Prof. Carmen Gal

Relaţii între tabele

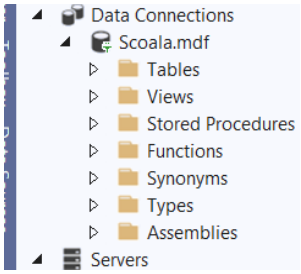
Cuprins

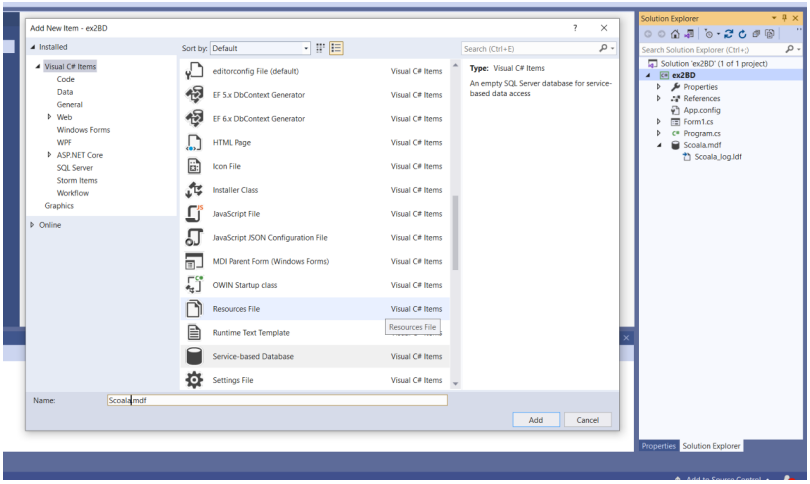
1. Creare BD Şcoala. Tabele.....1
2. Popularea tabelelor.....2
3. Stabilirea relaţiei.....2
4. Crearea unui meniu.....3
5. Încărcarea tabelelor în DataGridView.....3
6. Interogări. Proceduri stocate.....5
7. Rapoarte.....21

1. Creare BD Şcoala. Tabele.

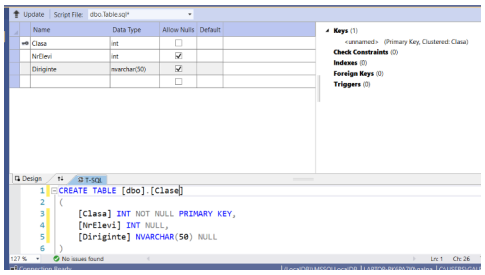
(click dreapta pe numele proiectului->Add->New Item->Service-based Database)

Se executa dublu click pe Scoala.mdf si se deschide Server-Explorer:



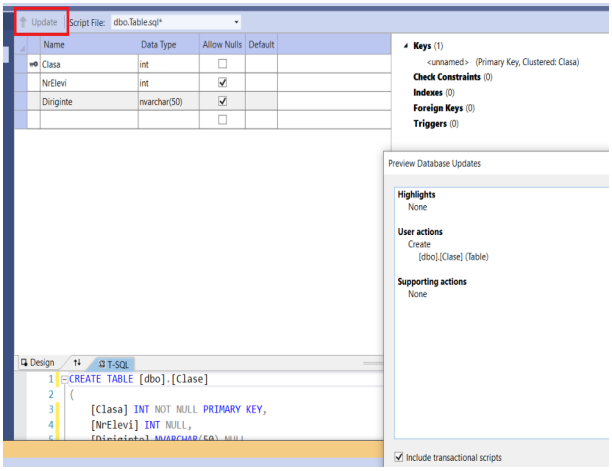


Se executa click dreapta pe Tables->Add Table->adaugam tabelul Clase:

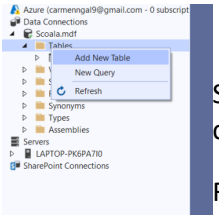


Clasa este cheie primara. In partea de jos apare **comanda sql pentru crearea unui tabel.**

Alegem update table din partea de sus.

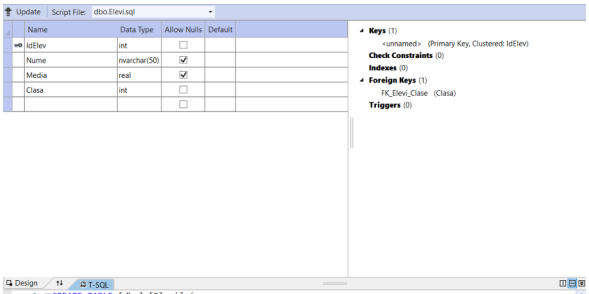


In mod similar procedam si pt Tabelul elevi cu structura:

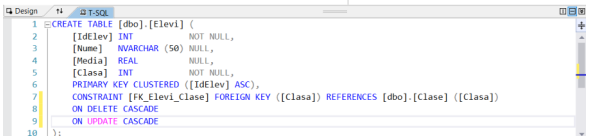


Se executa click dreapta pe Foreign Keys si dupa se completeaza linia corespunzatoare

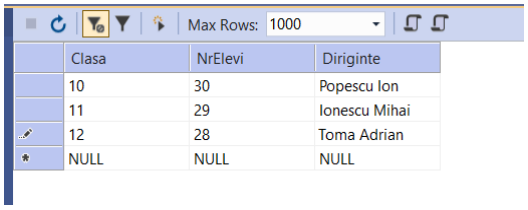
Relatia va fi de tipul 1:n (pentru o clasa din tabelul Clase vom avea mai multe inregistrari in tabelul Elevi).



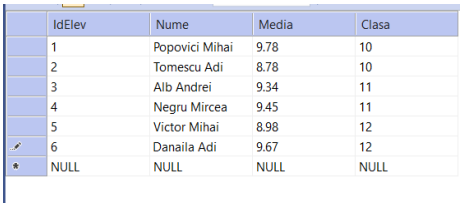
De asemenea se completeaza liniile pentru stergere si actualizare in cascada



2. Popularea tabelelor

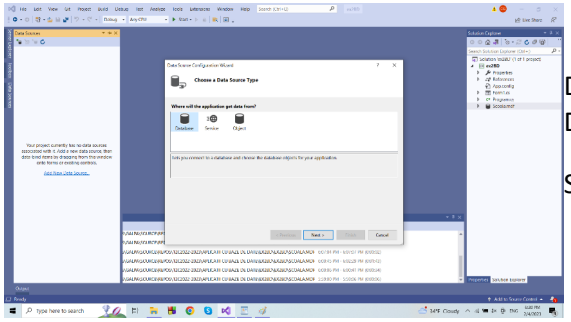


Clasa	NrElevi	Diriginte
10	30	Popescu Ion
11	29	Ionescu Mihai
12	28	Toma Adrian
NULL	NULL	NULL



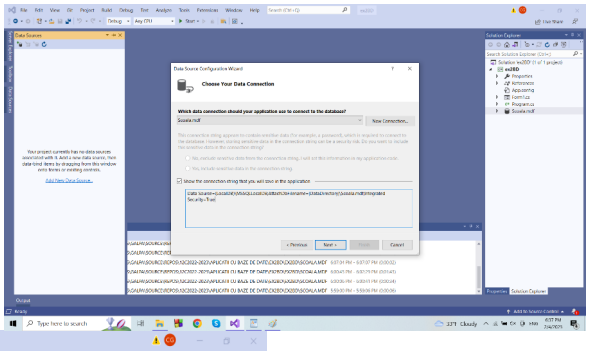
IdElev	Nume	Media	Clasa
1	Popovici Mihai	9.78	10
2	Tomescu Adi	8.78	10
3	Alb Andrei	9.34	11
4	Negru Mircea	9.45	11
5	Victor Mihai	8.98	12
6	Danaila Adi	9.67	12
NULL	NULL	NULL	NULL

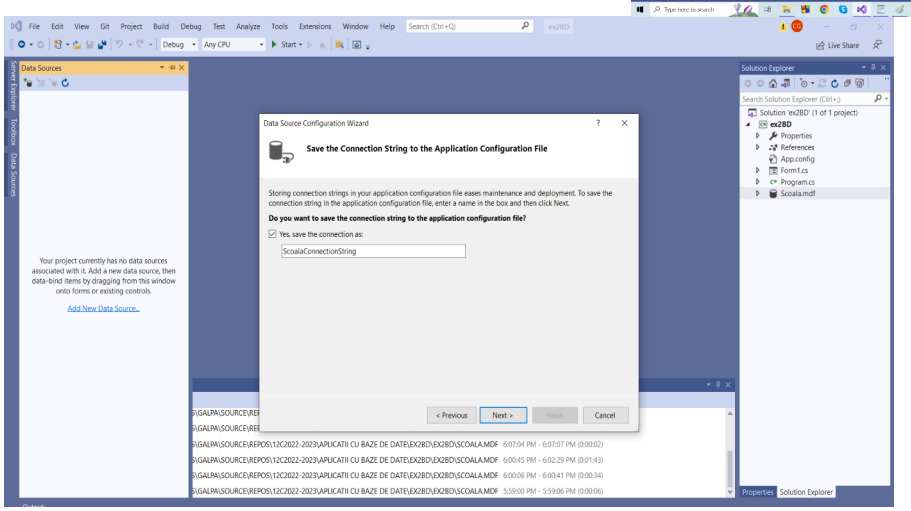
3. Stabilirea relației



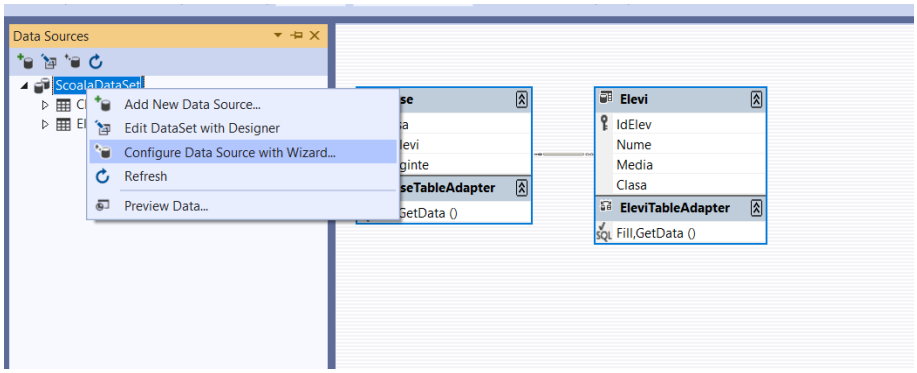
Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\Scoala.mdf;Integrated

Security=True



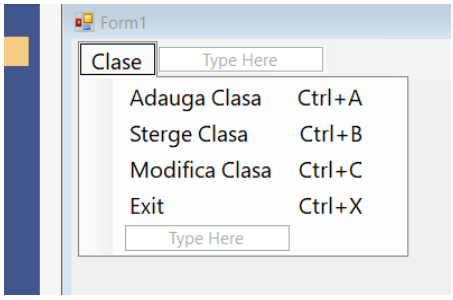


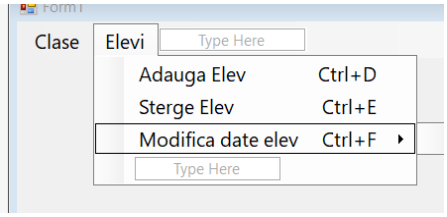
Se folosește Wizard -> apar cele două tabele legate

