Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**11**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Create CRUD application using Webform in ASP.Net for Employees table as discussed in class. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

Date: 28/06/2022

**Task No. 1: Create After Insert , After Update , After Delete**

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

**Solution:**

**Audit Table: -Graphical user interface, application

Description automatically generated**

**Triggers:-**

create TRIGGER [shipperInsertTrigger]

ON [Shippers]

for INSERT

AS

declare @shipperid int,@companyName nvarchar(40),@phone nvarchar(24),@audit\_action

nvarchar(100)

BEGIN

-- Insert statements for trigger here

select @shipperid = i.shipperid from inserted i

select @companyName = i.companyName from inserted i

select @phone = i.phone from inserted i

set @audit\_action = 'Inserted Record -- After insert trigger'

insert into Shippers\_Audit values

(@shipperid,@companyName,@phone,@audit\_action,GETDATE())

print 'After insert trigger on shippers table fired'

END

create TRIGGER [dbo].[tgrAfterUpdate]

ON [dbo].[Shippers]

FOR UPDATE

AS

declare @shipperid int,@companyName nvarchar(40),@phone nvarchar(24),@auditAction

nvarchar(100)

BEGIN

-- Insert statements for trigger here

select @shipperid = i.shipperid from inserted i

select @companyName = i.companyName from inserted i

select @phone = i.phone from inserted i

select @auditAction = 'Updated Record -- After Update Trigger'

insert into Shippers\_Audit values

(@shipperid,@companyName,@phone,@auditAction,GETDATE())

print 'After update trigger on shipper table fired'

END

CREATE TRIGGER shipperDeleteTrigger

ON Shippers

FOR DELETE

AS

declare @shipperid int,@companyName nvarchar(40),@phone nvarchar(24),@audit\_action

nvarchar(100)

BEGIN

-- Insert statements for trigger here

select @shipperid = d.shipperid from deleted d

select @companyName = d.companyName from deleted d

select @phone = d.phone from deleted d

set @audit\_action = 'Deleted Record -- After delete trigger'

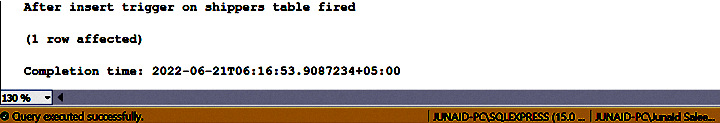
insert into Shippers\_Audit values

(@shipperid,@companyName,@phone,@audit\_action,GETDATE())

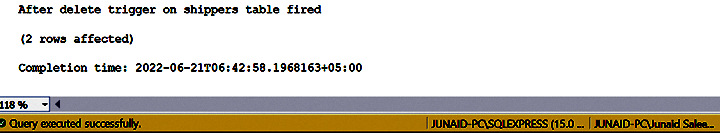
print 'After delete trigger on shippers table fired'

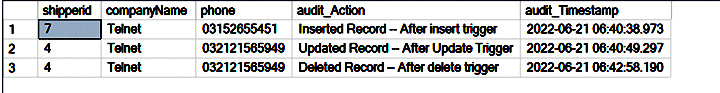
END

**Output:**

insert into Shippers values ('Telnet','03152655451')

update shippers set Phone = '032121565949' where CompanyName = 'Telnet'

delete Shippers where CompanyName = 'Telnet'

**Audit Table: -**

**Task No. 2: Create Instead of Insert , Instead of Update , Instead of Delete**

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

**Solution:**

**Audit Table: -**

create table OrdersAudit( OrderID int not null, CustomerID nchar(5), EmployeeID int, OrderDate datetime,

RequiredDate datetime, ShippedDate datetime, ShipVia int , Freight money, ShipName nvarchar(40),

ShipAddress nvarchar(60), ShipCity nvarchar(20), ShipRegion nvarchar(20), ShipPostalCode nvarchar(10),

ShipCountry nvarchar(20), ActionPerformed nvarchar(50), ActionTimestamp datetime )

**Triggers: -**

CREATE TRIGGER ordersInsteadOfInsertTrigger

ON orders

INSTEAD OF INSERT AS

DECLARE

@ord\_id int, @cust\_id nchar(5), @emp\_id int, @ord\_date datetime, @req\_date datetime, @ship\_date datetime,

@ship\_via int, @freight money, @ship\_name nvarchar(40), @ship\_add nvarchar(60), @ship\_city nvarchar(15),

@ship\_region nvarchar(15), @ship\_post\_code nvarchar(10), @ship\_count nvarchar(15), @audit\_action nvarchar(100)

select @ord\_id = i.OrderID from inserted i

select @cust\_id = i.CustomerID from inserted i

select @emp\_id = i.EmployeeID from inserted i

select @ord\_date = i.OrderDate from inserted i

select @req\_date = i.RequiredDate from inserted i

select @ship\_date = i.ShippedDate from inserted i

select @ship\_via = i.ShipVia from inserted i

select @freight = i.Freight from inserted i

select @ship\_name = i.ShipName from inserted i

select @ship\_add = i.ShipAddress from inserted i

select @ship\_city = i.ShipCity from inserted i

select @ship\_region = i.ShipRegion from inserted i

select @ship\_post\_code = i.ShipPostalCode from inserted i

select @audit\_action = 'Insert'

