**DATABASE**

Creation:

create database AutomobileShop;

use AutomobileShop

create table CarDetails(

CarID int primary key identity(1,1),

Manufacturer nvarchar(50) not null,

ModelName nvarchar(50) not null,

YearMade nvarchar(50) not null,

CarType nvarchar(50) not null,

CarColor nvarchar(50) not null

)

Stored Procedures:

--get all rows from CarDetails

create procedure GetAll

as

begin

select \* from CarDetails

end

go

--get one row from CarDetails

create procedure GetInfo

@id int

as

begin

select \* from CarDetails where CarID = @id

end

go

--insert into CarDetails

create procedure AddInfo

@manufacturer nvarchar(50),

@model nvarchar(50),

@year nvarchar(50),

@type nvarchar(50),

@color nvarchar(50)

as

begin

insert into CarDetails

values (@manufacturer, @model, @year, @type, @color)

end

go

--delete from CarDetails

create procedure DeleteInfo

@id int

as

begin

delete from CarDetails where CarID = @id

end

go

-- update in CarDetails

create procedure UpdateInfo

@id int,

@manufacturer nvarchar(50),

@model nvarchar(50),

@year nvarchar(50),

@type nvarchar(50),

@color nvarchar(50)

as

begin

update CarDetails

set Manufacturer = @manufacturer, ModelName = @model, YearMade = @year, CarType = @type, CarColor = @color

where CarID = @id

end

go

--get all rows in CarDetails\_Audit

create procedure GetAudit

as

begin

select \* from CarDetails\_Audit

end

go

Triggers:

create table CarDetails\_Audit(

CarID int,

Manufacturer nvarchar(50) not null,

ModelName nvarchar(50) not null,

YearMade nvarchar(50) not null,

CarType nvarchar(50) not null,

CarColor nvarchar(50) not null,

AuditAction varchar(100),

AuditTimeStamp datetime

)

--after insert

create trigger trrgAfterInsert on CarDetails

for insert

as

declare @CarID int;

declare @Manufacturer nvarchar(50);

declare @ModelName nvarchar(50);

declare @YearMade nvarchar(50);

declare @CarType nvarchar(50);

declare @CarColor nvarchar(50);

declare @AuditAction varchar(100);

select @CarID = i.CarID from inserted i;

select @Manufacturer = i.Manufacturer from inserted i;

select @ModelName = i.ModelName from inserted i;

select @YearMade = i.YearMade from inserted i;

select @CarType = i.CarType from inserted i;

select @CarColor = i.CarColor from inserted i;

set @AuditAction = 'Inserted Record - after insert trigger';

insert into CarDetails\_Audit

(CarID, Manufacturer, ModelName, YearMade, CarType, CarColor, AuditAction, AuditTimeStamp)

values (@CarID, @Manufacturer, @ModelName, @YearMade, @CarType, @CarColor, @AuditAction, getdate());

print 'after insert trigger fired'

go

--after update

create trigger trrgAfterUpdate on CarDetails

for update

as

declare @CarID int;

declare @Manufacturer nvarchar(50);

declare @ModelName nvarchar(50);

declare @YearMade nvarchar(50);

declare @CarType nvarchar(50);

declare @CarColor nvarchar(50);

declare @AuditAction varchar(100);

select @CarID = i.CarID from inserted i;

select @Manufacturer = i.Manufacturer from inserted i;

select @ModelName = i.ModelName from inserted i;

select @YearMade = i.YearMade from inserted i;

select @CarType = i.CarType from inserted i;

select @CarColor = i.CarColor from inserted i;

if update(CarID)

set @AuditAction = 'Updated Record - after update trigger';

if update(Manufacturer)

set @AuditAction = 'Updated Record - after update trigger';

if update(ModelName)

set @AuditAction = 'Updated Record - after update trigger';

if update(YearMade)

set @AuditAction = 'Updated Record - after update trigger';

if update(CarType)

set @AuditAction = 'Updated Record - after update trigger';

if update(CarColor)

set @AuditAction = 'Updated Record - after update trigger';