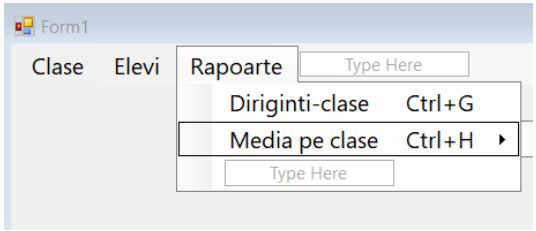


4. Crearea unui meniu

Adăugăm pe suprafața formularului un control MenuStrip cu opțiunile(cu proprietatea ShortCutKeys modificată):

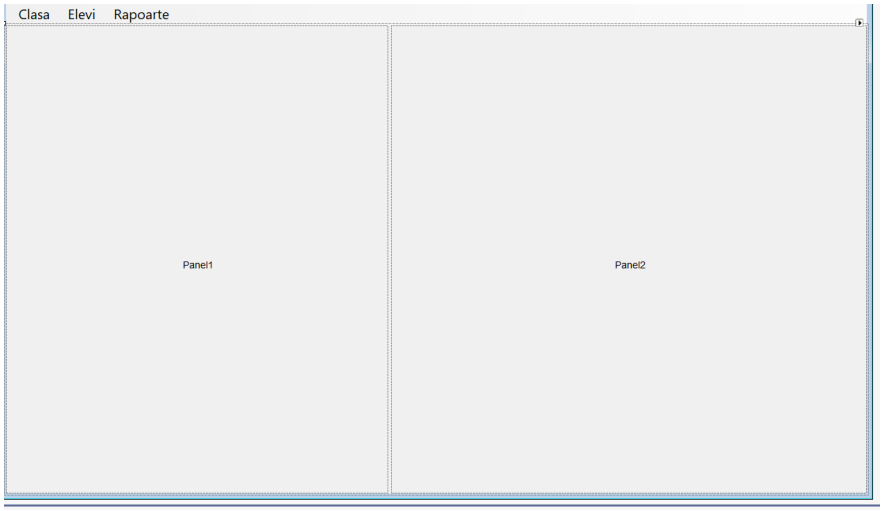




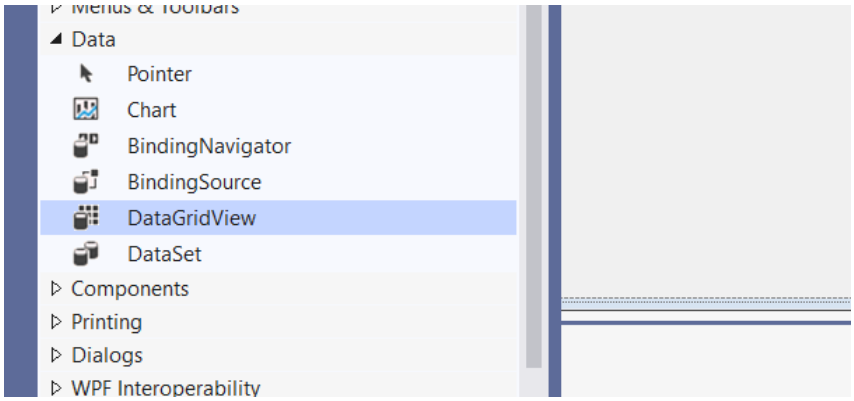


5. Încărcarea tabelelor în DataGridView

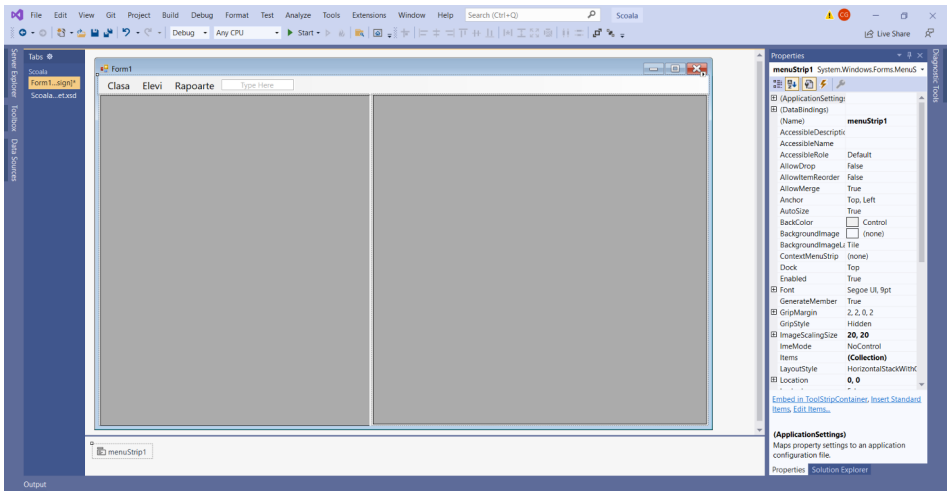
Adaugam pe suprafata Formularului un control de tipul SplitContainer:



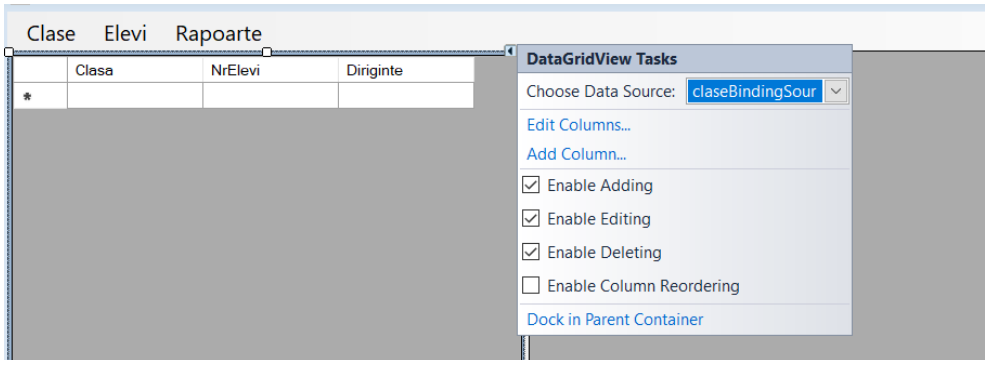
In fiecare din cele doua panouri dispunem un control de tipul DataGridView:



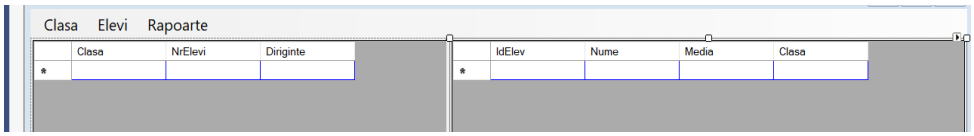
Prof. Carmen Gal



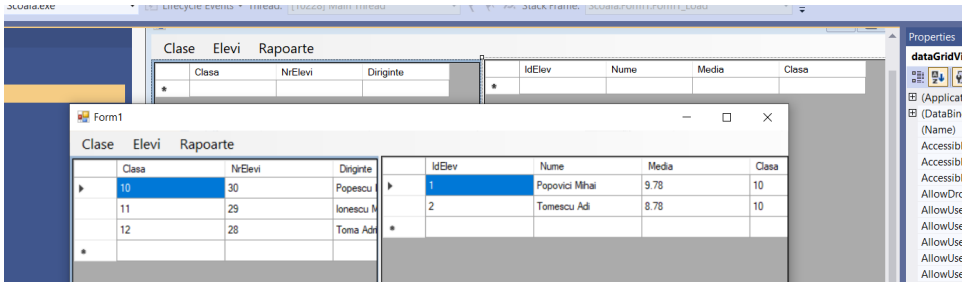
Adaugarea tabelului Clase:



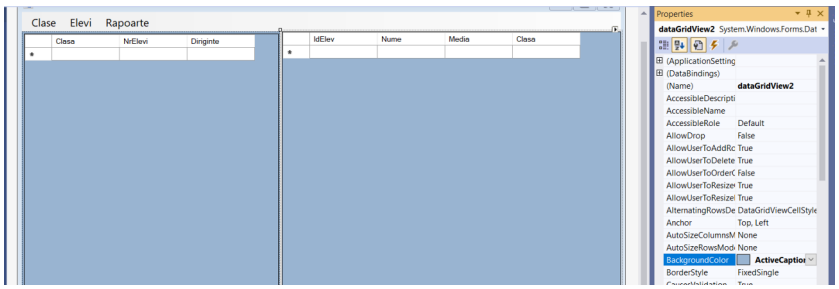
Se adauga si tabelul elevi

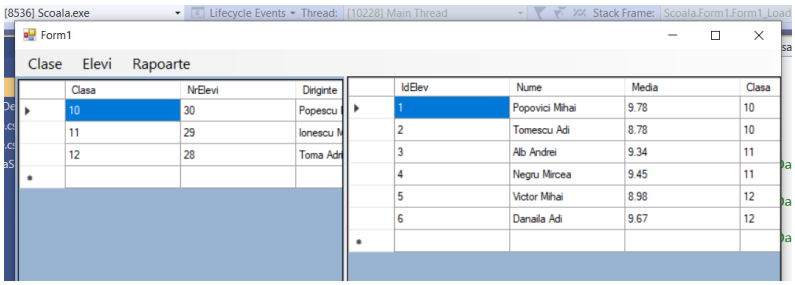


La rulare o să apară:



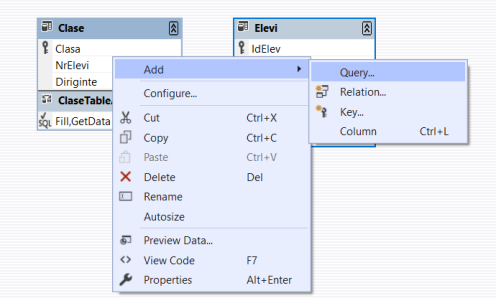
Am modificat culoarea pentru DataGridView:

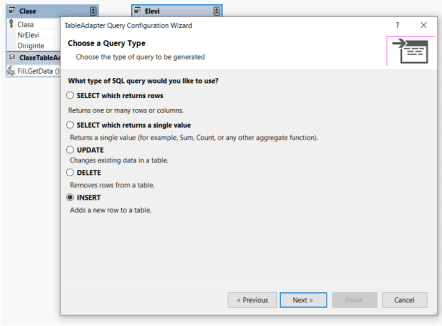


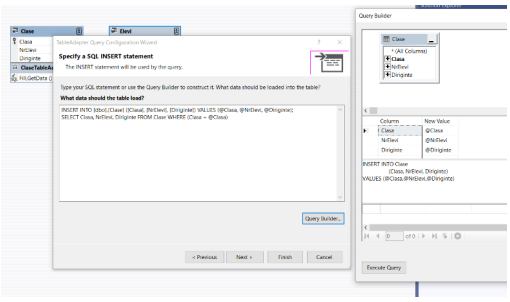


6. Interogări. Proceduri stocate

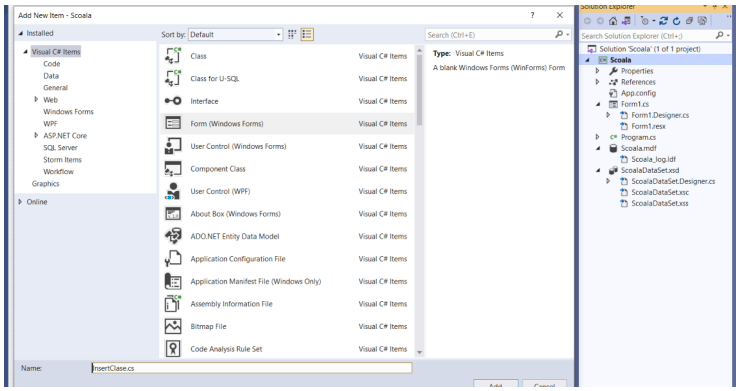
In Solution Explorer executăm dublu click pe ScoalaDataSet, iar in DataSet Designer executăm click dreapta pe tabelul Clase si Add Query:

