

**1 Instead of Insert**

**2 Instead of Update**

**3 Instead of Delete**

**Triggers on Orders Table in Northwind Database. Record query and their respective results in output.**

**Audit Table:**

create table Orders\_Audit(

[OrderID] int NOT NULL,

[CustomerID] nchar(5) NULL,

[EmployeeID] int NULL,

[OrderDate] datetime NULL,

[RequiredDate] datetime NULL,

[ShippedDate] datetime NULL,

[ShipVia] int NULL,

[Freight] money NULL,

[ShipName] nvarchar(40) NULL,

[ShipAddress] nvarchar(60) NULL,

[ShipCity] nvarchar(15) NULL,

[ShipRegion] nvarchar(15) NULL,

[ShipPostalCode] nvarchar(10) NULL,

[ShipCountry] nvarchar(15) NULL,

ActionPerformed nvarchar(50)

)

**Insert:**

**Query:**

create trigger tgrInsteadOfInsert on Orders after insert as declare @orderid int,@customerid nchar(5),@employeeid int,@shipvia int,

@freight money, @shipname varchar(40),@action varchar(50);

select @orderid=i.OrderID from inserted i;

select @customerid=i.CustomerID from inserted i;

select @employeeid=i.EmployeeID from inserted i;

select @shipvia=i.ShipVia from inserted i;

select @freight=i.Freight from inserted i;

select @shipname=i.ShipName from inserted i;

set @action='Insert';

begin transaction

if @employeeid > 8

begin

PRINT 'You have entered incorrect employeeid. Please insert the correct information';

Rollback;

end

else

BEGIN

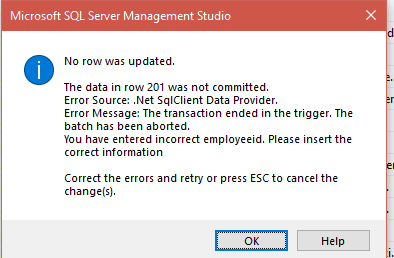
insert into Orders\_Audit values(@orderid,@customerid,@employeeid,@shipvia,@freight,@shipname,@action);

commit;

print 'Values Inserted - Trigger Instead of Insert Fired Successfully';

END;

**Output:**



**Update:**

**Query:**

create trigger tgrInsteadOfUpdate on Orders after update as declare @orderid int,@customerid nchar(5),@employeeid int,@shipvia int,

@freight money, @shipname varchar(40),@action varchar(50);

select @orderid=i.OrderID from inserted i;

select @customerid=i.CustomerID from inserted i;

select @employeeid=i.EmployeeID from inserted i;

select @shipvia=i.ShipVia from inserted i;

select @freight=i.Freight from inserted i;

select @shipname=i.ShipName from inserted i;

set @action='Update - Instead of Trigger';

begin transaction

if @freight < 50

begin

PRINT 'Too low freight. Data cannot be updated';

Rollback;

end

else

BEGIN

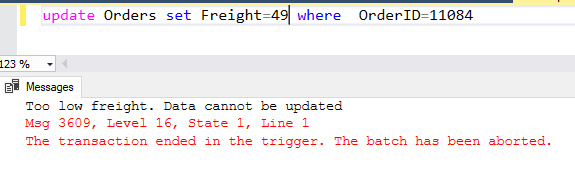
insert into Orders\_Audit values(@orderid,@customerid,@employeeid,@shipvia,@freight,@shipname,@action);

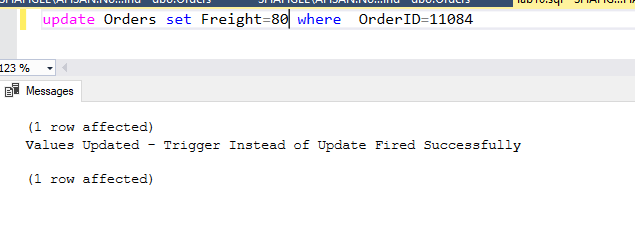
commit;

print 'Values Inserted - Trigger Instead of Insert Fired Successfully';

END;

**Output:**





**Delete:**

**Query:**

create trigger tgrInsteadOfDelete on Orders after delete as declare @orderid int,@customerid nchar(5),@employeeid int,@shipvia int,

@freight money, @shipname varchar(40),@action varchar(50);

select @orderid=i.OrderID from deleted i;

select @customerid=i.CustomerID from deleted i;

select @employeeid=i.EmployeeID from deleted i;

select @shipvia=i.ShipVia from deleted i;

select @freight=i.Freight from deleted i;

select @shipname=i.ShipName from deleted i;

set @action='Delete - Instead of Trigger';

begin transaction

if @freight > 70

begin

PRINT 'Too high freight. Data cannot be deleted';

Rollback;

end

else

BEGIN

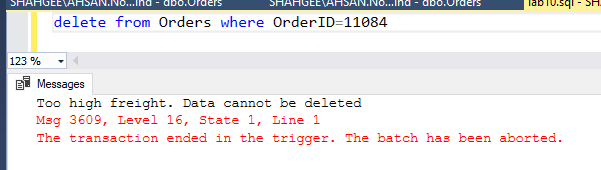
insert into Orders\_Audit values(@orderid,@customerid,@employeeid,@shipvia,@freight,@shipname,@action);

commit;

print 'Values Deleted - Trigger Instead of Delete Fired Successfully';

END;

**Output:**



**Orders Audit Table:**

