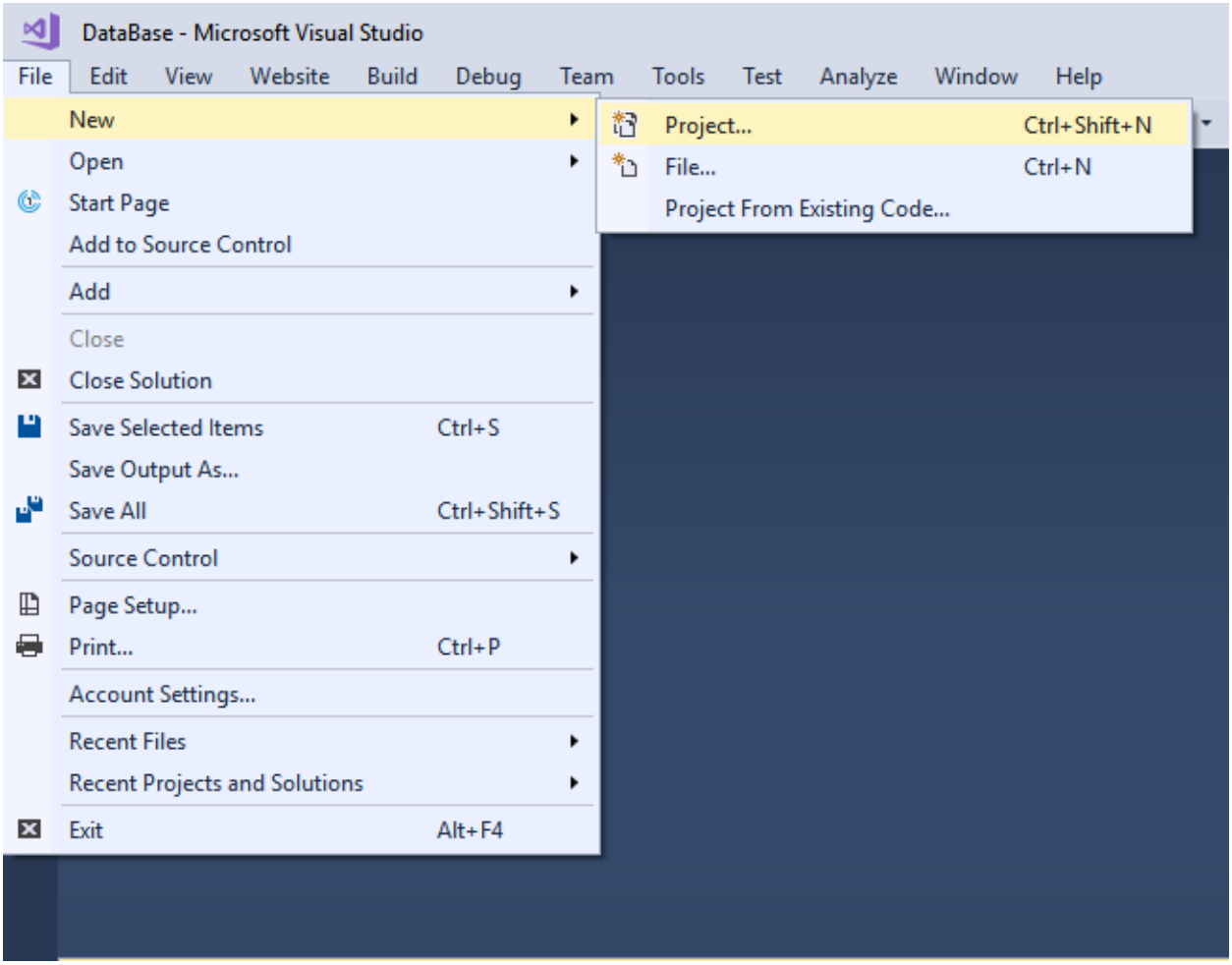
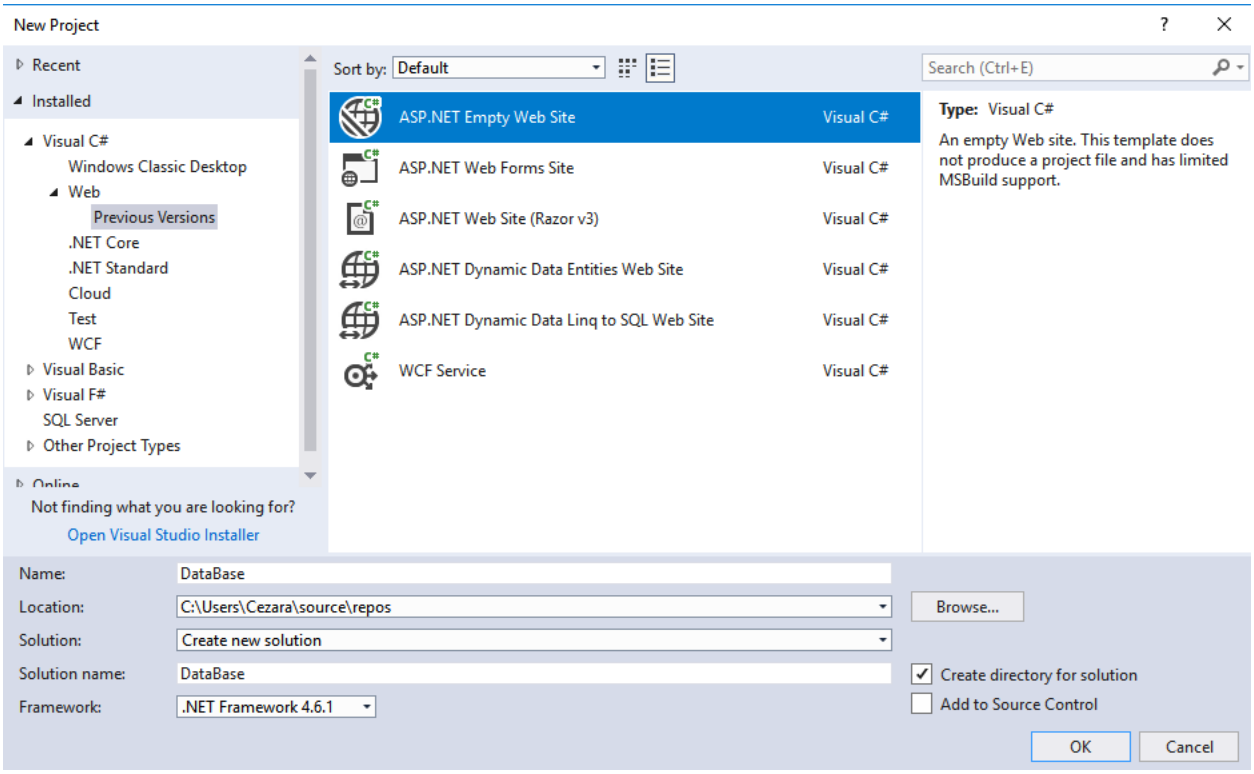
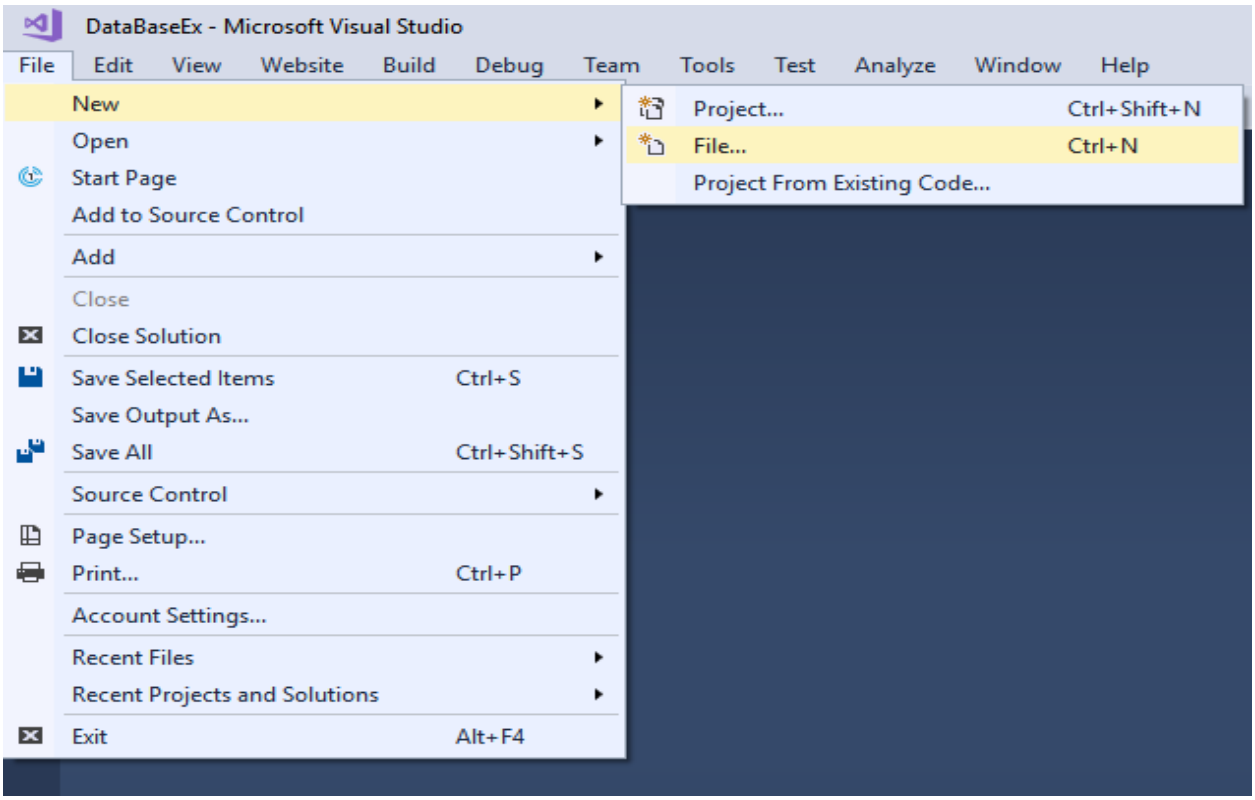
Baza de date. Crearea si modificarea tabelelor. Operatii asupra datelor (CREATE/READ/UPDATE/DELETE). Controale Data-Bound. Cheie primara/ Cheie externa. Cautare in baza de date.