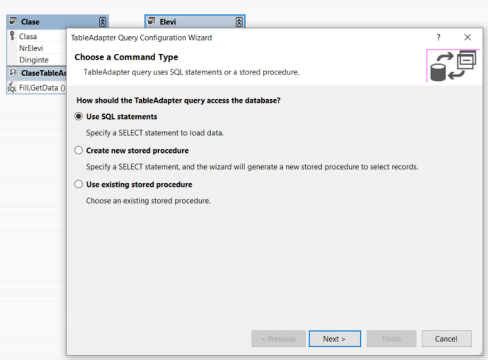




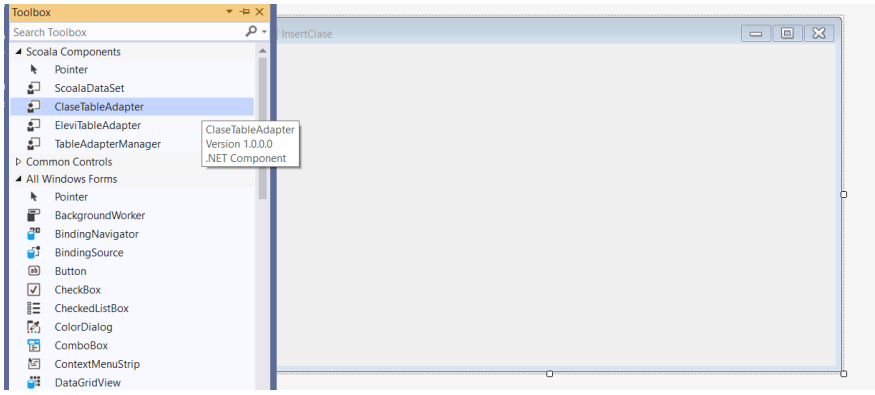


Vom adauga la proiectul nostru un nou formular pentru introducerea datelor in tabelul clase. Pentru aceasta in SolutionExplorer executam click dreapta pe Add-> New Item->Windows Form:

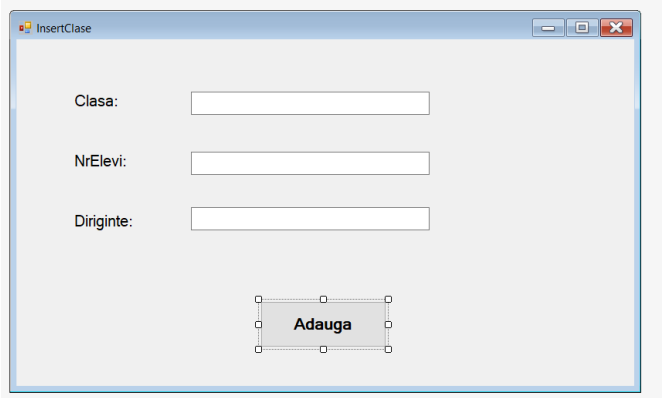




Din ToolBox selectam ClaseTableAdaptor pe care il dispunem pe suprafata formularului:



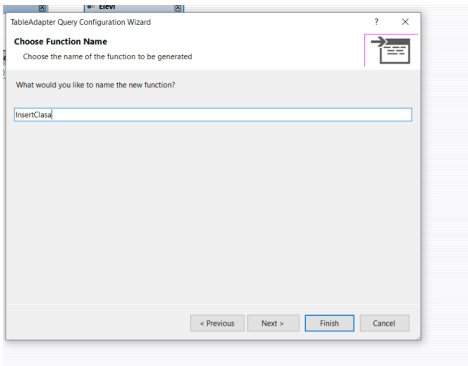
Pe suprafata noului formular dispunem 3 etichete, 3 casete de text si un buton:



Adaugam cod la evenimentul Click al butonului Adauga:

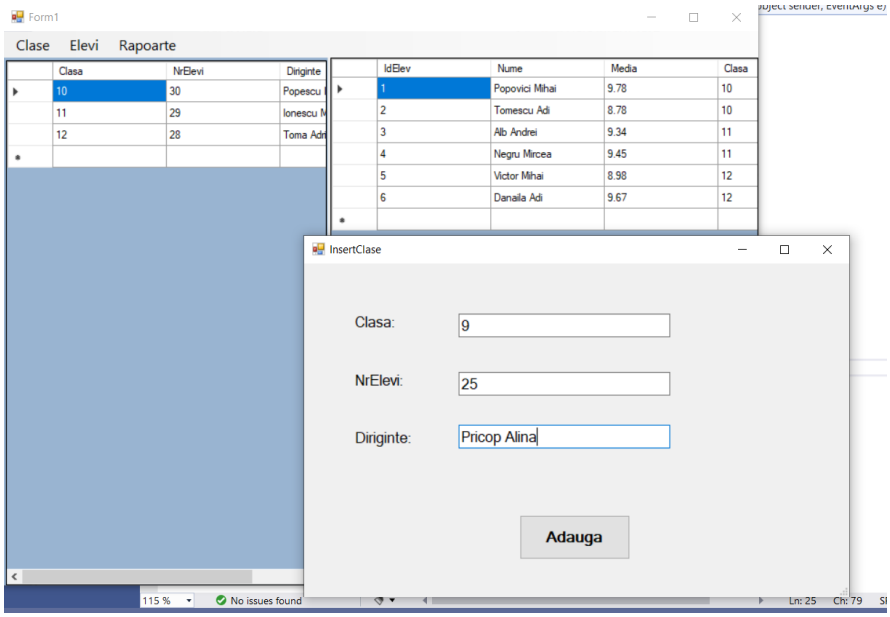
```
namespace Scoala
{
    4 references
    public partial class InsertClase : Form
    {
        1 reference
        public InsertClase()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "" && textBox2.Text != "" &&
                textBox3.Text != "")
            {
                claseTableAdapter1.InsertClasa(Convert.ToInt32(textBox1.Text),
                    Convert.ToInt32(textBox2.Text), textBox3.Text);
                this.Close();
            }
            else MessageBox.Show("Completati informatiile!");
        }
    }
}
```

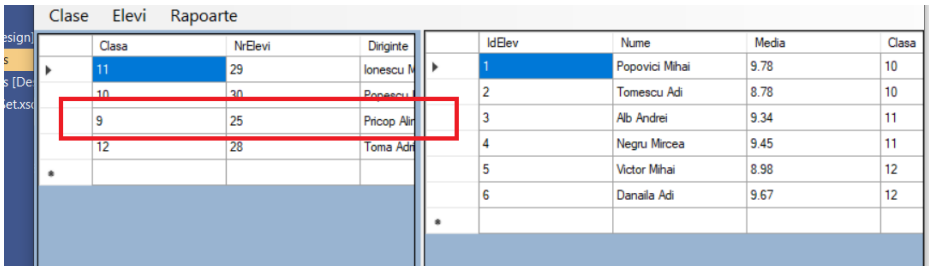
Acum trebuie sa facem legatura dintre optiunea din meniu Adauga Clasa (dublu click pentru a deschide evenimentul asociat)si lansarea in executie a **formularului InsertClasa**:



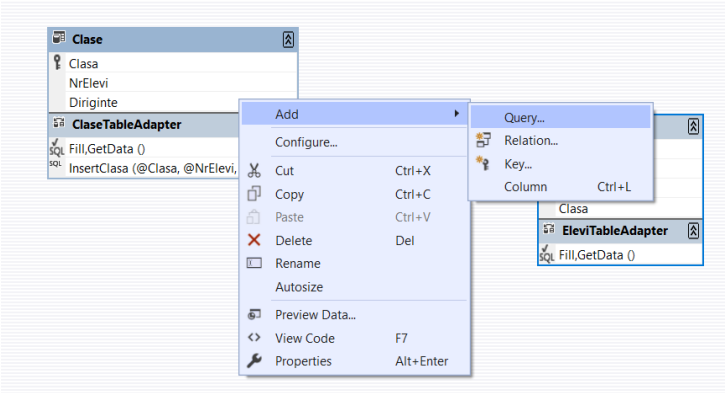

```
private void adaugaClasaToolStripMenuItem_Click(object sender, EventArgs e)

        {

            InsertClase f1 = new InsertClase();

            f1.ShowDialog();

            this.claseTableAdapter.Fill(this.scoalaDataSet.Clase);

        }
```