insert into CarDetails\_Audit

(CarID, Manufacturer, ModelName, YearMade, CarType, CarColor, AuditAction, AuditTimeStamp)

values (@CarID, @Manufacturer, @ModelName, @YearMade, @CarType, @CarColor, @AuditAction, getdate());

print 'after update trigger fired'

go

--after delete

create trigger trrgAfterDelete on CarDetails

for delete

as

declare @CarID int;

declare @Manufacturer nvarchar(50);

declare @ModelName nvarchar(50);

declare @YearMade nvarchar(50);

declare @CarType nvarchar(50);

declare @CarColor nvarchar(50);

declare @AuditAction varchar(100);

select @CarID = d.CarID from deleted d;

select @Manufacturer = d.Manufacturer from deleted d;

select @ModelName = d.ModelName from deleted d;

select @YearMade = d.YearMade from deleted d;

select @CarType = d.CarType from deleted d;

select @CarColor = d.CarColor from deleted d;

set @AuditAction = 'Delete Record - after delete trigger';

insert into CarDetails\_Audit

(CarID, Manufacturer, ModelName, YearMade, CarType, CarColor, AuditAction, AuditTimeStamp)

values (@CarID, @Manufacturer, @ModelName, @YearMade, @CarType, @CarColor, @AuditAction, getdate());

print 'after delete trigger fired'

go

**ASP.NET**

Web.config:

<connectionStrings>

<add name="ConnString" connectionString="Data Source=NCMS219CL7-17;Initial Catalog=AutomobileShop;User ID=sa;Password=bimcs" providerName="System.Data.SqlClient" />

</connectionStrings>

CarDetailsPage:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Web.Configuration;

namespace oel\_2

{

public partial class CarDetailsPage : System.Web.UI.Page

{

SqlConnection conn = new SqlConnection(WebConfigurationManager.ConnectionStrings["ConnString"].ConnectionString);

protected void Page\_Load(object sender, EventArgs e)

{

GetCarDetails();

}

public void GetAudit()

{

tablenamelbl.Text = "Changes Made:";

string qry = "exec GetAudit";

SqlCommand cmd = new SqlCommand(qry, conn);

SqlDataAdapter sd = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sd.Fill(dt);

GridView1.DataSource = dt;

GridView1.DataBind();

}

public void GetCarDetails()

{

tablenamelbl.Text = "Car Details:";

string qry = "exec GetAll";

SqlCommand cmd = new SqlCommand(qry, conn);

SqlDataAdapter sd = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sd.Fill(dt);

GridView1.DataSource = dt;

GridView1.DataBind();

}

public void ClearFields()

{

idtxt.Text = "";

mantxt.Text = "";

modeltxt.Text = "";

yeartxt.Text = "";

typetxt.Text = "";

colortxt.Text = "";

}

protected void addbtn\_Click(object sender, EventArgs e)

{

try

{

conn.Open();

string qry = "exec AddInfo '" + mantxt.Text + "','" + modeltxt.Text + "','" + yeartxt.Text + "','" + typetxt.Text + "','" + colortxt.Text + "'";

SqlCommand cmd = new SqlCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

GetCarDetails();

ClearFields();

}

catch { }

}

protected void updatebtn\_Click(object sender, EventArgs e)

{

try

{

conn.Open();

string qry = "exec UpdateInfo '" + idtxt.Text + "','" + mantxt.Text + "','" + modeltxt.Text + "','" + yeartxt.Text + "','" + typetxt.Text + "','" + colortxt.Text + "'";

SqlCommand cmd = new SqlCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

GetCarDetails();

ClearFields();

}

catch { }

}

protected void changesbtn\_Click(object sender, EventArgs e)

{

GetAudit();

ClearFields();

}

protected void detailsbtn\_Click(object sender, EventArgs e)

{

GetCarDetails();

ClearFields();

}

protected void deletebtn\_Click(object sender, EventArgs e)

{

try

{

conn.Open();

string qry = "exec DeleteInfo '" + idtxt.Text + "'";

SqlCommand cmd = new SqlCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

GetCarDetails();

ClearFields();

}

catch { }

}

}

}

**OUTPUT**