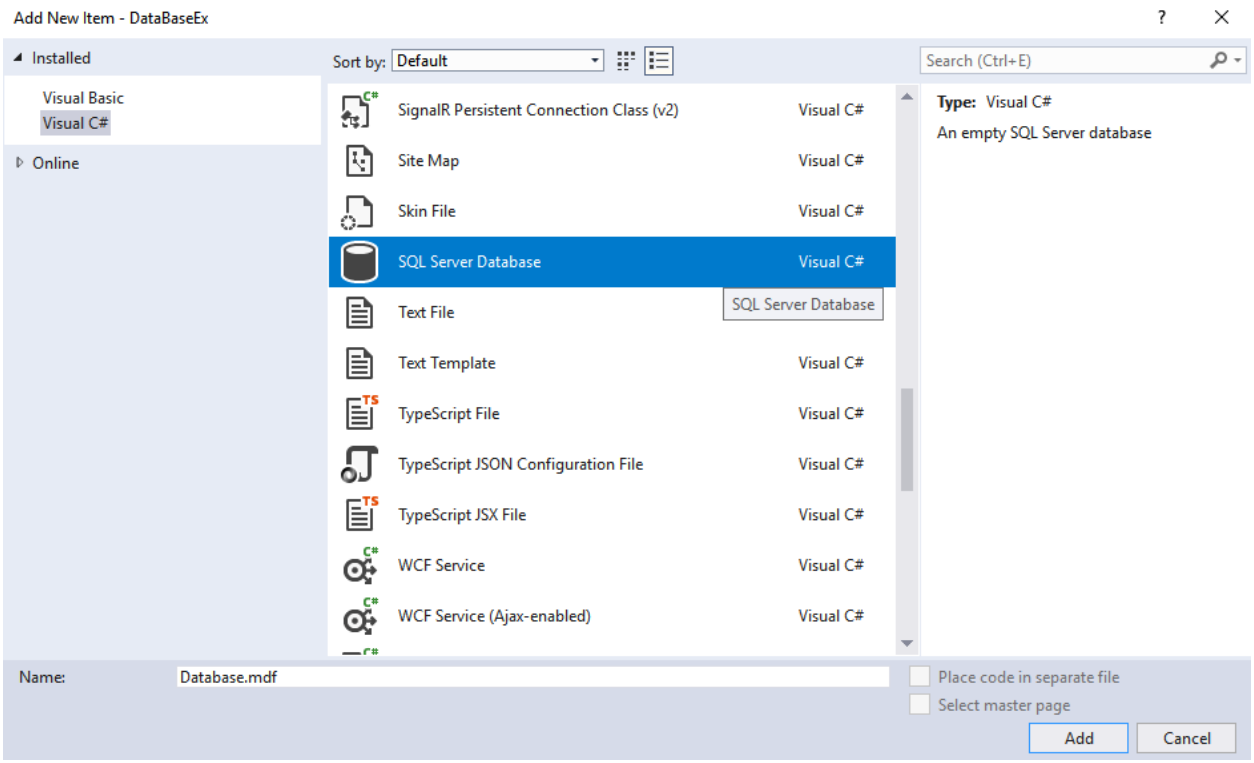
Baza de date. Crearea si modificarea tabelelor

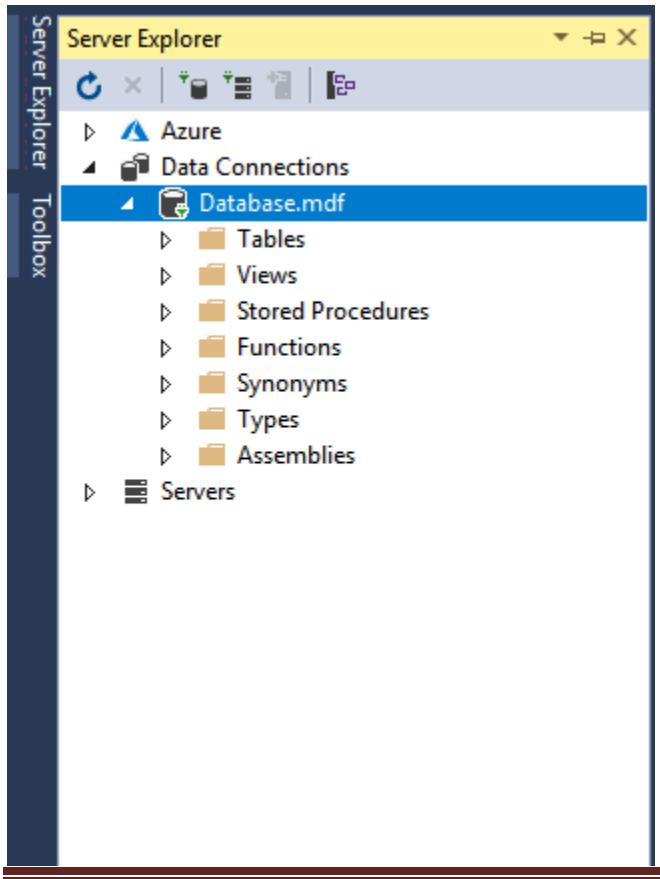
Pasii folositi pentru crearea bazei de date, a tabelelor si popularea tabelelor cu date:

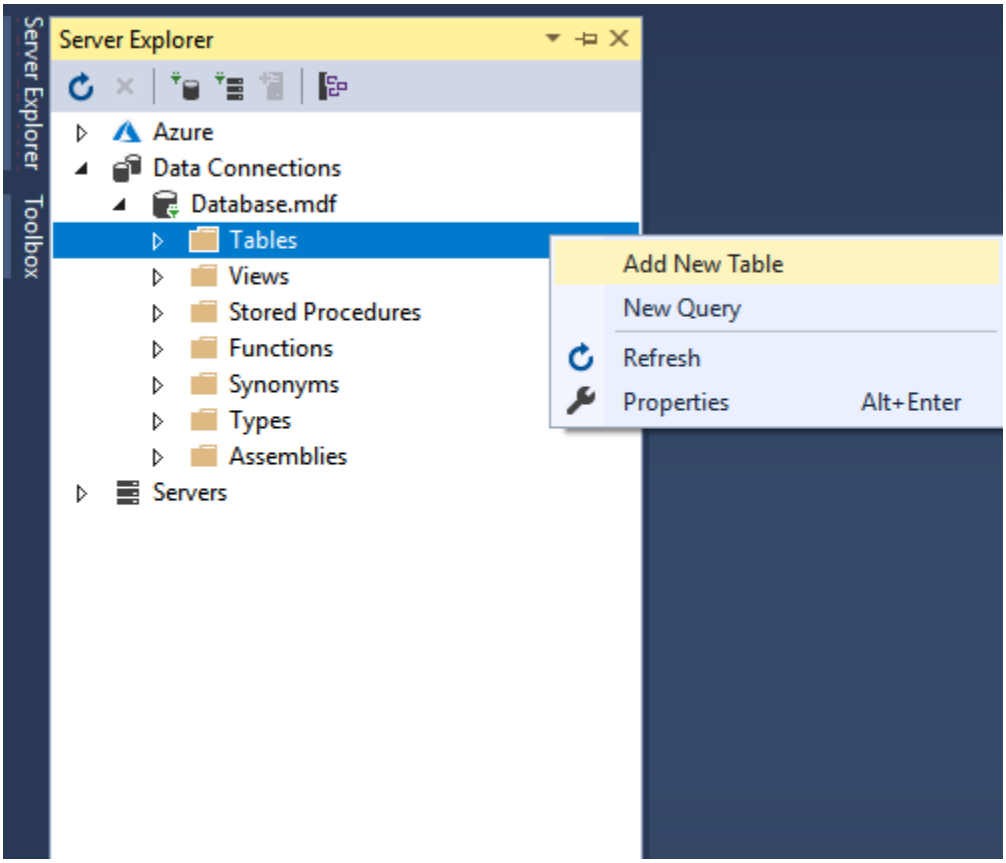


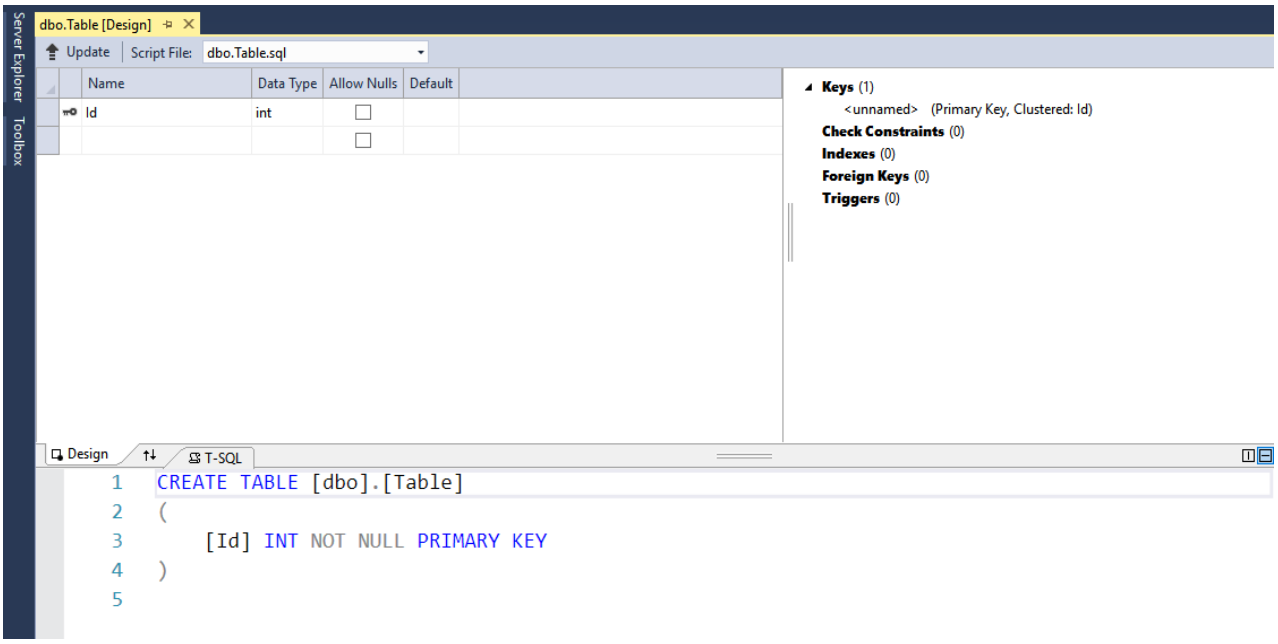

New Project
Recent
Installed
Visual C#
Windows Classic Desktop
Web
Previous Versions
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual F#
SQL Server
Other Project Types
Online
Not finding what you are looking for?
Open Visual Studio Installer
Sort by: Default
Search (Ctrl+E)
ASP.NET Empty Web Site Visual C#
ASP.NET Web Forms Site Visual C#
ASP.NET Web Site (Razor v3) Visual C#
ASP.NET Dynamic Data Entities Web Site Visual C#
ASP.NET Dynamic Data Linq to SQL Web Site Visual C#
WCF Service Visual C#
Type: Visual C#
An empty Web site. This template does not produce a project file and has limited MSBuild support.
Name: DataBase
Location: C:\Users\Cezara\source\repos
Browse...
Solution: Create new solution
Solution name: DataBase
Framework: .NET Framework 4.6.1
Create directory for solution
Add to Source Control
OK
Cancel