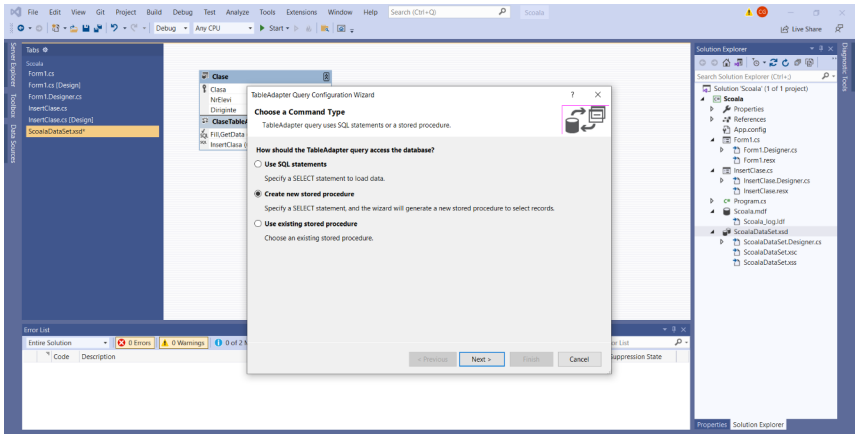
Prof. Carmen Gal

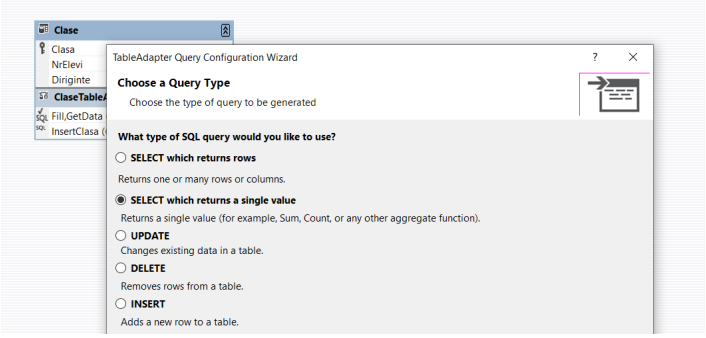


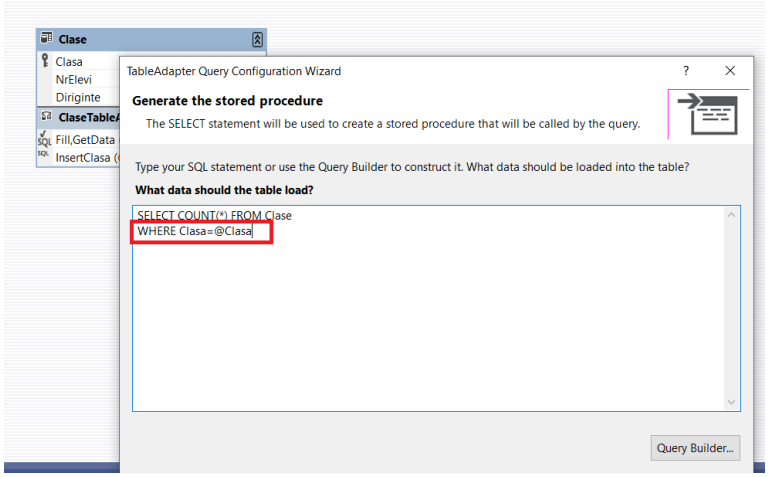


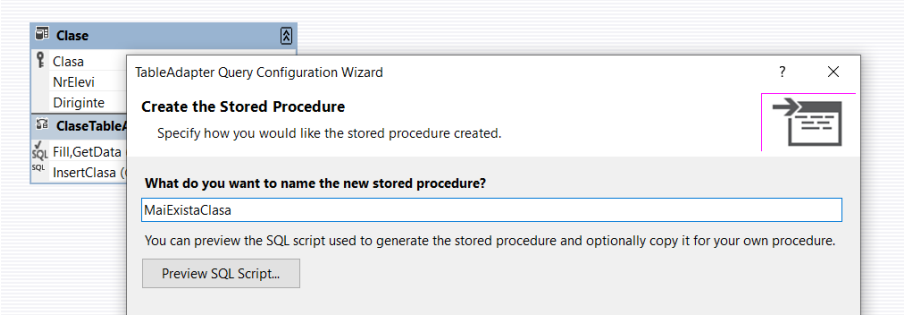
Generarea procedurilor stocate:

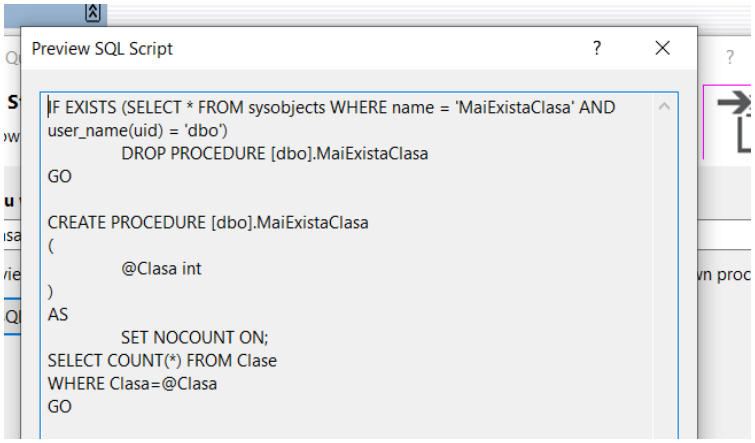








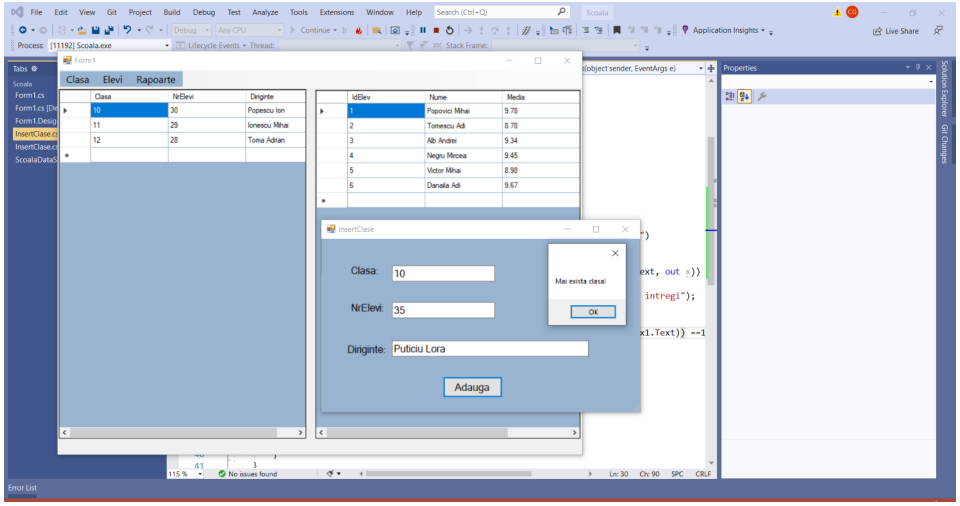


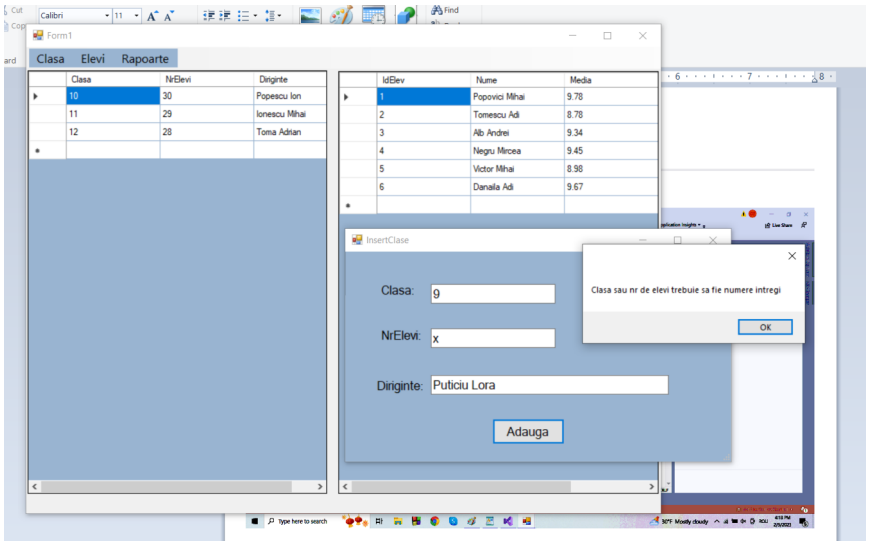


Modificam codul din **formularului InsertClasa astfel încât să se verifice dacă clasa a mai fost sau nu adăugată sau dacă s-au introdus valori intregi pt clasa si nr de elevi:**

```
4 references
public partial class InsertClase : Form
{
    1 reference
    public InsertClase()
    {
        InitializeComponent();
    }

    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        if (textBox1.Text != "" && textBox2.Text != "" && textBox2.Text != "")
        {
            int x=0;
            if(!int.TryParse(textBox1.Text,out x)|| !int.TryParse(textBox2.Text, out x))
            {
                MessageBox.Show("Clasa sau nr de elevi trebuie sa fie numere intregi");
                return;
            }
            if((int)claseTableAdapter1.MaiExistaClasa(Convert.ToInt32(textBox1.Text)) ==1)
            {
                MessageBox.Show("Mai exista clasa!");
                return;
            }
            claseTableAdapter1.InsertClasa(Convert.ToInt32(textBox1.Text),
               Convert.ToInt32(textBox2.Text), textBox3.Text);
            this.Close();
        }
        else MessageBox.Show("Completati campurile");
    }
}
}
```

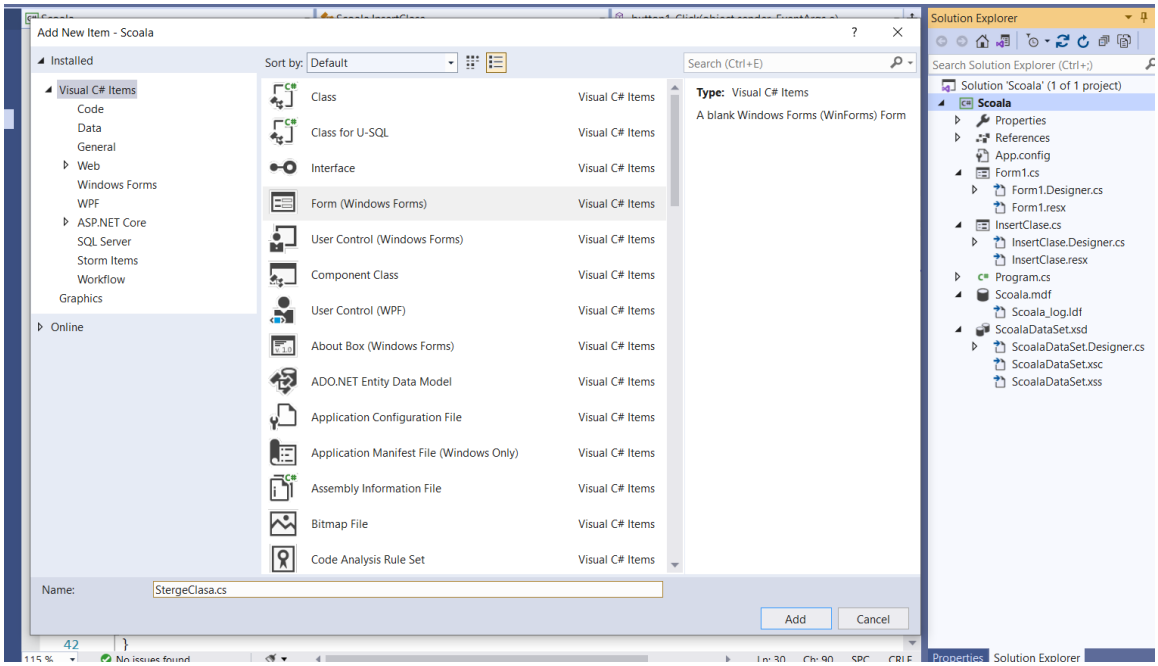
De data aceasta la rularea aplicației vom avea verificările:



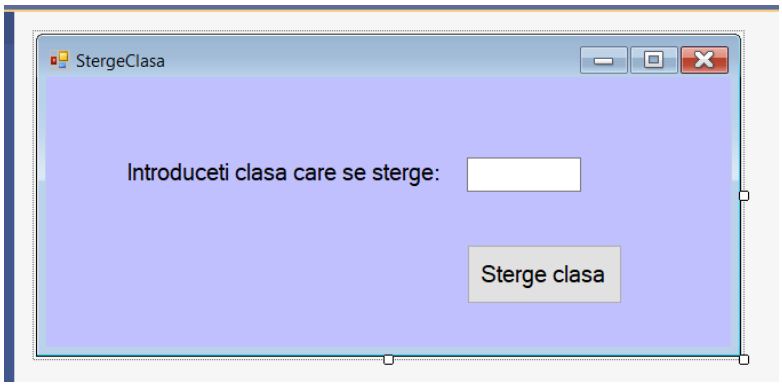


Stergerea unei clase:

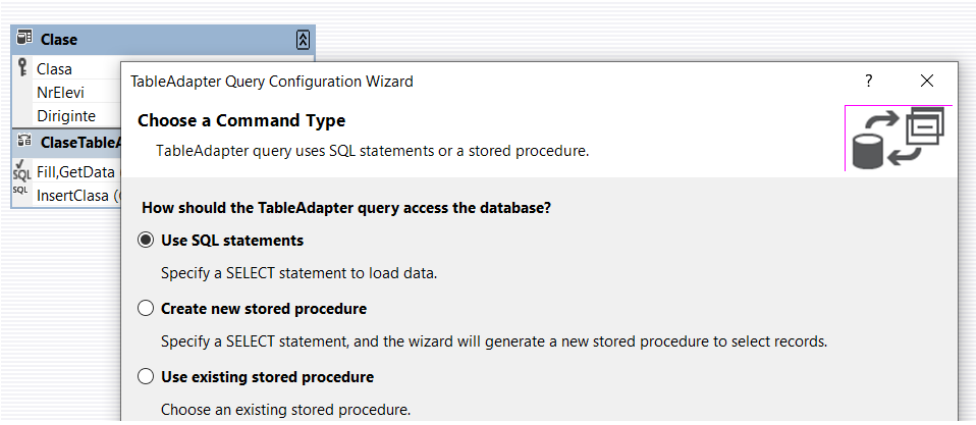
Crearea unui nou formular denumit StergeClasa:

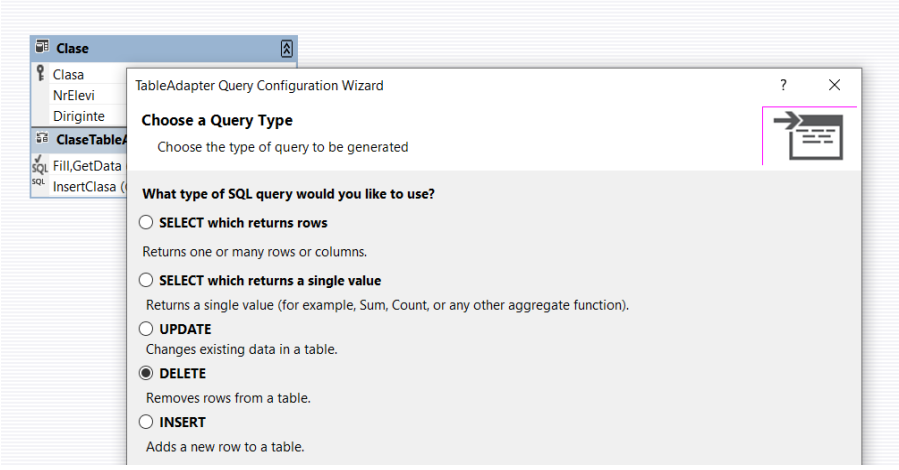


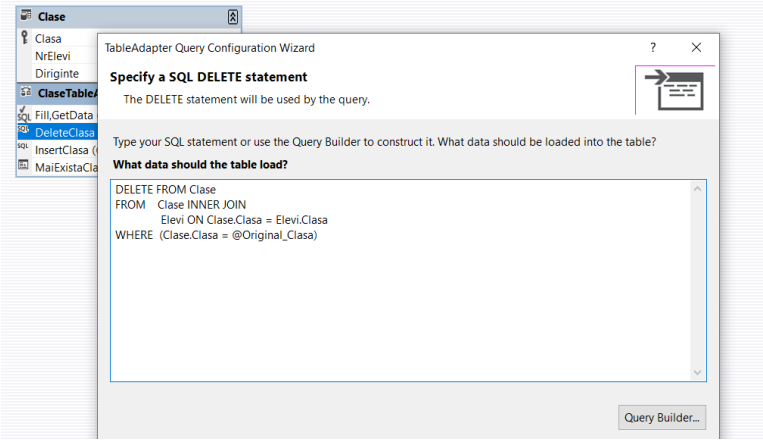
Adaugarea elementelor de interfata:

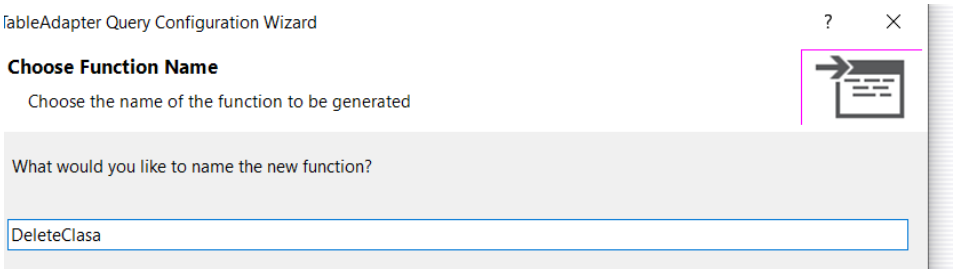


Vom crea în continuare interogarea pentru ștergerea clasei și a rândurilor corespunzătoare din tabela Elevi:

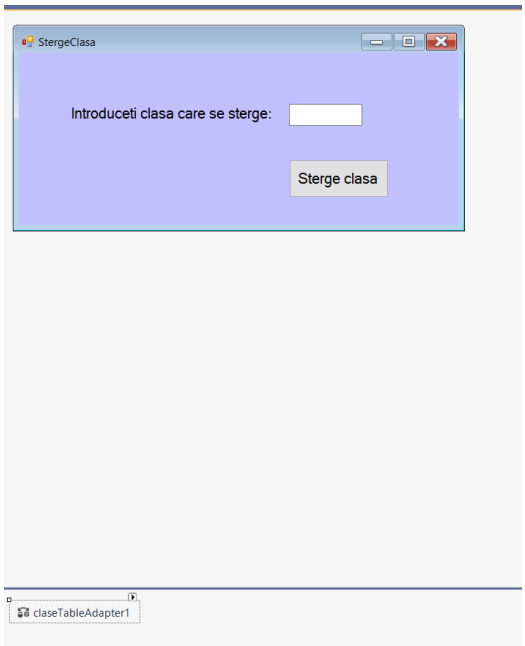








Punem un Control de tipul ClasaTableAdapter pe suprafata formularului StergeClasa:



Introducem codul la evenimentul Click asociat butonului în formularul de ștergere:

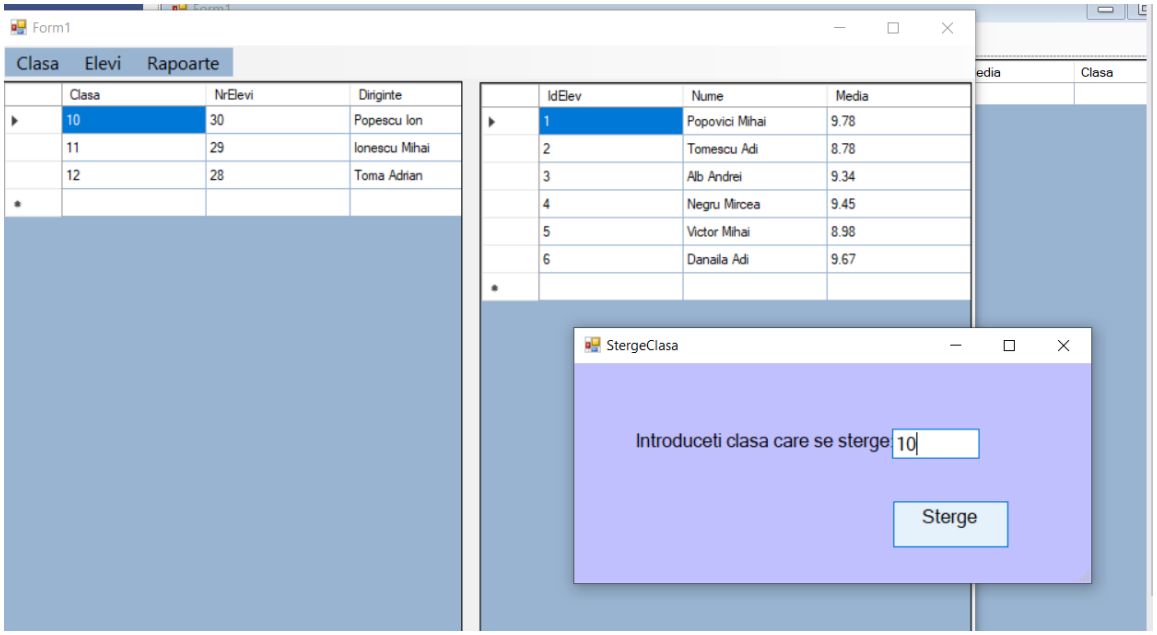
```
 private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "")
            {
                int cl = 0;
                if (int.TryParse(textBox1.Text, out cl))
                {
                    int x= Convert.ToInt32(textBox1.Text);
                    if ((int)this.claseTableAdapter1.MaiExistaClasa(x) == 1)
                        claseTableAdapter1.DeleteClasa(x);
                    else
                        MessageBox.Show("Nu exista clasa!");
                    Close();
                }
                else
                    MessageBox.Show("Introduceti o valoare numerica");
            }
            else
                MessageBox.Show("Completati caseta de text cu o valoare numerica");
        }
```

In meniul formularului principal Form1 la optiunea StergeClasa introducem codul pt stergerea unei clase:

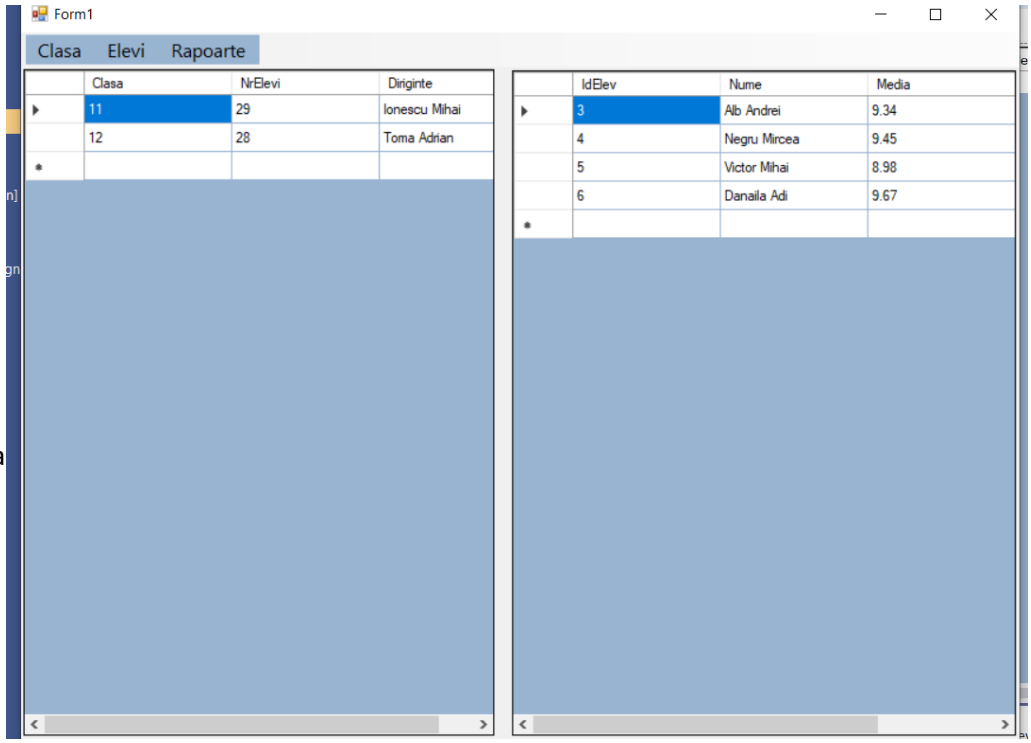
```
private void stergeClasaToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
            StergeClasa f2 = new StergeClasa();//instantiarea unui formular de tipul StergeClasa
            f2.ShowDialog();
            this.claseTableAdapter.Fill(this.scoalaDataSet.Clase);
            this.eleviTableAdapter.Fill(this.scoalaDataSet.Elevi);
        }
```

