**Triggers**

**Insert**

create trigger TR\_InsertAfter\_For\_Shipper on [Shippers]

after insert

as begin

declare @id int,@companyname nvarchar(50),@phone nvarchar(50)

select @id=[ShipperID],@companyname=[CompanyName],@phone=[Phone] from inserted

insert into Shippers\_Audit values(@id,@companyname,@phone,'Insert',GETDATE(),

cast(SUSER\_SNAME() as varchar(50))+' Inserted The Value In Shipper Table' )

print ('Trigger Fire Successfully For Insert Operation')

end

**Update**

create trigger TR\_InsertUpdate\_For\_Shipper on [Shippers]

after update

as begin

declare @id int,@companyname nvarchar(50),@phone nvarchar(50),

@companyname1 nvarchar(50),@phone1 nvarchar(50)

select @id=[ShipperID],@companyname=[CompanyName],@phone=[Phone] from inserted

select @companyname1=[CompanyName],@phone1=[Phone] from deleted

insert into Shippers\_Audit values(@id,@companyname,@phone,'Update',GETDATE(),

cast(SUSER\_SNAME() as varchar(50))+' Update The Value '+cast(@companyname1 as varchar(50))

+' to '+cast(@companyname as varchar(50)) +' And '+@phone1 +' To '+

@phone+' In Shipper Table' )

print ('Trigger Fire Successfully For Upate Operation')

end

**Delete**

create trigger TR\_InsertDelete\_For\_Shipper on [Shippers]

after delete

as begin

declare @id int,@companyname nvarchar(50),@phone nvarchar(50)

select @id=[ShipperID],@companyname=[CompanyName],@phone=[Phone] from deleted

insert into Shippers\_Audit values(@id,@companyname,@phone,'Delete',GETDATE(),

cast(SUSER\_SNAME() as varchar(50))+' Delete The Row Where ID = '+cast(@id as varchar(10))+' In Shipper Table' )

print ('Trigger Fire Successfully For Delete Operation')

end

**Instead of Insert**

create trigger TR\_InsteadOfInsert on Orders

instead of insert

as begin

declare @orderid int,@customerid nchar(5),@employeeid int,@shipvia int,

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

select @orderid=[OrderID],@customerid=[CustomerID],@employeeid=[EmployeeID],

@shipvia=[ShipVia],@freight=[Freight],@shipname=[ShipName] from inserted ;

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

end

**Instead of Update**

select @orderid=[OrderID],@customerid=[CustomerID],@employeeid=[EmployeeID],

@shipvia=[ShipVia],@freight=[Freight],@shipname=[ShipName] from inserted ;

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

end

**Instead of Delete**

create trigger TR\_InsteadOfDelete on Orders

instead of delete

as begin

declare @orderid int,@customerid nchar(5),@employeeid int,@shipvia int,

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

select @orderid=[OrderID],@customerid=[CustomerID],@employeeid=[EmployeeID],

@shipvia=[ShipVia],@freight=[Freight],@shipname=[ShipName] from deleted ;

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

end

**DATABASE CRUD**

SqlConnection con = new SqlConnection(cs);

string query = "ManageData @name,@age,@pic,@gender,@contact,@id,@operation";

SqlCommand cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@name", txtname.Text);

cmd.Parameters.AddWithValue("@age", txtage.Text);

cmd.Parameters.AddWithValue("@gender", cmbgender.Text);

cmd.Parameters.AddWithValue("@contact", txtcontact.Text);

cmd.Parameters.AddWithValue("@pic", SavePhoto());

cmd.Parameters.AddWithValue("@operation", "Insert");

cmd.Parameters.AddWithValue("@id", Convert.ToInt32(txtID.Text));

con.Open();

int a = cmd.ExecuteNonQuery();

if (a > 0)

{

MessageBox.Show("Inserted Data Successfully", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);

BindGridView();

clearControls();

}

else

{

MessageBox.Show("Insertion Failed", "Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

**DATA GRID VIEW**

string query = "select \* from EmpDetails ";

SqlDataAdapter sda = new SqlDataAdapter(query, con);

DataTable data = new DataTable();

sda.Fill(data);

dataGridView1.DataSource = data;

DataGridViewImageColumn dgv = new DataGridViewImageColumn();

dgv = (DataGridViewImageColumn)dataGridView1.Columns[5];

dgv.ImageLayout = DataGridViewImageCellLayout.Stretch;

dataGridView1.RowTemplate.Height = 50; con.Close();

string cs = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

**Stored procedure**

CREATE PROCEDURE ManageData

@name varchar(255),

@age int,

@image image,

@gender varchar(50),

@contact int,

@id int,

@operation varchar(10)

AS

BEGIN

IF @operation = 'INSERT'

BEGIN

INSERT INTO EmpDetails (name, age, gender, contact,picture)

VALUES (@name, @age, @gender, @contact,@image)

END

ELSE IF @operation = 'UPDATE'

BEGIN

UPDATE EmpDetails

SET name = @name, age = @age, gender = @gender, contact = @contact,picture=@image

WHERE ID = @id

END

ELSE IF @operation = 'DELETE'

BEGIN

DELETE FROM EmpDetails

WHERE ID = @id

END

END