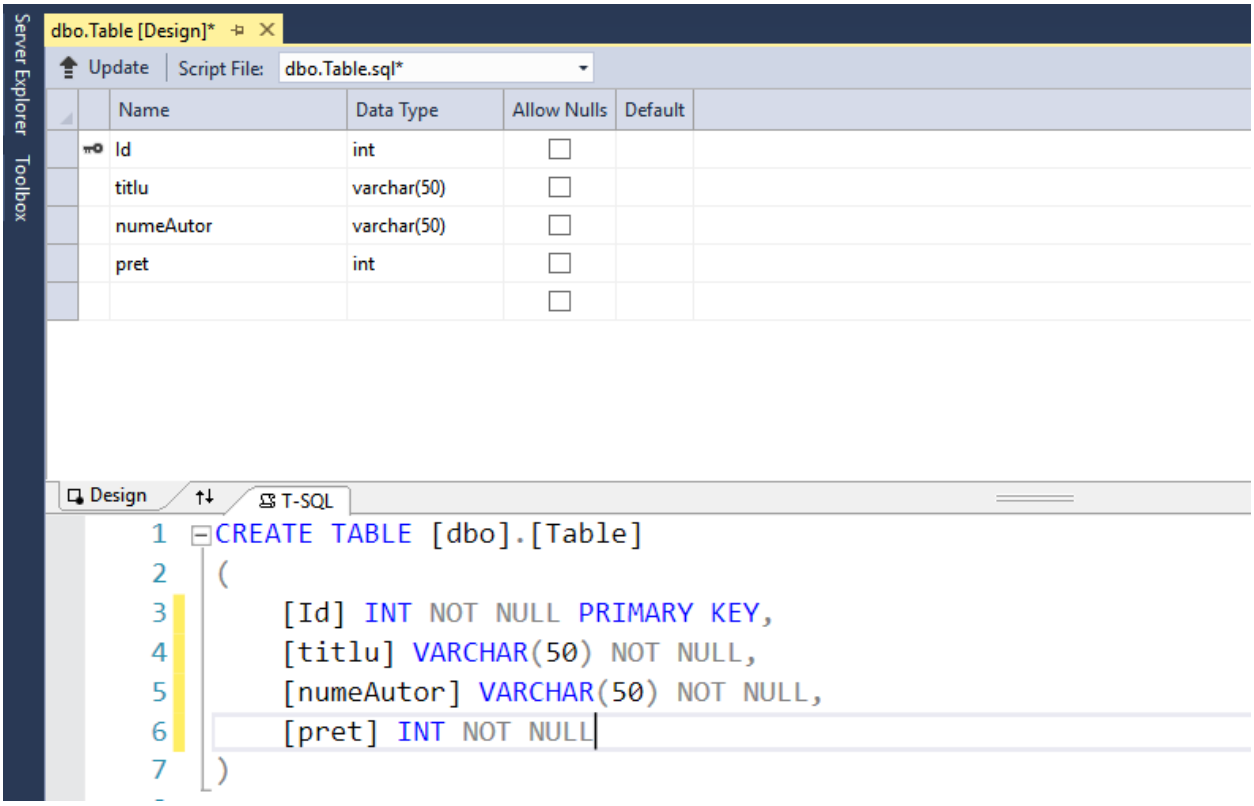

DataBaseEx - Microsoft Visual Studio
File Edit View Website Build Debug Team Tools Test Analyze Window Help
New
Open
Start Page
Add to Source Control
Add
Close
Close Solution
Save Selected Items Ctrl+S
Save Output As...
Save All Ctrl+Shift+S
Source Control
Page Setup...
Print... Ctrl+P
Account Settings...
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
File... Ctrl+N
Project From Existing Code...

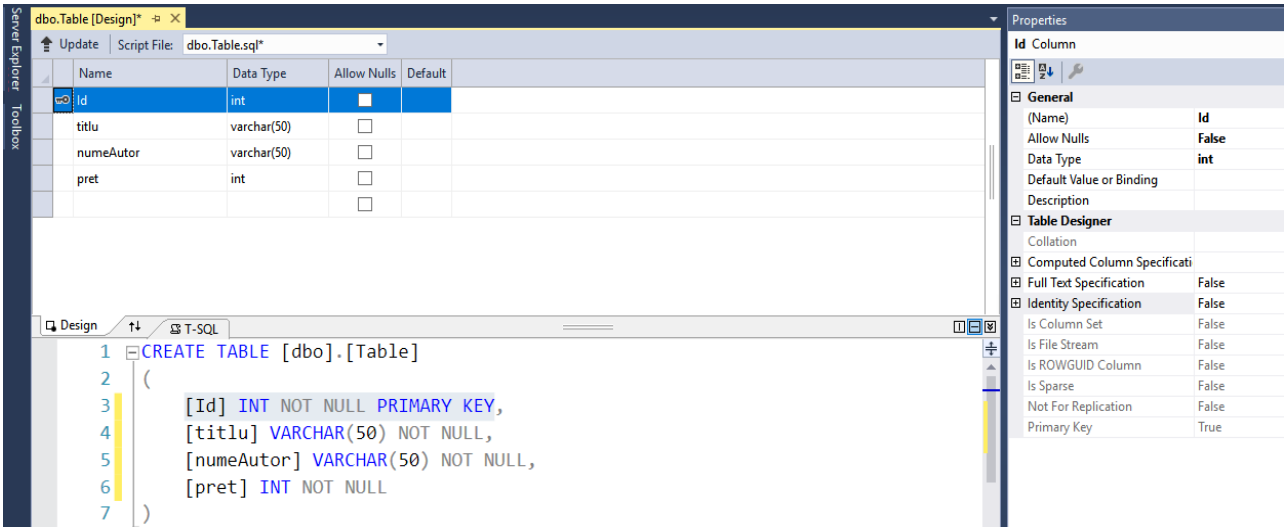

Add New Item - DataBaseEx
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Search (Ctrl+E)
SignalR Persistent Connection Class (v2) Visual C#
Site Map Visual C#
Skin File Visual C#
SQL Server Database Visual C#
Text File Visual C#
Text Template Visual C#
TypeScript File Visual C#
TypeScript JSON Configuration File Visual C#
TypeScript JSX File Visual C#
WCF Service Visual C#
WCF Service (Ajax-enabled) Visual C#
SQL Server Database
Type: Visual C#
An empty SQL Server database
Name: Database.mdf
Place code in separate file
Select master page
Add
Cancel

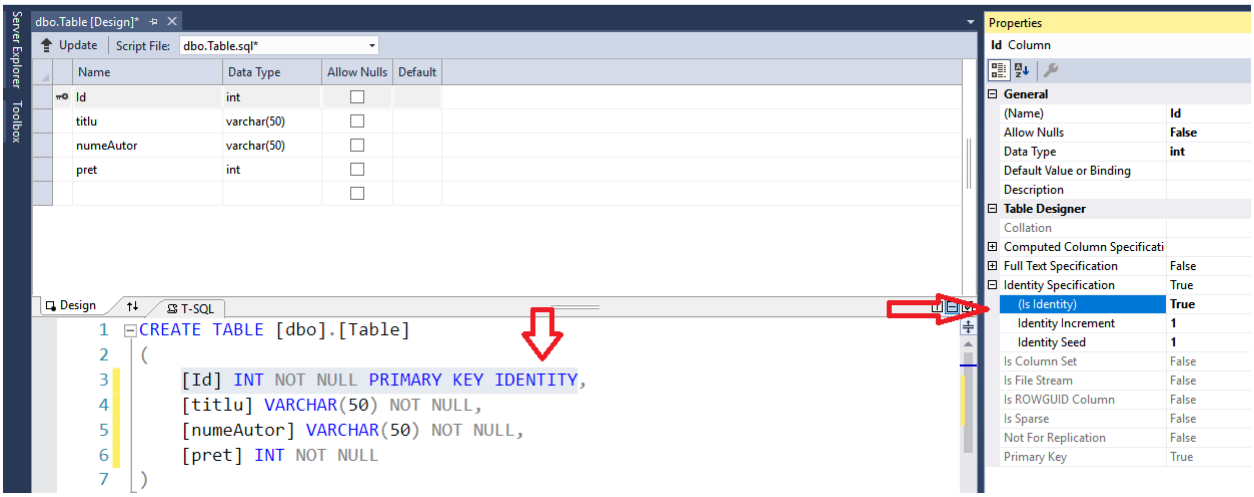

Server Explorer
Toolbox
Server Explorer
Azure
Data Connections
Database.mdf
Tables
Views
Stored Procedures
Functions
Synonyms
Types
Assemblies
Servers

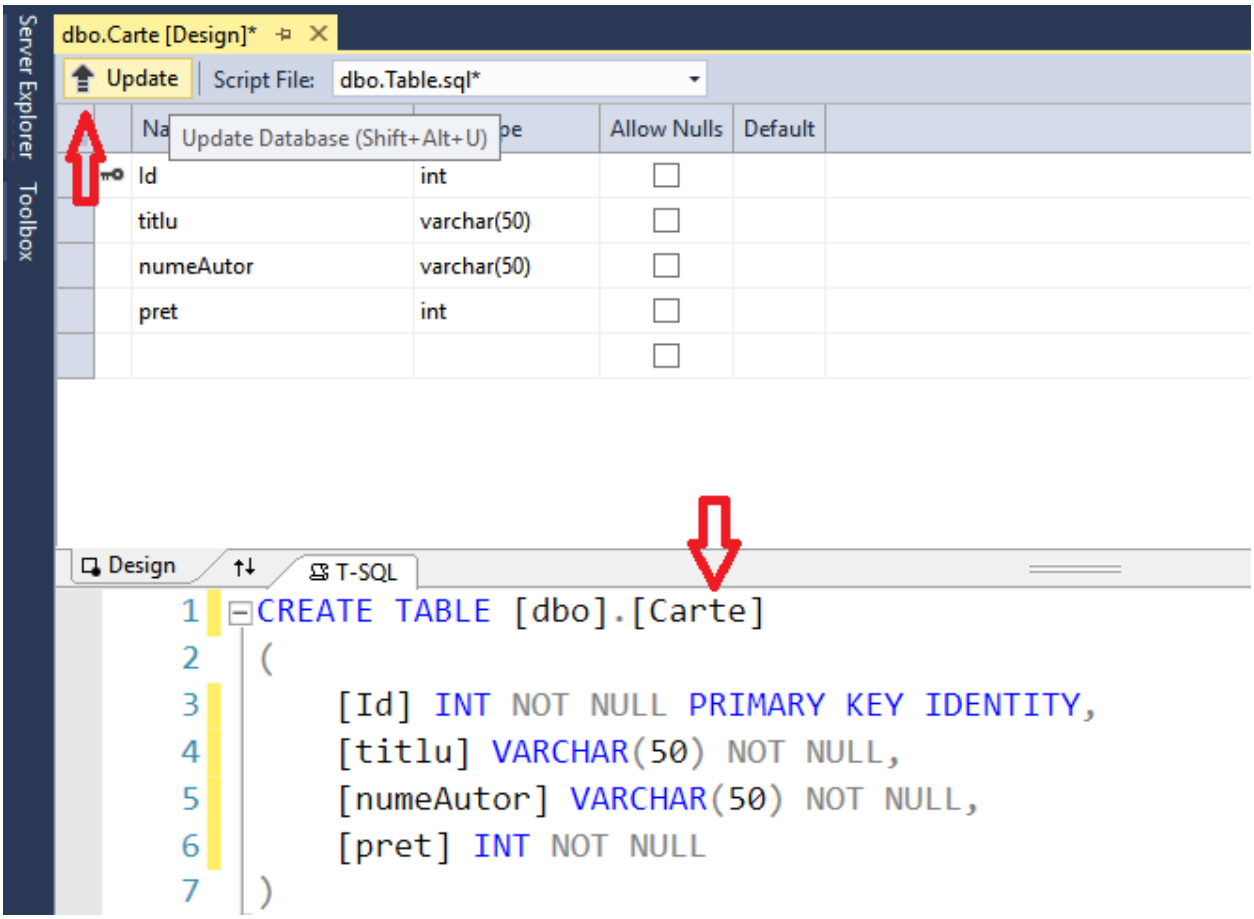

Server Explorer
Toolbox
Server Explorer
Azure
Data Connections
Database.mdf
Tables
Views
Stored Procedures
Functions
Synonyms
Types
Assemblies
Servers
Add New Table
New Query
Refresh
Properties
Alt+Enter