Efectul:



Pentru a ModificaClasa vom crea o noua interogare:

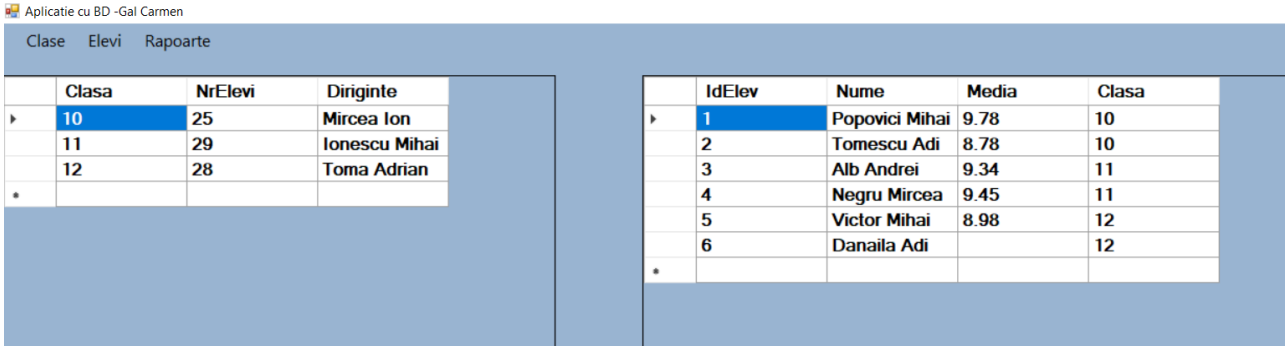


În continuare vom crea formularul ModificaClasa:

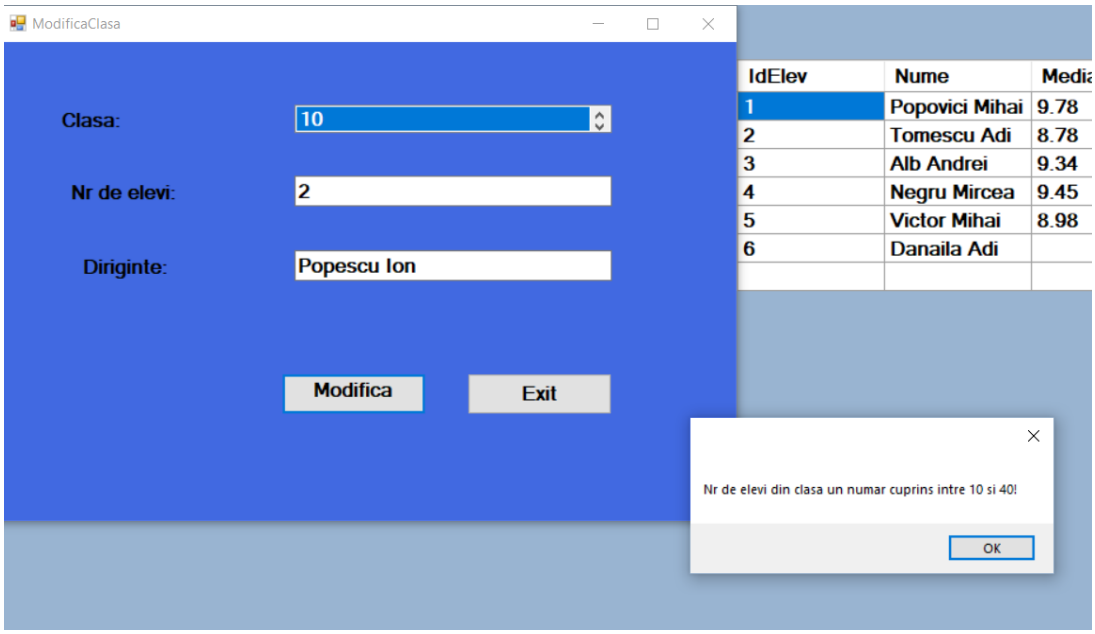
Pe suprafata formularului dispunem 3 etichete, un ListBox si doua butoane, asa cum rezulta si din figura de mai sus.

La apasarea butonului Modifica se opereaza modificarile in baza de date

care vor fi vizibile in tabelul Clasa la închiderea formularului Modifică Clasa (s-a modificat numărul de elevi și dirigintele la clasa a 10-a.



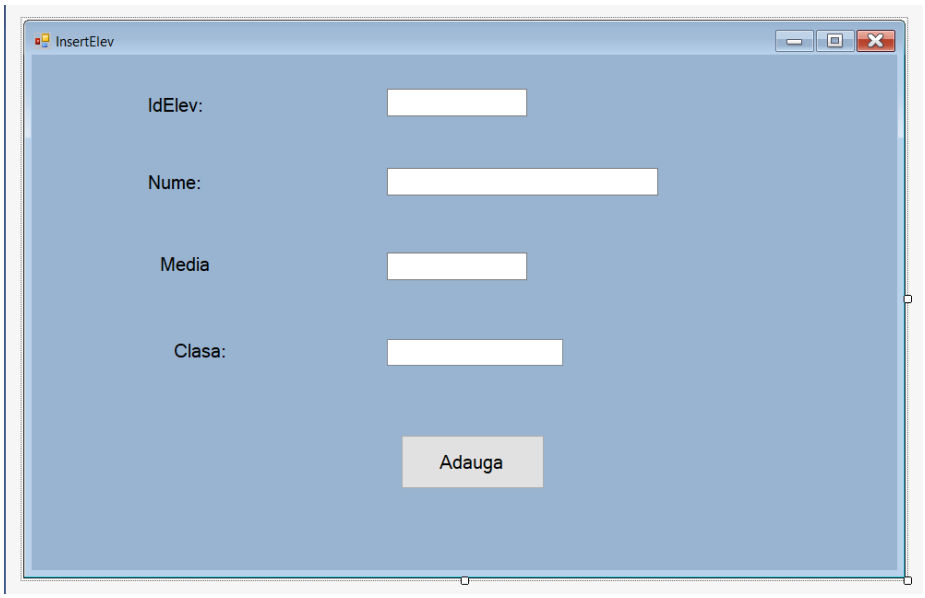
La selectarea din listă a mai multor clase pot fi operate mai multe modificări în tabela Clasa.

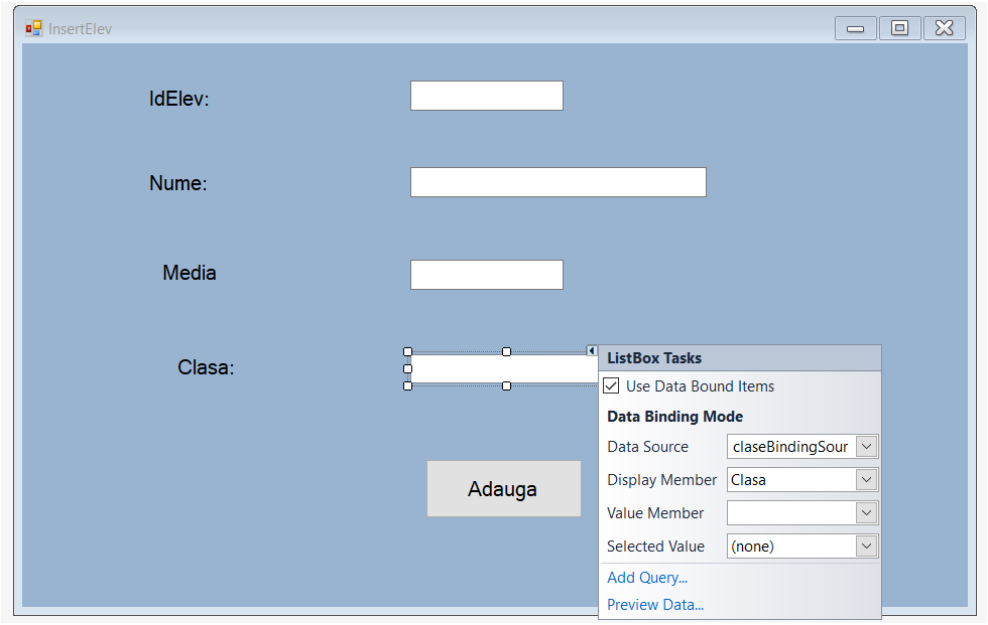


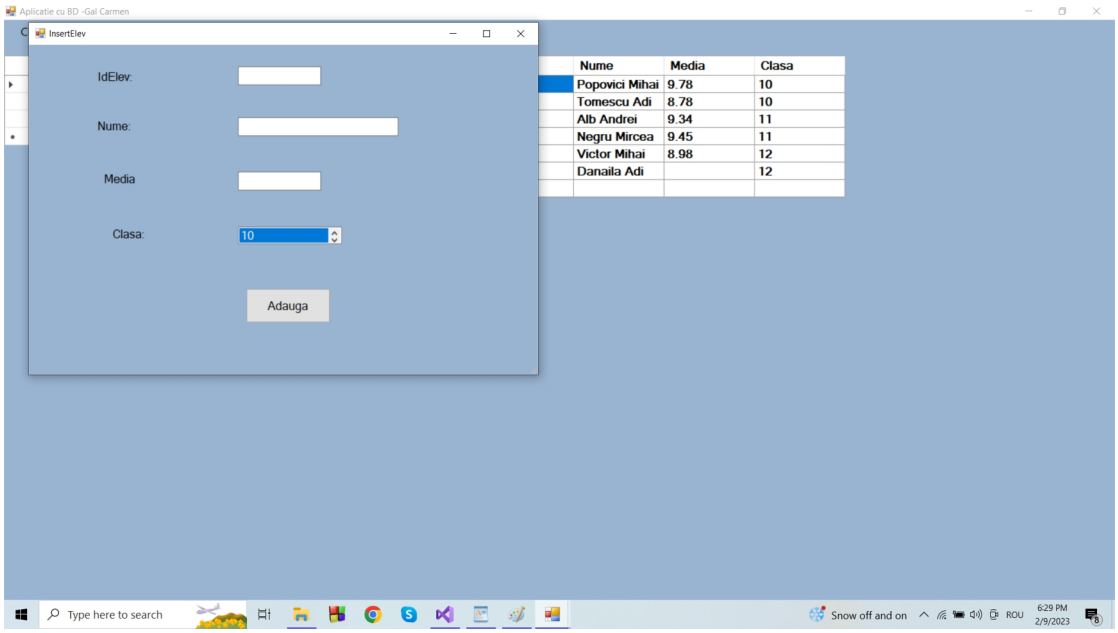

```
private void button1_Click(object sender, EventArgs e)
{
    claseBindingSource.EndEdit();
    if (int.TryParse(textBox1.Text,out int y) &&
        (Convert.ToInt32(textBox1.Text)>=10 && Convert.ToInt32(textBox1.Text)<=40))
    {
        int x = claseTableAdapter.Update(scoalaDataSet);
        this.claseTableAdapter.Fill(this.scoalaDataSet.Clase);
    }
   else
    if (!(Convert.ToInt32(textBox1.Text) >= 10 && Convert.ToInt32(textBox1.Text) <= 40))
        MessageBox.Show("Nr de elevi din clasa un numar cuprins intre 10 si 40!");
}
```