BEGIN

-- Insert statements for trigger here

BEGIN TRANSACTION

if (@emp\_id > 8)

begin

print 'Cannot insert data because it is not fulfilling the requirement of employee

id'

rollback

end

else

begin

insert into OrdersAudit values

(@ord\_id,@cust\_id,@emp\_id,@ord\_date,@req\_date,@ship\_date,@ship\_via,@freight,@ship\_name,@ship\_add,@ship\_city,@ship\_region,@ship\_post\_code,@ship\_count,@audit\_action,GETDATE())

commit

end

END

go

CREATE TRIGGER ordersInsteadOfUpdateTrigger

ON orders

INSTEAD OF UPDATE

AS

DECLARE

@ord\_id int, @cust\_id nchar(5), @emp\_id int, @ord\_date datetime, @req\_date datetime, @ship\_date datetime,

@ship\_via int, @freight money, @ship\_name nvarchar(40), @ship\_add nvarchar(60), @ship\_city nvarchar(15),

@ship\_region nvarchar(15), @ship\_post\_code nvarchar(10), @ship\_count nvarchar(15), @audit\_action nvarchar(100)

select @ord\_id = i.OrderID from inserted i

select @cust\_id = i.CustomerID from inserted i

select @emp\_id = i.EmployeeID from inserted i

select @ord\_date = i.OrderDate from inserted i

select @req\_date = i.RequiredDate from inserted i

select @ship\_date = i.ShippedDate from inserted i

select @ship\_via = i.ShipVia from inserted i

select @freight = i.Freight from inserted i

select @ship\_name = i.ShipName from inserted i

select @ship\_add = i.ShipAddress from inserted i

select @ship\_city = i.ShipCity from inserted i

select @ship\_region = i.ShipRegion from inserted i

select @ship\_post\_code = i.ShipPostalCode from inserted i

select @ship\_count = i.ShipCountry from inserted i

select @audit\_action = 'Update'

BEGIN

-- Insert statements for trigger here

BEGIN TRANSACTION

if (@freight < 50)

begin

print 'Cannot update data because it is not fulfilling the requirement of freight'

rollback

end

else

begin

insert into OrdersAudit values

(@ord\_id,@cust\_id,@emp\_id,@ord\_date,@req\_date,@ship\_date,@ship\_via,@freight,@ship\_name,@ship\_add,@ship\_city,@ship\_region,@ship\_post\_code,@ship\_count,@audit\_action,GETDATE())

commit

end

END

go

CREATE TRIGGER ordersInsteadOfDeleteTrigger

ON orders

INSTEAD OF DELETE

AS

DECLARE

@ord\_id int, @cust\_id nchar(5), @emp\_id int, @ord\_date datetime, @req\_date datetime, @ship\_date datetime,

@ship\_via int, @freight money, @ship\_name nvarchar(40), @ship\_add nvarchar(60), @ship\_city nvarchar(15),

@ship\_region nvarchar(15), @ship\_post\_code nvarchar(10), @ship\_count nvarchar(15), @audit\_action nvarchar(100)

select @ord\_id = d.OrderID from deleted d

select @cust\_id = d.CustomerID from deleted d

select @emp\_id = d.EmployeeID from deleted d

select @ord\_date = d.OrderDate from deleted d

select @req\_date = d.RequiredDate from deleted d

select @ship\_date = d.ShippedDate from deleted d

select @ship\_via = d.ShipVia from deleted d

select @freight = d.Freight from deleted d

select @ship\_name = d.ShipName from deleted d

select @ship\_add = d.ShipAddress from deleted d

select @ship\_city = d.ShipCity from deleted d

select @ship\_region = d.ShipRegion from deleted d

select @ship\_post\_code = d.ShipPostalCode from deleted d

select @ship\_count = d.ShipCountry from deleted d

select @audit\_action = 'Delete'

BEGIN

-- Insert statements for trigger here

BEGIN TRANSACTION

if (@freight > 70)

begin

print 'Cannot Delete data because it is not fulfilling the requirement of freight'

rollback

end

else

begin

insert into OrdersAudit values

(@ord\_id,@cust\_id,@emp\_id,@ord\_date,@req\_date,@ship\_date,@ship\_via,@freight,@ship\_name,@ship\_add,@ship\_city,@ship\_region,@ship\_post\_code,@ship\_count,@audit\_action,GETDATE())

commit

end

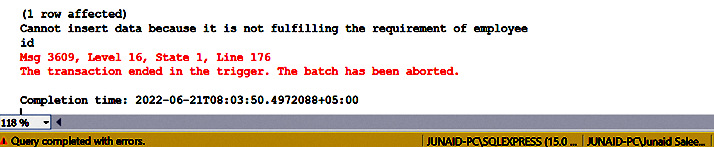
END

**Output:**

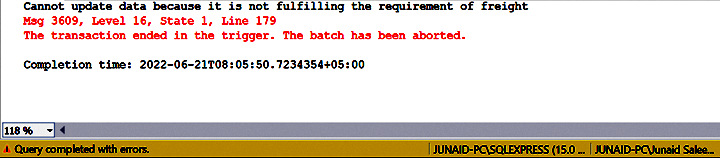
insert into Orders values

('ZING',8,GETDATE(),GETDATE(),GETDATE(),1,79.00,'Telnet','','','','','')

insert into Orders values

('SOAP',9,GETDATE(),GETDATE(),GETDATE(),1,61.00,'Telnet','','','','','')

update orders set freight = 90 where orderid = 10248



delete from orders where orderid = 10255