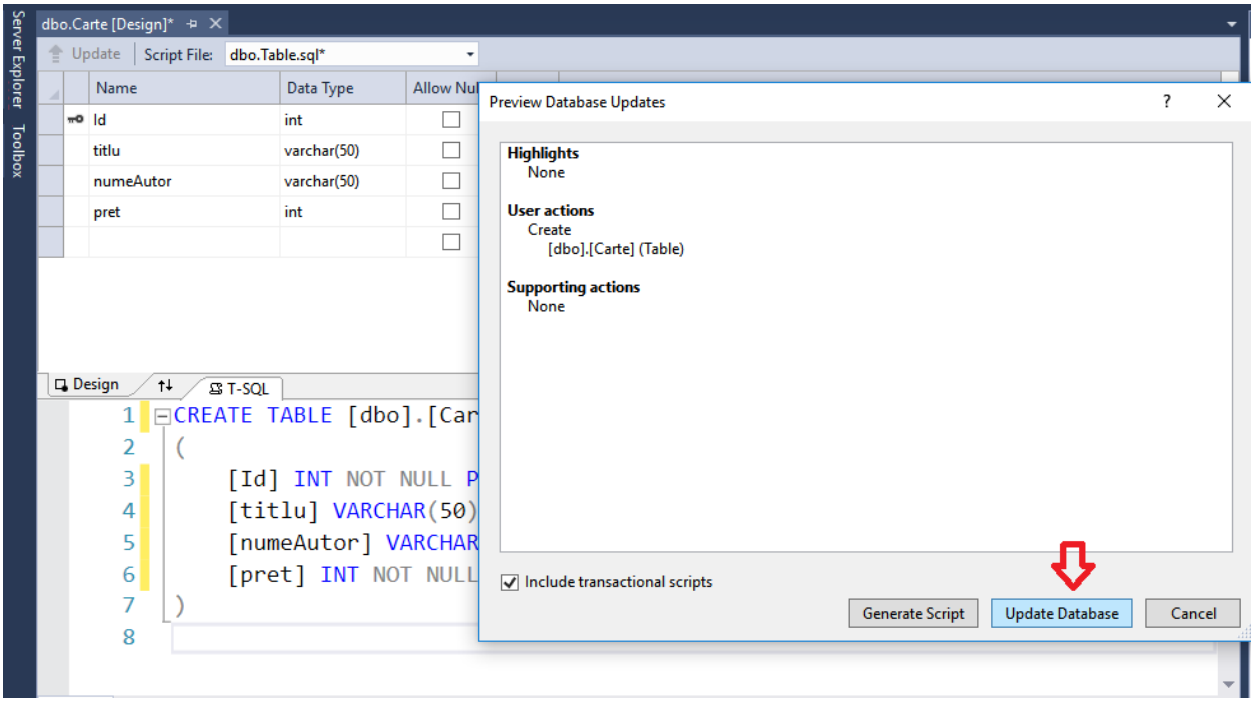

dbo.Table [Design]
Update
Script File:
dbo.Table.sql
Name
Data Type
Allow Nulls
Default
Id
int
Keys (1)
<unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)
Design
T-SQL
1 CREATE TABLE [dbo].[Table]
2 (
3 [Id] INT NOT NULL PRIMARY KEY
4)
5

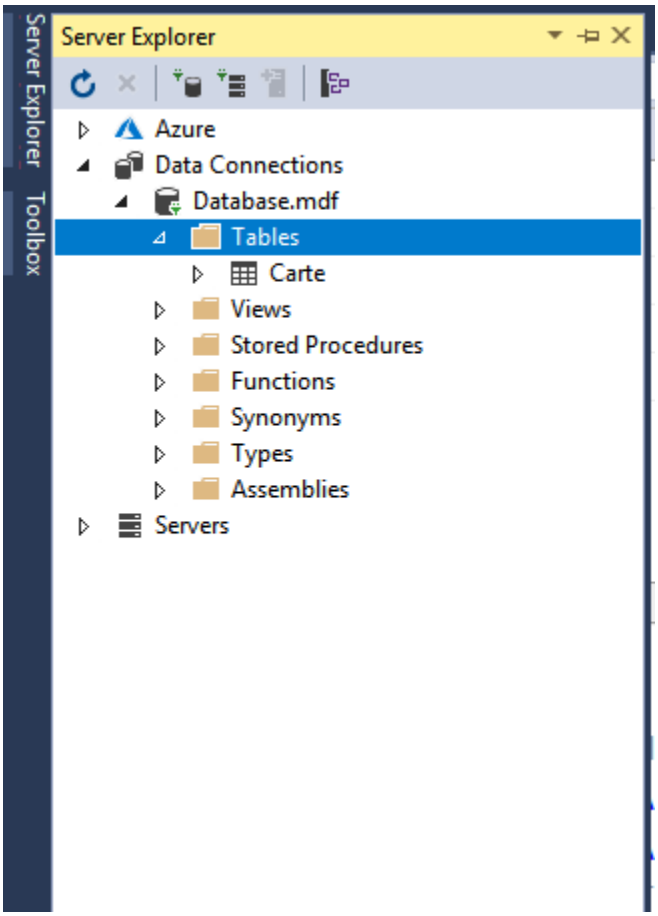

Server Explorer
Toolbox
dbo.Table [Design]*
Update
Script File: dbo.Table.sql*
Name
Data Type
Allow Nulls
Default
Id
int
titlu
varchar(50)
numeAutor
varchar(50)
pret
int
Design
T-SQL
CREATE TABLE [dbo].[Table]
(
[Id] INT NOT NULL PRIMARY KEY,
[titlu] VARCHAR(50) NOT NULL,
[numeAutor] VARCHAR(50) NOT NULL,
[pret] INT NOT NULL
)

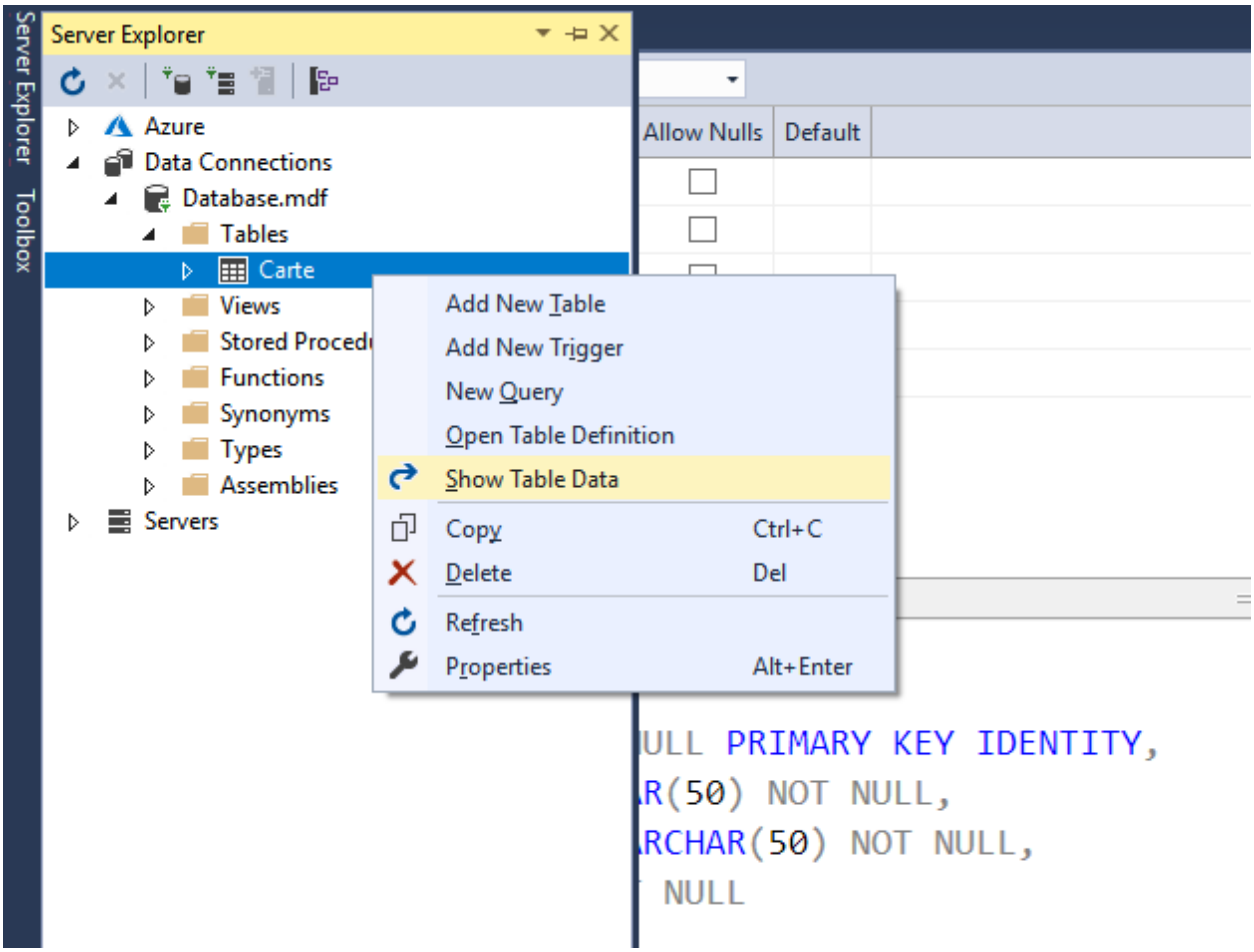

Properties
Id Column
General
(Name) Id
Allow Nulls False
Data Type int
Default Value or Binding
Description
Table Designer
Collation
Computed Column Specificati
Full Text Specification False
Identity Specification False
Is Column Set False
Is File Stream False
Is ROWGUID Column False
Is Sparse False
Not For Replication False
Primary Key True

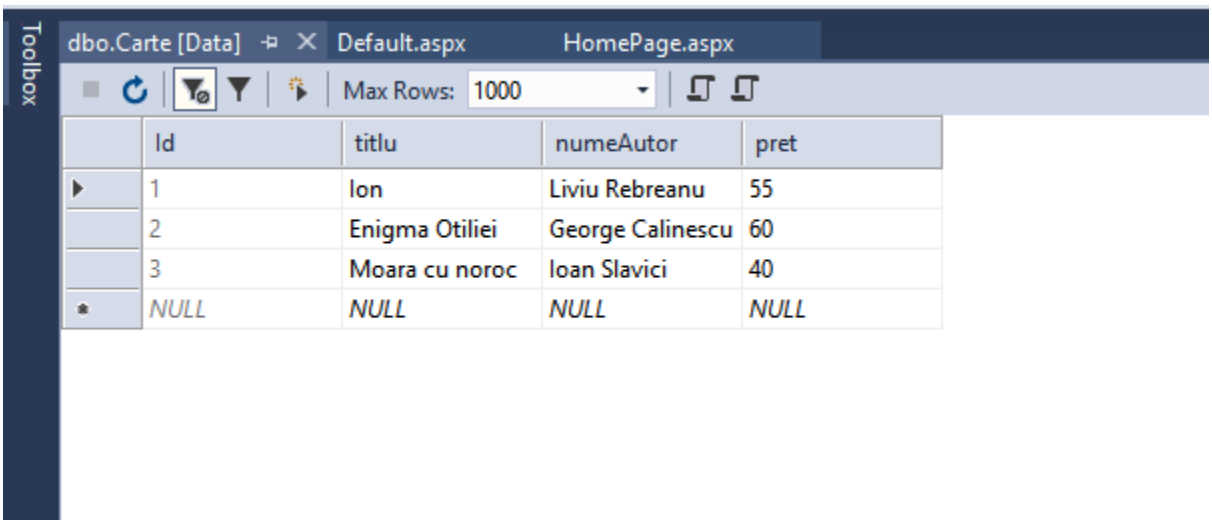

dbo.Table [Design]*
Update
Script File: dbo.Table.sql*
Name
Data Type
Allow Nulls
Default
Id
int
titlu
varchar(50)
numeAutor
varchar(50)
pret
int
Design
T-SQL
CREATE TABLE [dbo].[Table]
(
[Id] INT NOT NULL PRIMARY KEY IDENTITY,
[titlu] VARCHAR(50) NOT NULL,
[numeAutor] VARCHAR(50) NOT NULL,
[pret] INT NOT NULL
)
Properties
Id Column
General
(Name) Id
Allow Nulls False
Data Type int
Default Value or Binding
Description
Table Designer
Collation
Computed Column Specificati
Full Text Specification False
Identity Specification True
(Is Identity) True
Identity Increment 1
Identity Seed 1
Is Column Set False
Is File Stream False
Is ROWGUID Column False
Is Sparse False
Not For Replication False
Primary Key True