Adaugă elev:

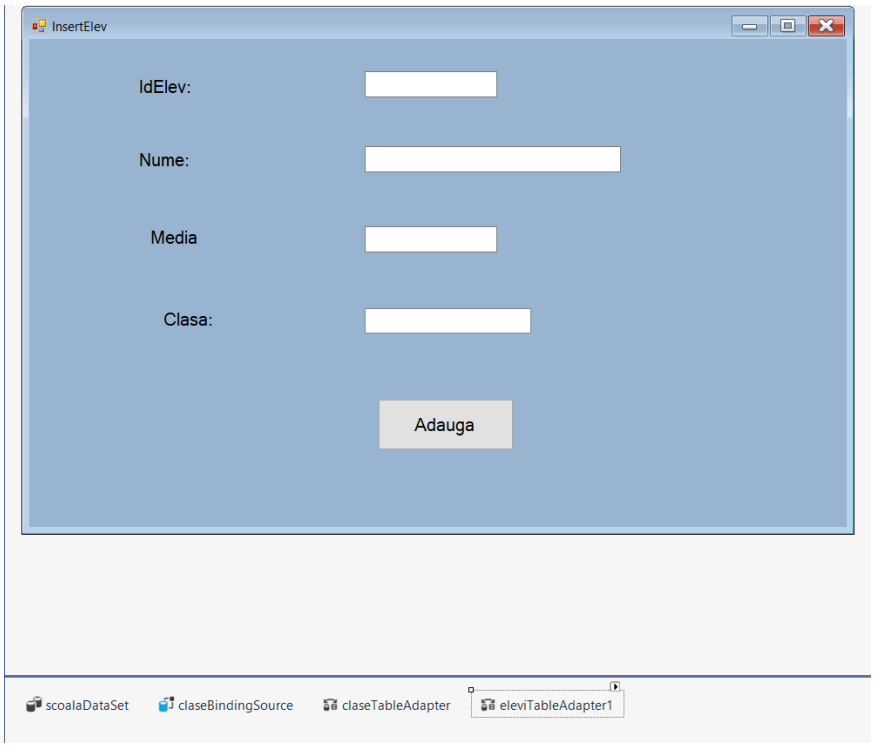
Se crează un nou formular InsertElev, la Clasa se adaugă o lista care fi legată de câmpul Clasa din tabelul Clase (practic preluăm din câmpul Clasa valorile pentru a evita introducerea datelor eronate), restul fiind casete de texte simple care nu sunt legate de câmpurile din tabel :



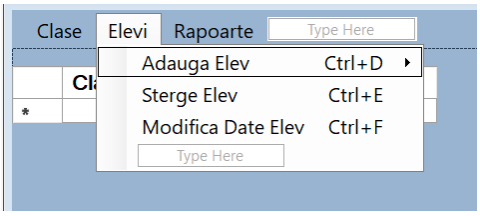




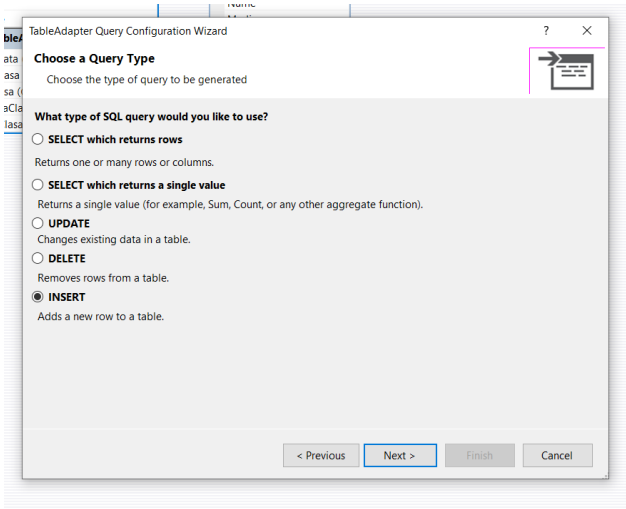
De asemenea, pe suprafata formularului trebuie sa avem următoarele Controale (adăugate automat sau din ToolBox):

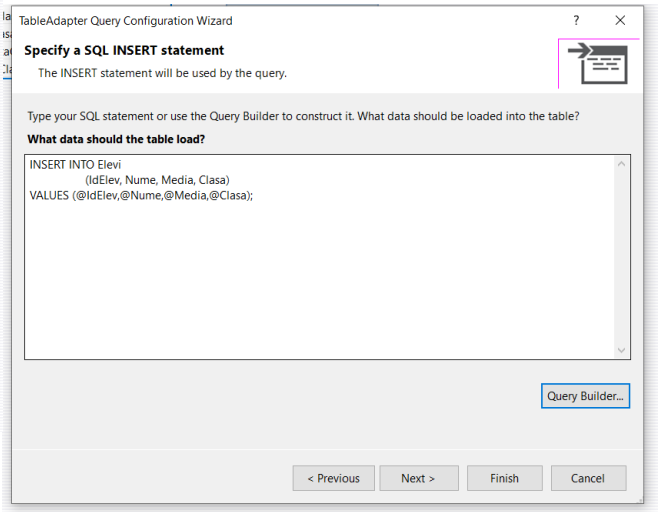


În formularul principal se adaugă codul în meniu care face legătura cu formularul InsertElev la opțiunea AdaugaElev din meniu:

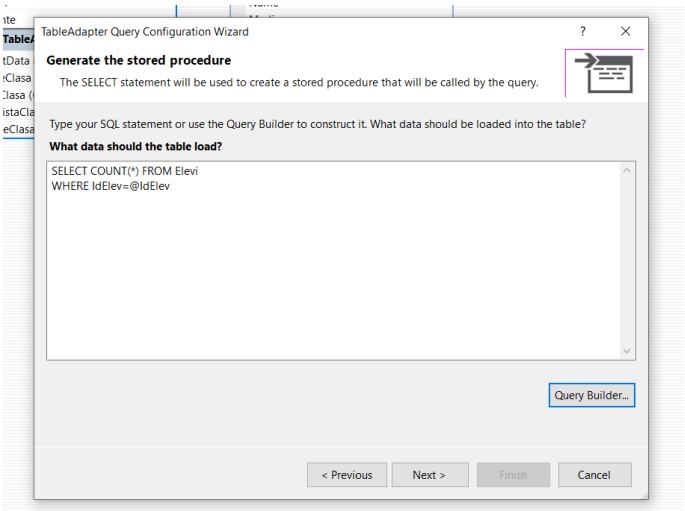


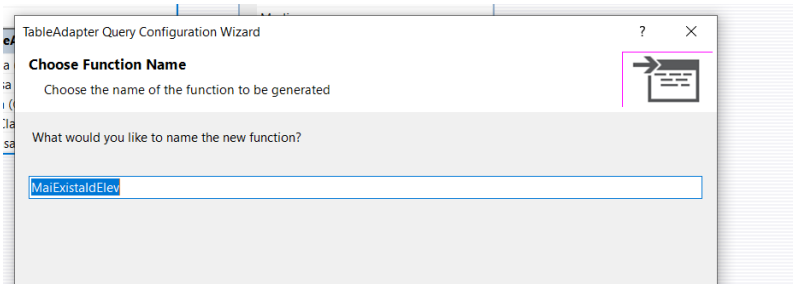
Vom crea funcția InsertElev în ScoalaDataSet:





Vom crea și o procedură pentru a verifica dacă elevul cu acel Id exista sau nu:

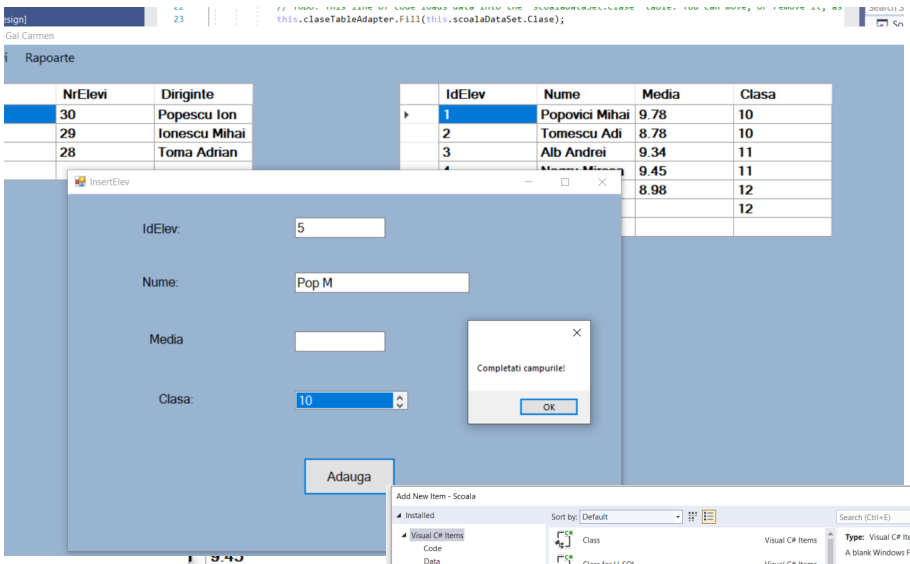




Adaugăm cod la evenimentul Click al butonului din formularul InsertElev:

Am verificat să fie completate toate câmpurile, să nu se introducă un ID care deja există în baza de date.

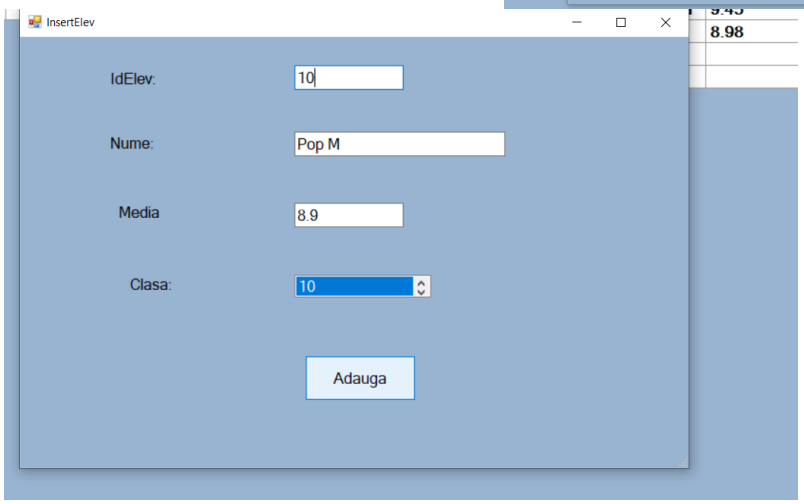
La rulare se obține:

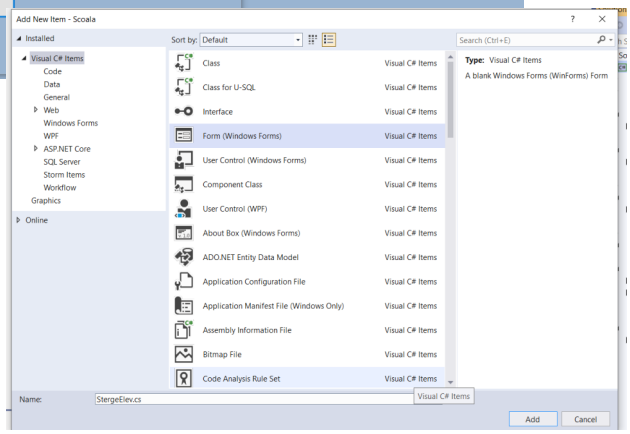


Se completează datele în tabel:

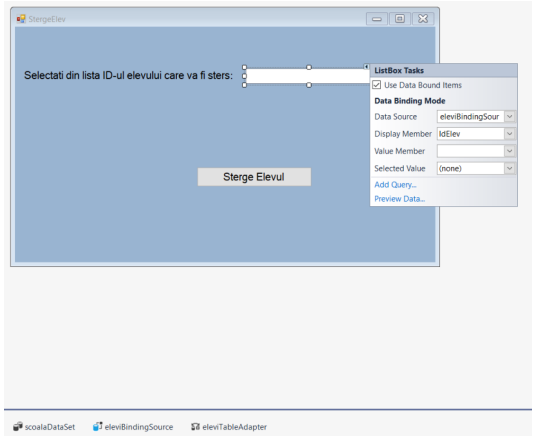
Ne vom ocupa în continuare de stergerea unui elev.