Server Explorer
Toolbox
dbo.Carte [Design]*
Update
Script File: dbo.Table.sql*
Update Database (Shift+Alt+U)
Allow Nulls
Default
Id
int
titlu
varchar(50)
numeAutor
varchar(50)
pret
int
Design
T-SQL
CREATE TABLE [dbo].[Carte]
(
[Id] INT NOT NULL PRIMARY KEY IDENTITY,
[titlu] VARCHAR(50) NOT NULL,
[numeAutor] VARCHAR(50) NOT NULL,
[pret] INT NOT NULL
)


dbo.Carte [Design]*
Update
Script File:
dbo.Table.sql*
Name
Data Type
Allow Nul
Id
int
titlu
varchar(50)
numeAutor
varchar(50)
pret
int
Design
T-SQL
CREATE TABLE [dbo].[Car
(
[Id] INT NOT NULL P
[titlu] VARCHAR(50)
[numeAutor] VARCHAR
[pret] INT NOT NULL
)
Preview Database Updates
Highlights
None
User actions
Create
[dbo].[Carte] (Table)
Supporting actions
None
Include transactional scripts
Generate Script
Update Database
Cancel

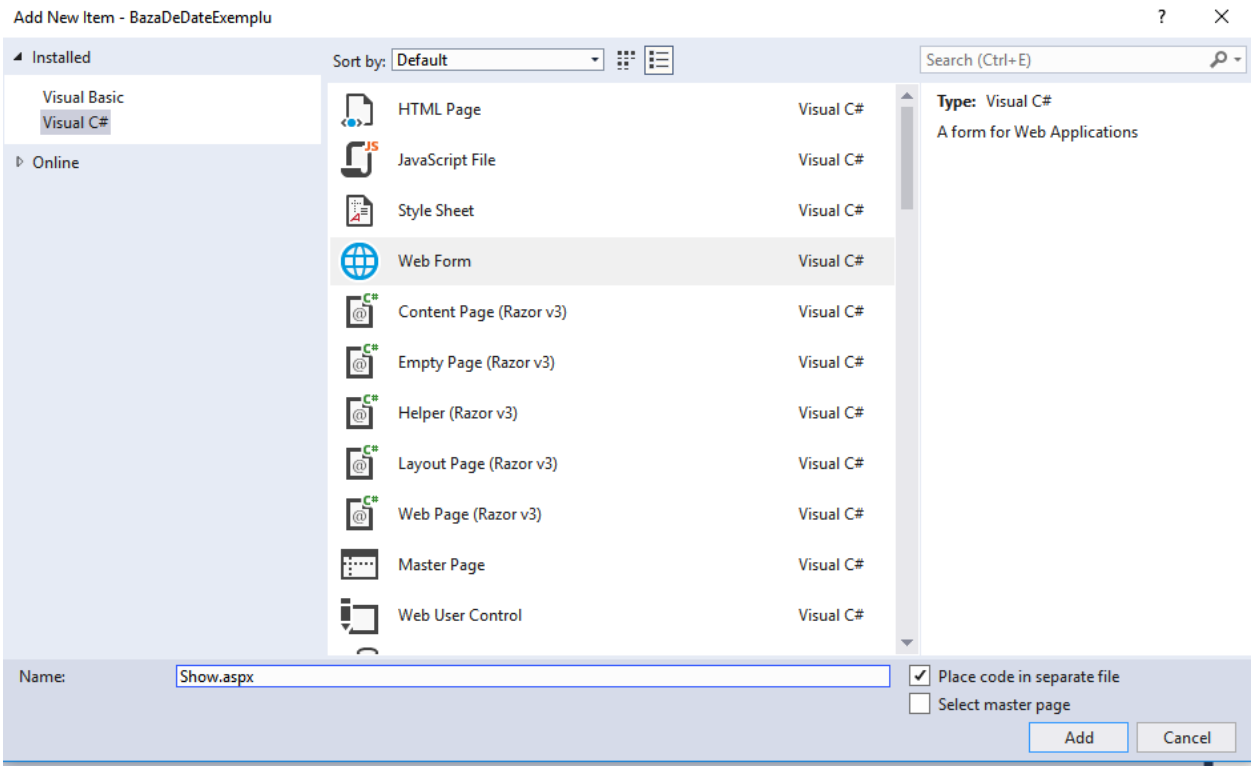

Server Explorer
Toolbox
Server Explorer
Azure
Data Connections
Database.mdf
Tables
Carte
Views
Stored Procedures
Functions
Synonyms
Types
Assemblies
Servers



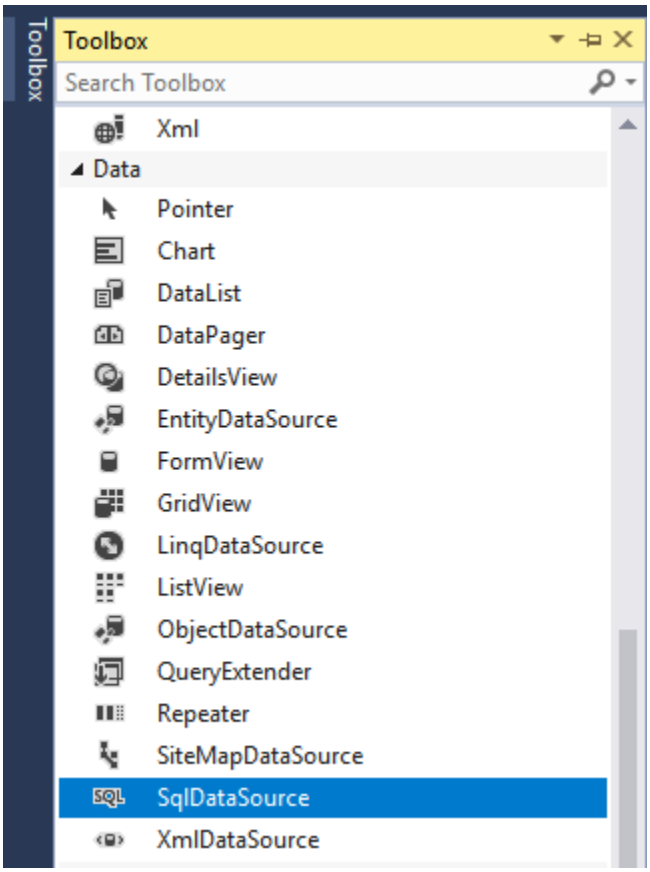


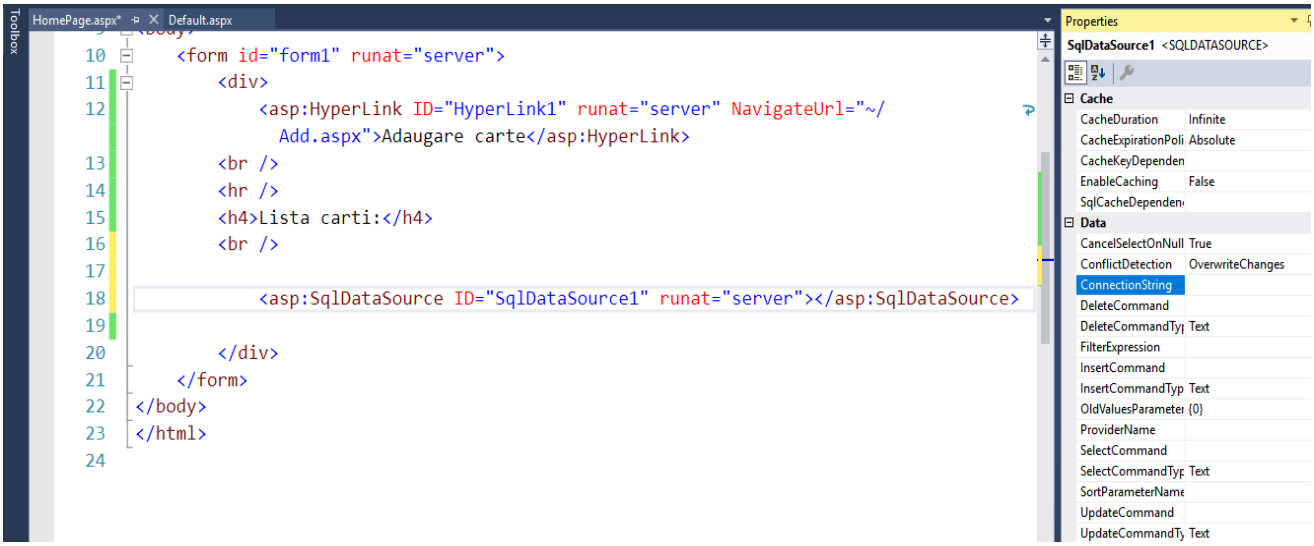
Id	titlu	numeAutor	pret
1	Ion	Liviu Rebreanu	55
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40
NULL	NULL	NULL	NULL

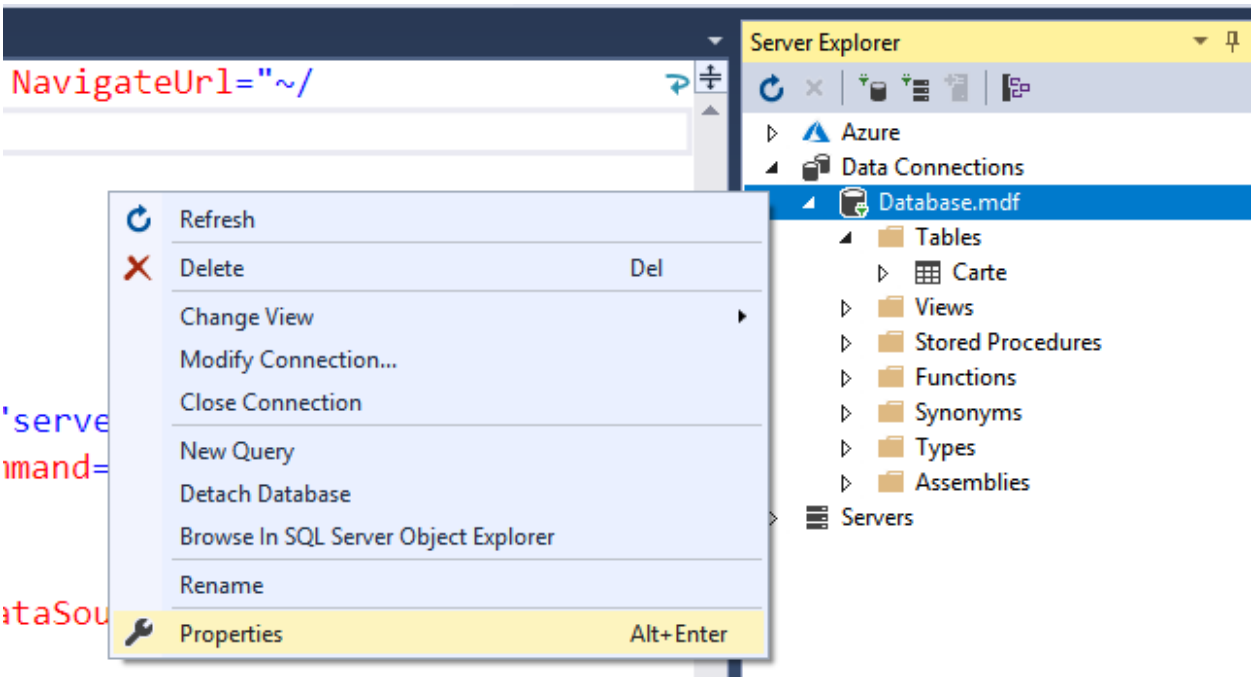
Adaugarea paginilor necesare:

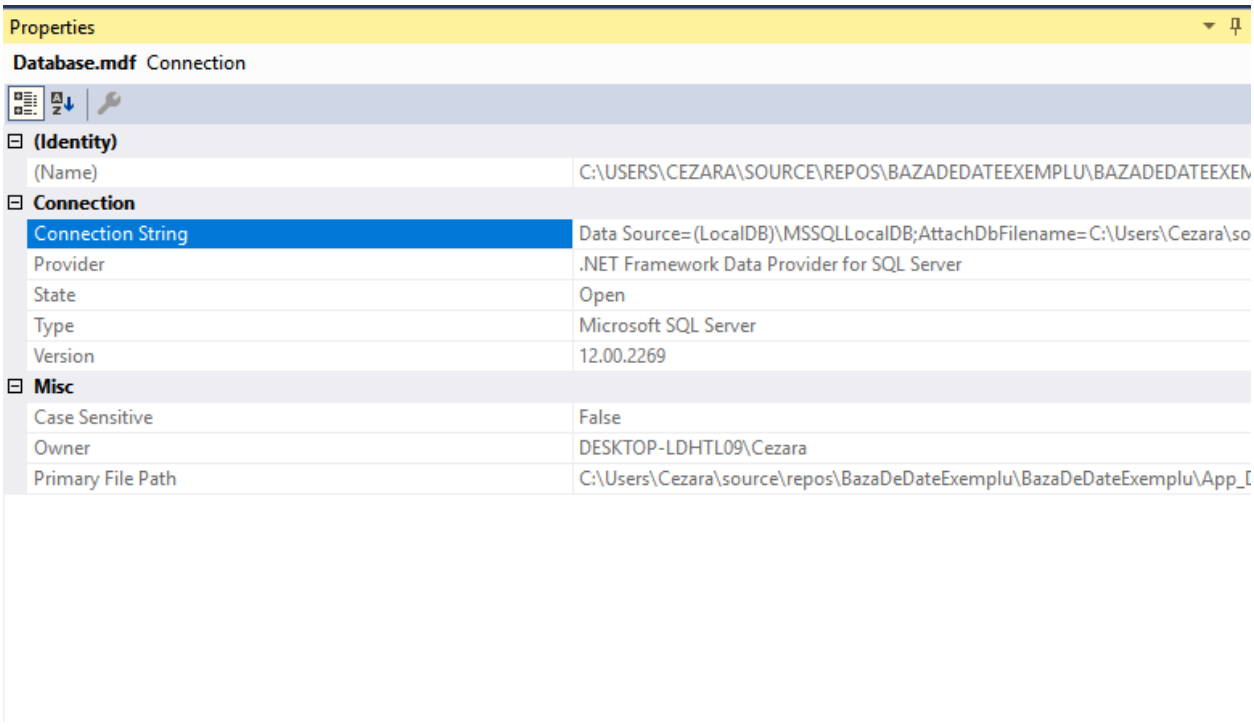


Conexiunea cu baza de date:






NavigateUrl="~/
Server Explorer
Azure
Data Connections
Database.mdf
Tables
Carte
Views
Stored Procedures
Functions
Synonyms
Types
Assemblies
Servers
Refresh
Delete Del
Change View
Modify Connection...
Close Connection
New Query
Detach Database
Browse In SQL Server Object Explorer
Rename
Properties Alt+Enter


Properties
Database.mdf Connection
(Identity)
(Name)
C:\USERS\CEZARA\SOURCE\REPOS\BAZADEDATEEXEMPLU\BAZADEDATEEXEM
Connection
Connection String
Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\so
Provider
.NET Framework Data Provider for SQL Server
State
Open
Type
Microsoft SQL Server
Version
12.00.2269
Misc
Case Sensitive
False
Owner
DESKTOP-LDHTL09\Cezara
Primary File Path
C:\Users\Cezara\source\repos\BazaDeDateExemplu\BazaDeDateExemplu\App_

Selectarea datelor din baza de date:

HomePage.aspx

Pagina principala va avea urmatoarea structura:

Adaugare carte

Lista carti:

Afisare folosind ListView

Ion	Liviu Rebreanu	55	Vizualiare | Editare | Stergere
Enigma Otiliei	George Calinescu	60	Vizualiare | Editare | Stergere
Moara cu noroc	Ioan Slavici	40	Vizualiare | Editare | Stergere

---

Afisare folosind GridView

Id	titlu	numeAutor	pret
1	Ion	Liviu Rebreanu	55
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40

```
<body>
    <form id="form1" runat="server">
        <div>
            <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Add.aspx">Adaugare carte</asp:HyperLink>
        <br />
        <hr />
        <h4>Lista carti:</h4>
        <br />

          <!-- Varianta 1 -->

          <!--  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True"
SelectCommand="select * from carte"></asp:SqlDataSource>  -->

            <!-- Varianta 2 -->

            <asp:SqlDataSource ID="SqlDataSource2" runat="server"
                ConnectionString='<%$ connectionStrings:DBConnection %>'
SelectCommand="select * from carte"></asp:SqlDataSource>
```

```
            <asp:ListView ID="ListView1" ItemPlaceholderID="placeholder"
runat="server" DataSourceID="SqlDataSource1">
                <LayoutTemplate>
                    <h1>Afisare folosind ListView</h1>
                    <table class="table">
                        <asp:PlaceHolder runat="server" ID="placeholder" />
                    </table>
                </LayoutTemplate>
                <ItemTemplate>
                    <tr id="Tr1" runat="server">
                        <td style="color: #ac00dc"><%#Eval("titlu") %></td>
                        <td style="color: #09c"><%#Eval("numeAutor") %></td>
                        <td style="color: #09c"><%#Eval("pret") %></td>
                        <td>
                            <a href="Show.aspx?id=<%#Eval("id") %>">Vizualiare</a> |
                            <a href="Edit.aspx?id=<%#Eval("id") %>">Editare</a> |
                            <a href="Delete.aspx?id=<%#Eval("id") %>">Stergere</a>
                        </td>
                    </tr>
                </ItemTemplate>
            </asp:ListView>

            <hr />
            <h2>Afisare folosind GridView</h2>
            <asp:GridView ID="GridView1" runat="server"
DataSourceID="SqlDataSource1">
            </asp:GridView>

        </div>
    </form>
</body>
</html>
```

Daca folositi Varianta 2, in Web.config, in <configuration> trebuie inclusa urmatoarea secventa de cod:

```
    <connectionStrings>

        <add name="DBConnection" connectionString="Data
        Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\re
        pos\BazaDeDateExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated
        Security=True" providerName="System.Data.SqlClient"/>

    </connectionStrings>
```

connectionString – reprezinta string-ul care ofera informatii necesare pentru conexiunea la baza de date (Vezi paginile 10-12).

Controale Data-Bound

Sunt controale (ListView, GridView, Repeater, DetailsView, etc) care pot fi legate la o sursa de date (ex: **SqlDataSource**) pentru a facilita afisarea si modificarea datelor intr-o aplicatie. Toate aceste controale ofera o varietate de proprietati care pot fi setate pentru a controla aspectul interfetei. Dupa cum putem observa in exemplul de mai sus, unele controale de tip data-bound, cum este **ListView**, permit utilizarea template-urilor (utilizand HTML se specifica unde si cum vor fi afisate datele).

GridView – afiseaza datele sub forma unui tabel si ofera posibilitatea de a sorta coloanele sau de a edita si sterge o singura intrare din baza de date

ListView – afiseaza date dintr-o sursa de date intr-un format definit cu ajutorul template-urilor. ListView permite implicit si operatii de editare, inserare, stergere, precum si functionalitatea de sortare si de paginare.

Metoda DataBinder.Eval – din System.Web.UI

Foloseste "reflectia" pentru evaluarea expresiilor de tip data-binding la runtime (in timpul de executiei).

Reflectia este abilitatea unui program de a examina toate variabilele si metodele sale si de a modifica structura la runtime. Ajuta la gasirea unor clase, obiecte, care se potrivesc cu numarul de argumente si tipul acestora dintr-un anumit punct din cod – la runtime, prin reflectie se analizeaza codul si se cauta obiectul (ex: **Eval(numeCamp)** – prin analizarea codului => (DataRowView)Container.DataItem are un atribut numit "numeCamp", iar la runtime Eval(numeCamp) se transforma automat in (DataRowView)Container.DataItem["numeCamp"]).

Add.aspx

Titlul cartii

Nume autor

Pret

Adauga

Lista carti

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Add.aspx.cs" Inherits="Add"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">

        <asp:Literal ID="EroareBazaDate" runat="server"></asp:Literal>

        <div>
            <asp:Label ID="Label1" runat="server" Text="Titlul cartii"></asp:Label>
            <br />
            <asp:TextBox ID="Titlu" runat="server"></asp:TextBox>
        </div>

        <br />

        <div>
            <asp:Label ID="Label2" runat="server" Text="Nume autor"></asp:Label>
            <br />
            <asp:TextBox ID="NumeAutor" runat="server"></asp:TextBox>
        </div>

        <br />
```

```
        <div>
            <asp:Label ID="Label3" runat="server" Text="Pret"></asp:Label>
            <br />
            <asp:TextBox ID="Pret" runat="server"></asp:TextBox>
        </div>

        <br />
        <br />

        <div>
            <asp:Button ID="Button1" OnClick="AdaugareIntrare_Click" runat="server"
Text="Adauga" />
        </div>

    </form>

    <br />

    <asp:hyperlink ID="Hyperlink1" runat="server" NavigateUrl="~/HomePage.aspx">Lista
carti</asp:hyperlink>

</body>
</html>
```

Add.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;

public partial class Add : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void AdaugareIntrare_Click(object sender, EventArgs e)
    {
        if (Page.IsValid)
        {
            bool valid = true;

            string _Titlu = Titlu.Text;
            string _NumeAutor = NumeAutor.Text;
            string _Pret = Pret.Text;
```

```
            // TODO: Verificarea tuturor stringurilor de mai sus sau adaugarea
validarilor in frontend
            if (_Titlu == "")
            {
                valid = false;
                Response.Write("Eroare: Numele nu poate fi gol");
            }

            if (valid)
            {
                string query = "INSERT INTO carte(titlu, numeAutor, pret) OUTPUT
INSERTED.ID "
                   + " VALUES (@titlu, @numeAutor, @pret)";


             //Varianta 1

                //SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

             // Varianta 2

                SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings["DBConnection"].ConnectionStr
ing);
                con.Open();

                try
                {
                    // Introducem parametrii in cererea SQL
                    SqlCommand com = new SqlCommand(query, con);

                    // primul parametru, titlu, este din query reprezentat de @titlu
iar _Titlu este valoarea variabilei din formular

                    com.Parameters.AddWithValue("titlu", _Titlu);
                    com.Parameters.AddWithValue("numeAutor", _NumeAutor);
                    com.Parameters.AddWithValue("pret", _Pret);

                    int id = (int)com.ExecuteScalar();

                    if (id > 0)
                    {
                        EroareBazaDate.Text = "Informatiile au fost adaugate";
                    }
                    else
                    {
                        EroareBazaDate.Text = "Informatiile nu au fost adaugate";
                    }

                }
```

```
                catch (Exception ex)
                {
                    EroareBazaDate.Text = "Eroare din baza de date: " + ex.Message;
                }
                finally
                {
                    // Nu lasam conexiunea deschisa.
                    con.Close();
                }
            }
        }
    }
}
```

Show.aspx

Titlul cartii

Ion

Nume autor

Liviu Rebreanu

Pret

55

Lista carti

```
<body>
    <form id="form1" runat="server">

        <asp:Literal ID="EroareBazaDate" runat="server"></asp:Literal>

        <div>
            <asp:Label ID="Label1" runat="server" Text="Titlul cartii"></asp:Label>
            <br />
            <asp:TextBox ID="Titlu" runat="server"></asp:TextBox>
        </div>

        <br />

        <div>
            <asp:Label ID="Label2" runat="server" Text="Nume autor"></asp:Label>
            <br />
            <asp:TextBox ID="NumeAutor" runat="server"></asp:TextBox>
        </div>
```

```
        <br />

        <div>
            <asp:Label ID="Label3" runat="server" Text="Pret"></asp:Label>
            <br />
            <asp:TextBox ID="Pret" runat="server"></asp:TextBox>
        </div>

        <br />
        <br />

    </form>

    <br />
    <asp:hyperlink ID="Hyperlink1" runat="server" NavigateUrl="~/HomePage.aspx">Lista
carti</asp:hyperlink>

</body>
```

Show.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class Show : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        // Afisarea de formular
        if (!Page.IsPostBack && Request.Params["id"] != null)
        {
            // Luam ID-ul
            int ID = int.Parse(Request.Params["id"].ToString());

            // Salvam cererea SQL intr-un string
            string query = "SELECT *"
                    + " FROM carte"
                    + " WHERE id = @id";

            // Deschidem conexiunea la baza de date
            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

            con.Open();
```

```
            // Incercam sa executam comanda
            try
            {
                // Se construieste comanda SQL
                SqlCommand com = new SqlCommand(query, con);
                com.Parameters.AddWithValue("id", ID);

                // Se executa comanda si se returneaza valorile intr-un reader
                SqlDataReader reader = com.ExecuteReader();

                // Citim rand cu rand din baza de date
                while (reader.Read())
                {
                    Titlu.Text = reader["titlu"].ToString();
                    NumeAutor.Text = reader["numeAutor"].ToString();
                    Pret.Text = reader["pret"].ToString();
                }
            }
            catch (Exception ex)
            {
                EroareBazaDate.Text = "Eroare din baza de date: " + ex.Message;
            }
            finally
            {
                con.Close();
            }
        }
    }
}
```

Edit.aspx

Adaugare carte

Lista carti:

Afisare folosind ListView

ION	LIVIU REBREANU	100	Vizualizare | Editare | Stergere
Enigma Otiliei	George Calinescu	60	Vizualizare | Editare | Stergere
Moara cu noroc	Ioan Slavici	40	Vizualizare | Editare | Stergere
Capra cu trei iezi	Ion Creanga	35	Vizualizare | Editare | Stergere

Afisare folosind GridView

Id	titlu	numeAutor	pret
1	ION	LIVIU REBREANU	100
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40
4	Capra cu trei iezi	Ion Creanga	35

```
<body>
    <form id="form1" runat="server">

        <asp:Literal ID="EroareBazaDate" runat="server"></asp:Literal>

        <div>
            <asp:Label ID="Label1" runat="server" Text="Titlul cartii"></asp:Label>
            <br />
            <asp:TextBox ID="Titlu" runat="server"></asp:TextBox>
        </div>

        <br />

        <div>
            <asp:Label ID="Label2" runat="server" Text="Nume autor"></asp:Label>
            <br />
            <asp:TextBox ID="NumeAutor" runat="server"></asp:TextBox>
        </div>

        <br />
```

```
        <div>
            <asp:Label ID="Label3" runat="server" Text="Pret"></asp:Label>
            <br />
            <asp:TextBox ID="Pret" runat="server"></asp:TextBox>
        </div>

        <br />
        <br />

        <div>
            <asp:Button ID="Button1" OnClick="EditareIntrare_Click" runat="server"
Text="Editeaza" />
        </div>

    </form>

    <br />

    <asp:hyperlink ID="Hyperlink1" runat="server" NavigateUrl="~/HomePage.aspx">Lista
carti</asp:hyperlink>

</body>
```

Edit.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class Edit : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Afisarea formularului de editare
        if (!Page.IsPostBack && Request.Params["id"] != null)
        {
            // Luam ID-ul
            int ID = int.Parse(Request.Params["id"].ToString());

            // Salvam cererea SQL intr-un string
            string query = "SELECT *"
                    + " FROM carte"
                    + " WHERE id = @id";
```

```
            // Deschidem conexiunea la baza de date

            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

            con.Open();

            // Incercam sa executam comanda
            try
            {
                // Se construieste comanda SQL
                SqlCommand com = new SqlCommand(query, con);
                com.Parameters.AddWithValue("id", ID);

                // Se executa comanda si se returneaza valorile intr-un reader
                SqlDataReader reader = com.ExecuteReader();

                // Citim rand cu rand din baza de date
                while (reader.Read())
                {
                    Titlu.Text = reader["titlu"].ToString();
                    NumeAutor.Text = reader["numeAutor"].ToString();
                    Pret.Text = reader["pret"].ToString();
                }
            }
            catch (Exception ex)
            {
                EroareBazaDate.Text = "Eroare din baza de date: " + ex.Message;
            }
            finally
            {
                con.Close();
            }
        }
    }

    protected void EditareIntrare_Click(object sender, EventArgs e)
    {
        if (Page.IsValid && Request.Params["id"] != null)
        {
            int ID = int.Parse(Request.Params["id"].ToString());

            bool valid = true;

            string _Titlu = Titlu.Text;
            string _NumeAutor = NumeAutor.Text;
            string _Pret = Pret.Text;

            // TODO: Verificarea tuturor stringurilor de mai sus sau adaugarea
validarilor in frontend

            if (_Titlu == "")
```

```
            {
                valid = false;
                Response.Write("Eroare: Numele nu poate fi gol");
            }

            if (valid)
            {
                string query = "UPDATE carte "
                   + "SET titlu = @titlu, numeAutor = @numeAutor, pret = @pret"
                   + " WHERE Id = @id";

                SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

                con.Open();

                try
                {
                    // Introducem parametrii in cererea SQL
                    SqlCommand com = new SqlCommand(query, con);

                    // primul parametru, titlu, este din query reprezentat de @titlu
iar _Titlu este valoarea variabilei din formular

                    com.Parameters.AddWithValue("titlu", _Titlu);
                    com.Parameters.AddWithValue("numeAutor", _NumeAutor);
                    com.Parameters.AddWithValue("pret", _Pret);
                    com.Parameters.AddWithValue("id", ID);

                    // Update, delete si insert sunt NON QUERY -> nu returneaza un
SqlReader
                    int campuriAfectate = com.ExecuteNonQuery(); // Returneaza INT
reprezentand numarul de campuri afectate in bd

                    if (campuriAfectate > 0)
                    {
                        EroareBazaDate.Text = "Informatiile au fost modificate";
                    }
                    else
                    {
                        EroareBazaDate.Text = "Informatiile nu au fost modificate";
                    }

                }

                catch (Exception ex)
                {
                    EroareBazaDate.Text = "Eroare din baza de date: " + ex.Message;
                }
```

```
                finally
                {
                    // Nu lasam conexiunea deschisa.
                    con.Close();
                }
            }
        }
    }
}
```

Delete.aspx

```
<body>
    <asp:hyperlink ID="Hyperlink1" runat="server"
NavigateUrl="~/HomePage.aspx">Inapoi la pagina principala</asp:hyperlink>
</body>
```

Delete.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class Delete : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Afisarea formularului
        if (Request.Params["id"] != null)
        {
            // Luam ID-ul
            int ID = int.Parse(Request.Params["id"].ToString());

            // Salvam cererea SQL intr-un string
            string query = "DELETE"
                    + " FROM carte"
                    + " WHERE id = @id";

            // Deschidem conexiunea la baza de date
            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");
```

```
            // Incercam sa executam comanda
            try
            {
                con.Open();
                // Se construieste comanda SQL
                SqlCommand com = new SqlCommand(query, con);
                com.Parameters.AddWithValue("id", ID);

                int deleted = com.ExecuteNonQuery();

                if (deleted > 0)
                {
                    Response.Write("Intrarea a fost stearsa din baza de date");
                }
                else
                {
                    Response.Write("Intrarea nu a fost stearsa. Va rugam incercati
din nou");
                }

            }
            catch (Exception ex)
            {
                Response.Write("Eroare din baza de date: " + ex.Message);
            }
            finally
            {
                con.Close();
            }
        }
    }
}
```

Search.aspx

Cautare

calinescu

Rezultate cautare

Id	titlu	numeAutor	pret
2	Enigma Otiliei	George Calinescu	60

Cautare

ara

Rezultate cautare

Id	titlu	numeAutor	pret
3	Moara cu noroc	Ioan Slavici	40

```
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Literal ID="MessagePlaceholder" runat="server"></asp:Literal>

        <h1>Cautare</h1>

        <div>
            <asp:TextBox ID="SearchParam" runat="server" Width="79%" style="float:
left;"></asp:TextBox>
            <asp:Button ID="Button1" runat="server" OnClick="PerformSearch_Click"
Width="20%" style="float:left" Text="Cautare" />
        </div>

        <br /><br /><br />
```

```
        <div>

            <h4>Rezultate cautare</h4>
            <asp:GridView ID="GridView1" runat="server">
            </asp:GridView>

        </div>
    </div>
    </form>
</body>
```

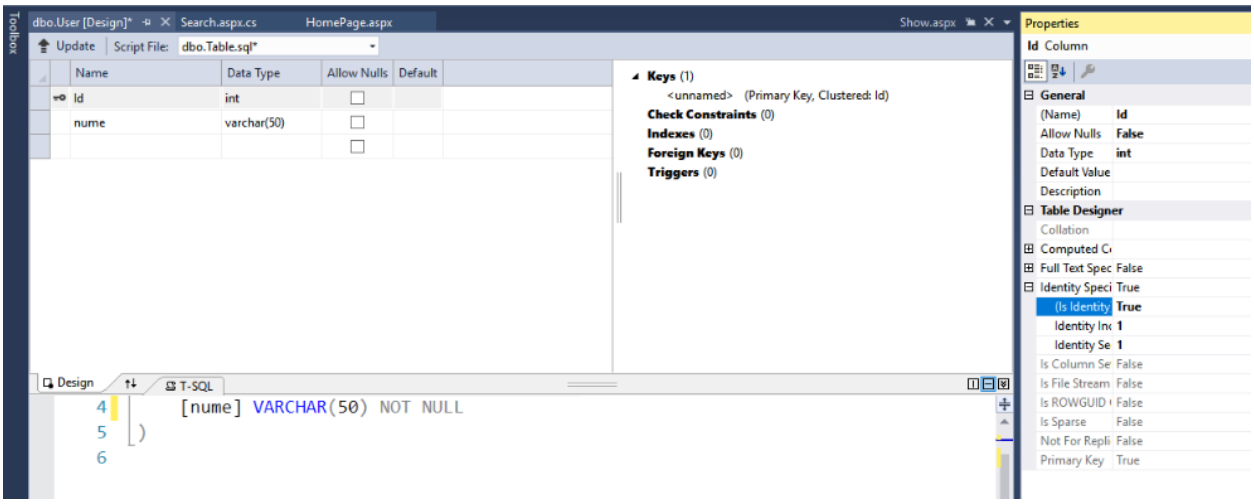
Search.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class Search : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            string query = "SELECT *"
                    + " FROM carte";

            // Deschidem conexiunea la baza de date
            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");


            // Incercam sa executam comanda
            try
            {
                con.Open();

                // Se construieste comanda SQL
                SqlCommand com = new SqlCommand(query, con);

                // Se executa comanda si se returneaza valorile intr-un reader

                SqlDataReader reader = com.ExecuteReader();
                GridView1.DataSource = reader; // Alocam readerul pentru citirea
datelor
                GridView1.DataBind(); // Incarca datele din reader

            }
```

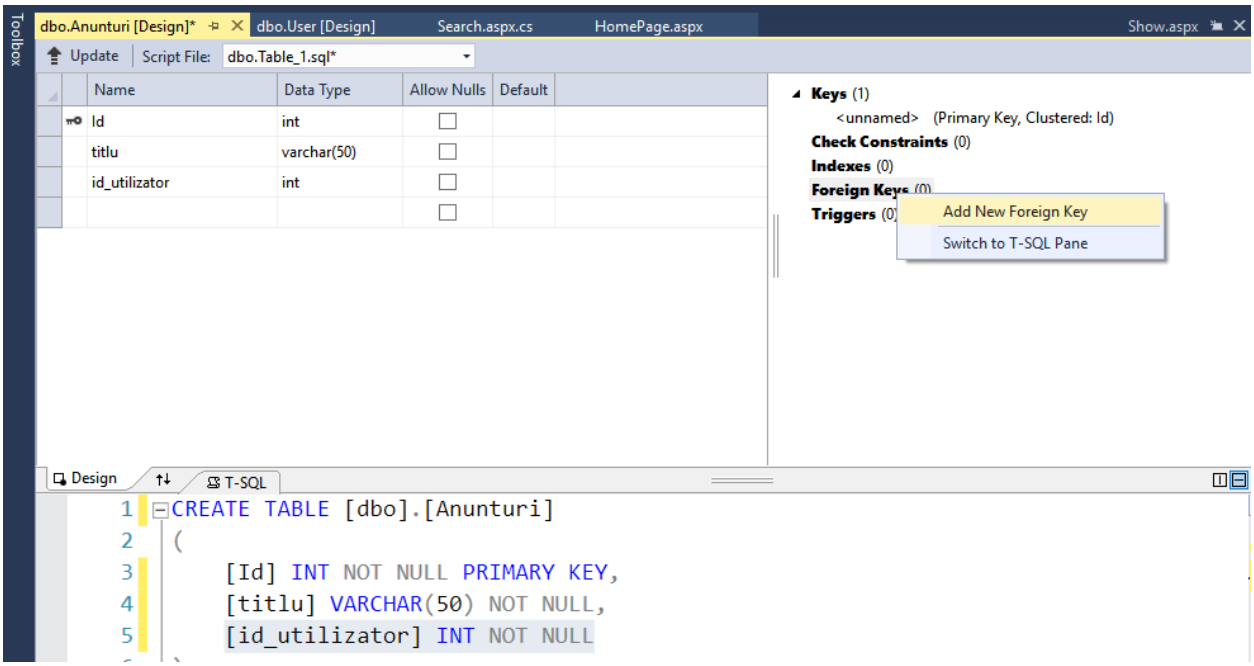
```
            catch (Exception ex)
            {
                MessagePlaceholder.Text = "Eroare din baza de date: " + ex.Message;
            }
            finally
            {
                con.Close();
            }
        }
    }


    protected void PerformSearch_Click(object sender, EventArgs e)
    {
        string search = "%" + SearchParam.Text.ToString() + "%";

        string query = "SELECT * FROM carte WHERE titlu LIKE @search OR numeAutor
LIKE @search OR pret LIKE @search";

        // Deschidem conexiunea la baza de date
        SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

        con.Open();

        // Incercam sa executam comanda

        try
        {
            // Se construieste comanda
            SqlCommand com = new SqlCommand(query, con);
            com.Parameters.AddWithValue("search", search);

            // Se executa comanda si se returneaza valorile intr-un reader
            SqlDataReader reader = com.ExecuteReader();
            GridView1.DataSource = reader; // Alocam readerul pentru citirea datelor
            GridView1.DataBind(); // Incarca datele din reader

        }
        catch (Exception ex)
        {
            MessagePlaceholder.Text = "Eroare din baza de date: " + ex.Message;
        }
        finally
        {
            con.Close();
        }
    }
}
```

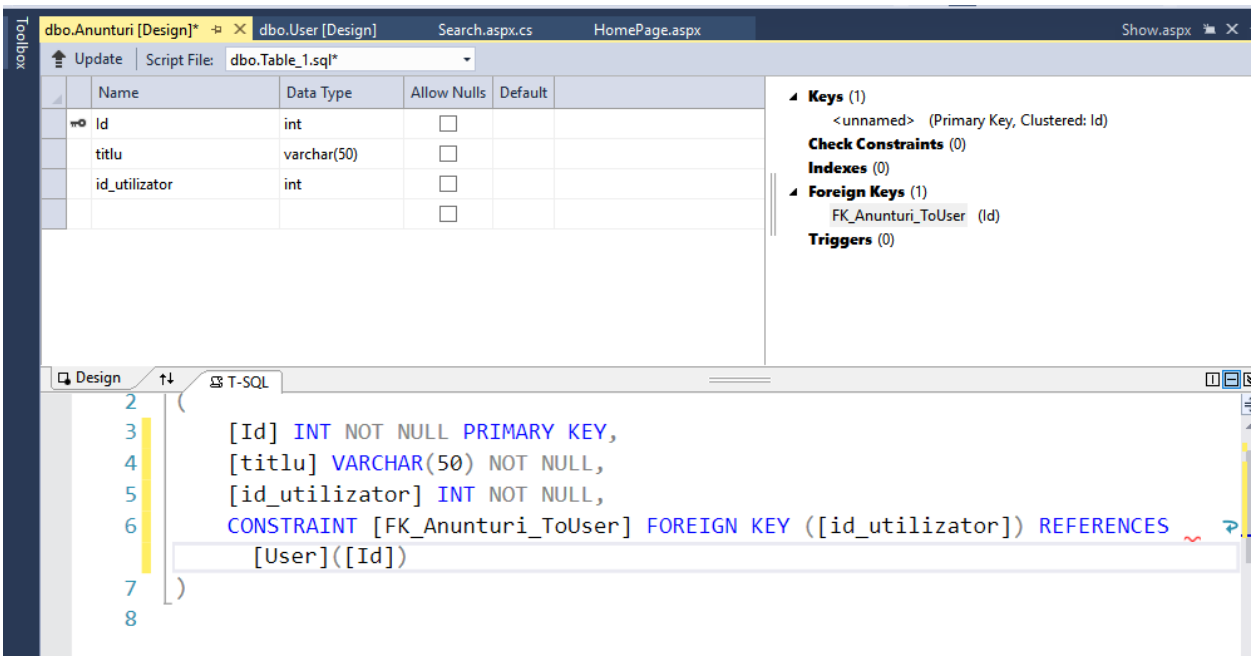
Cheie primara/ Cheie externa

Ca exemplu consideram tabelul **User** si tabelul **Anunturi** si relatia: **un utilizatori publica mai multe anunturi**



Adaugare **id_utilizator** cheie externa:




Toolbox
dbo.Anunturi [Design]*
dbo.User [Design]
Search.aspx.cs
HomePage.aspx
Show.aspx
Update
Script File: dbo.Table_1.sql*
Name
Data Type
Allow Nulls
Default
Id
int
titlu
varchar(50)
id_utilizator
int
Keys (1)
<unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
FK_Anunturi_ToUser (Id)
Triggers (0)
Design
T-SQL
(
[Id] INT NOT NULL PRIMARY KEY,
[titlu] VARCHAR(50) NOT NULL,
[id_utilizator] INT NOT NULL,
CONSTRAINT [FK_Anunturi_ToUser] FOREIGN KEY ([id_utilizator]) REFERENCES
[User]([Id])
)

SQL Injection

Sa analizam exemplul urmator de cautare in baza de date dupa anumite cuvinte cheie(titlu, numele autorului, pret) si de afisare a rezultatelor.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class Search : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            string query = "SELECT *"
                    + " FROM carte";

            SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

            try
            {
                con.Open();

                SqlCommand com = new SqlCommand(query, con);

                // Se executa comanda si se returneaza valorile intr-un reader
                SqlDataReader reader = com.ExecuteReader();
```

```
                GridView1.DataSource = reader; // Alocam readerul pentru citirea
datelor
                GridView1.DataBind(); // Incarca datele din reader
            }
            catch (Exception ex)
            {
                MessagePlaceholder.Text = "Eroare din baza de date: " + ex.Message;
            }
            finally
            {
                con.Close();
            }
        }
    }

    protected void PerformSearch_Click(object sender, EventArgs e)
    {
        string search = "%" + SearchParam.Text.ToString() + "%";

        // Folosim @search pentru a putea adauga valoarea folosind AddWithValue
        //string query = "SELECT * FROM carte WHERE titlu LIKE @search OR numeAutor
LIKE @search OR pret LIKE @search";

        // SELECT * FROM users WHERE email = 'user@example.com';

        // ';DELETE FROM carte WHERE pret = 123;--

        // Exemplu de posibil SQLI deoarece variabila "search" nu este escapata
        string query = "SELECT * FROM carte WHERE titlu LIKE '" + search + "' OR
numeAutor LIKE '"
                       + search + "' OR pret LIKE '" + search + "'";

        // Query-ul devine pentru ';DELETE FROM carte WHERE pret = 123;--
        // SELECT * FROM carte WHERE TITLU LIKE '%'; DELETE FROM carte WHERE pret =
123; --%' OR numeAuthor like..


        SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Cezara\source\repos\BazaDeDat
eExemplu\BazaDeDateExemplu\App_Data\Database.mdf;Integrated Security=True");

        con.Open();

        try
        {
            SqlCommand com = new SqlCommand(query, con);

            // Folosim AddWithValue pentru a face escape automat si a evita SQLI
            // com.Parameters.AddWithValue("search", search);

            // Se executa comanda si se returneaza valorile intr-un reader
            SqlDataReader reader = com.ExecuteReader();
            GridView1.DataSource = reader; // Alocam readerul pentru citirea datelor
            GridView1.DataBind(); // Incarca datele din reader
```

```
        }
        catch (Exception ex)
        {
            MessagePlaceholder.Text = "Eroare din baza de date: " + ex.Message;
        }
        finally
        {
            con.Close();
        }
    }
}
```

SQL Injection folosind urmatoarea varianta:

```
        string query = "SELECT * FROM carte WHERE titlu LIKE '" + search + "'
OR numeAutor LIKE '" + search + "' OR pret LIKE '" + search + "'";
```

Cautare

Rezultate cautare

Id	titlu	numeAutor	pret
1	ION	LIVIU REBREANU	100
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40
9	test sqli	test sqli	123
10	test sqli	test sqli	123
11	test sqli	test sqli	123
12	test sqli	test sqli	123
13	test sqli	test sqli	123
14	test sqli	test sqli	123
15	test sqli	test sqli	123

Cautare

';DELETE FROM carte WHERE pret = 123;--

Rezultate cautare

Id	titlu	numeAutor	pret
1	ION	LIVIU REBREANU	100
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40
9	test sqli	test sqli	123
10	test sqli	test sqli	123
11	test sqli	test sqli	123
12	test sqli	test sqli	123
13	test sqli	test sqli	123
14	test sqli	test sqli	123
15	test sqli	test sqli	123

Cautare

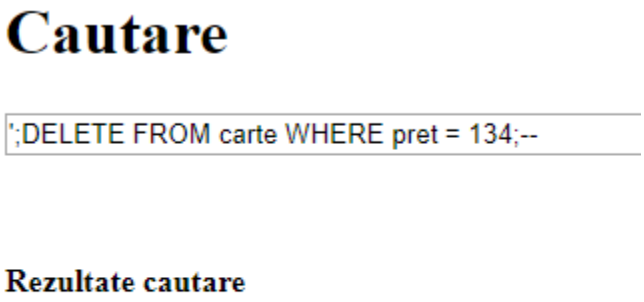
';DELETE FROM carte WHERE pret = 123;--

Rezultate cautare

Id	titlu	numeAutor	pret
1	ION	LIVIU REBREANU	100
2	Enigma Otiliei	George Calinescu	60
3	Moara cu noroc	Ioan Slavici	40

SQL Injection folosind urmatoarea varianta:

```
string query = "SELECT * FROM carte WHERE titlu LIKE @search OR
numeAutor LIKE @search OR pret LIKE @search";

com.Parameters.AddWithValue("search", search);
```



- **Se observa ca a doua varianta previne SQL Injection**