Vom crea un nou formuar StergeElev:

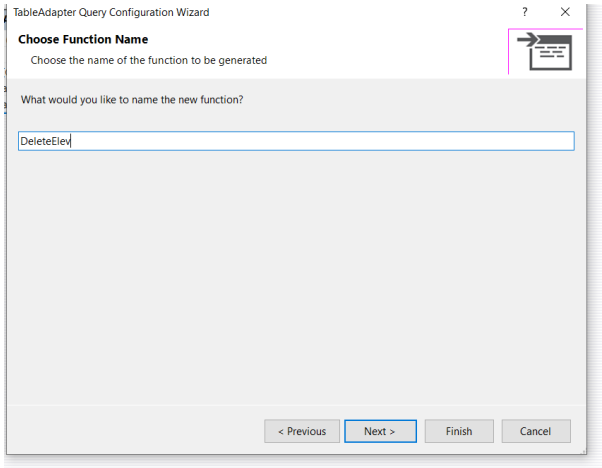


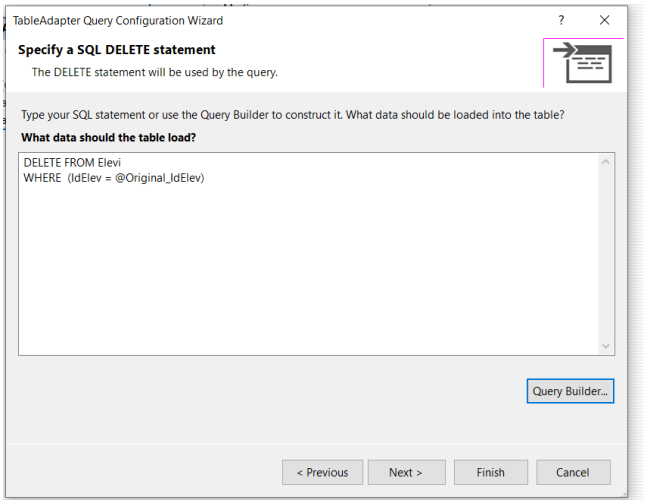


Elementele de interfață:



Se crează interogarea de ștergere:





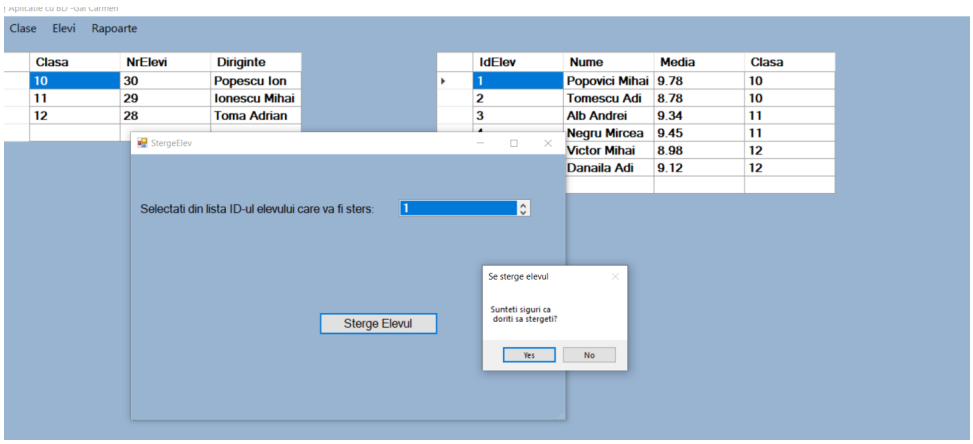
In formularul principal completam codul la optiunea din meniu:

```
private void stergeElevToolStripMenuItem_Click(object sender, EventArgs e)
{
    StergeElev f1 = new StergeElev();
    f1.ShowDialog();
    this.eleviTableAdapter.Fill(this.scoalaDataSet.Elevi);
}
```

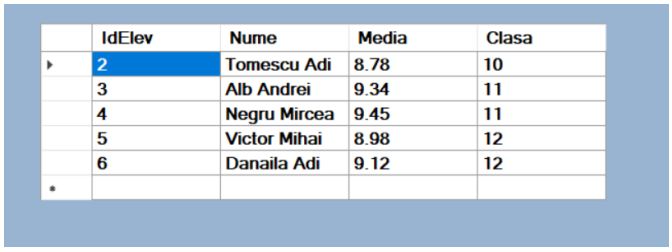
Apoi introducem codul la evenimentul Click al butonului de pe formularul StergeElev:

```
private void button1_Click(object sender, EventArgs e)
{
    DialogResult d = MessageBox.Show("Sunteti siguri ca \n doriti sa stergeti?", "Se sterge elevul", MessageBoxButtons.YesNo);
    if(d==DialogResult.Yes)
        this.eleviTableAdapter.DeleteElev(Convert.ToInt32(listBox1.GetItemText(listBox1.SelectedItem)));
    this.Close();
}
```

Am creat o fereastra de dialog in care suntem intrebati daca suntem siguri ca dorim operatia de stergere:



Efectul - va fi sters elevul cu ID-ul 1:



IdElev	Nume	Media	Clasa
2	Tomescu Adi	8.78	10
3	Alb Andrei	9.34	11
4	Negru Mircea	9.45	11
5	Victor Mihai	8.98	12
6	Danaila Adi	9.12	12

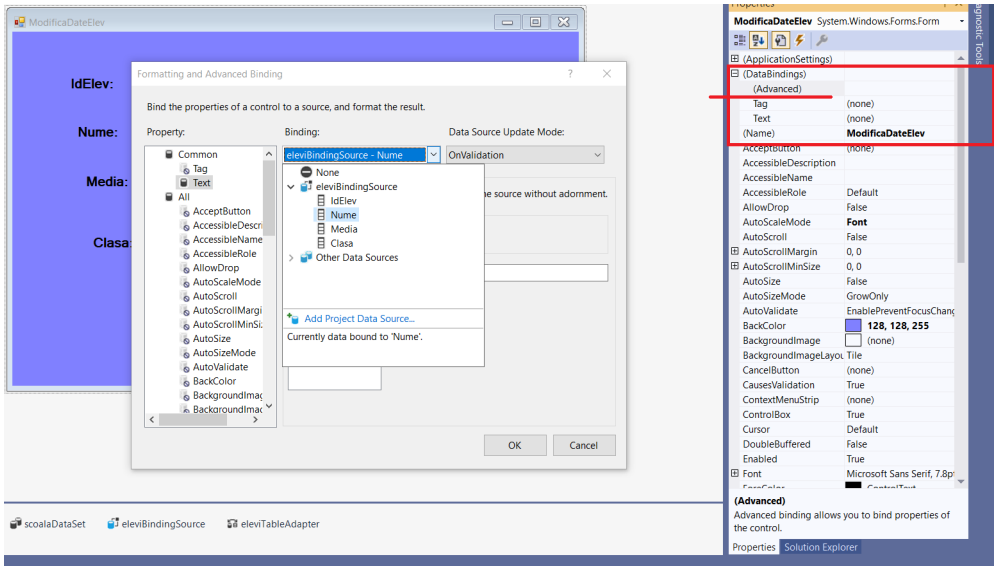
Dacă se selectează No, atunci elevul nu va fi șters.

Creăm un nou formular: ModificăDateElev:

De data aceasta vom insera doua liste pe suprafata formularului care vor contine IdElev si clasa valori ce nu pot fi modificate, doar selectate si 2 casete de text pentru Nume și Media.

Se leagă și casetele de text de câmpurile Nume și Media ale tabelului Elev din fereastra Properties- DataBindings - Advanced si apoi din fereastra care apare se selectează câmpul corespunzător:

Completăm codul în formularul principal la opțiunea din meniu asociată:




```
private void modificaDateElevToolStripMenuItem_Click(object sender, EventArgs e)
{
    ModificaDateElev f3 = new ModificaDateElev();
    f3.ShowDialog();
    this.eleviTableAdapter.Fill(this.scoalaDataSet.Elevi);
}
```

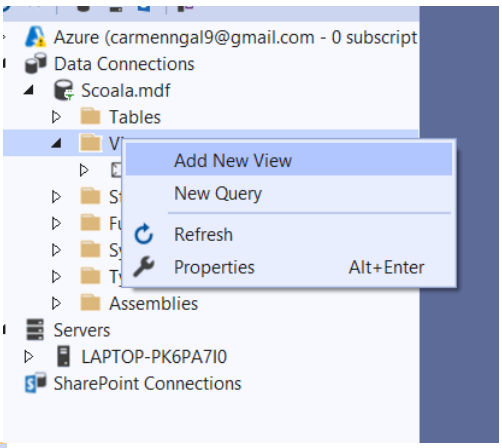
Apoi codul din formularul ModificaDateElev:

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    eleviBindingSource.EndEdit();
    if (double.TryParse(textBox2.Text, out double y))
    {
        if (Convert.ToDouble(textBox2.Text) >= 1 && Convert.ToDouble(textBox2.Text) <= 10)
        {
            int x = eleviTableAdapter.Update(scoalaDataSet);
            this.eleviTableAdapter.Fill(this.scoalaDataSet.Elevi);
        }
        else
         MessageBox.Show("Media e cuprinsa intre 1 si 10!");
    }
    else MessageBox.Show("Media este un numar real!");



}

1 reference
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

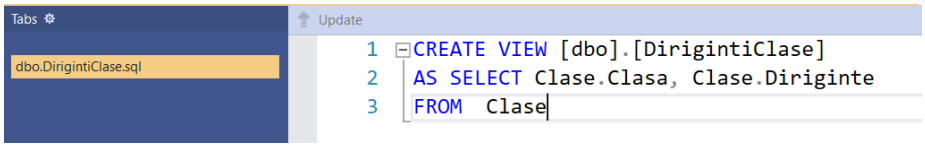
Efectul este:

IdElev	Nume	Media	Clasa
1	Popovici Mihai	8.9	10
2	Tomescu Adi	8.78	10
3	Alb Andrei	9.34	11
4	Negru Mircea	9.45	11
5	Victor Mihai	8.98	12
6	Danaila Adi	9.12	12

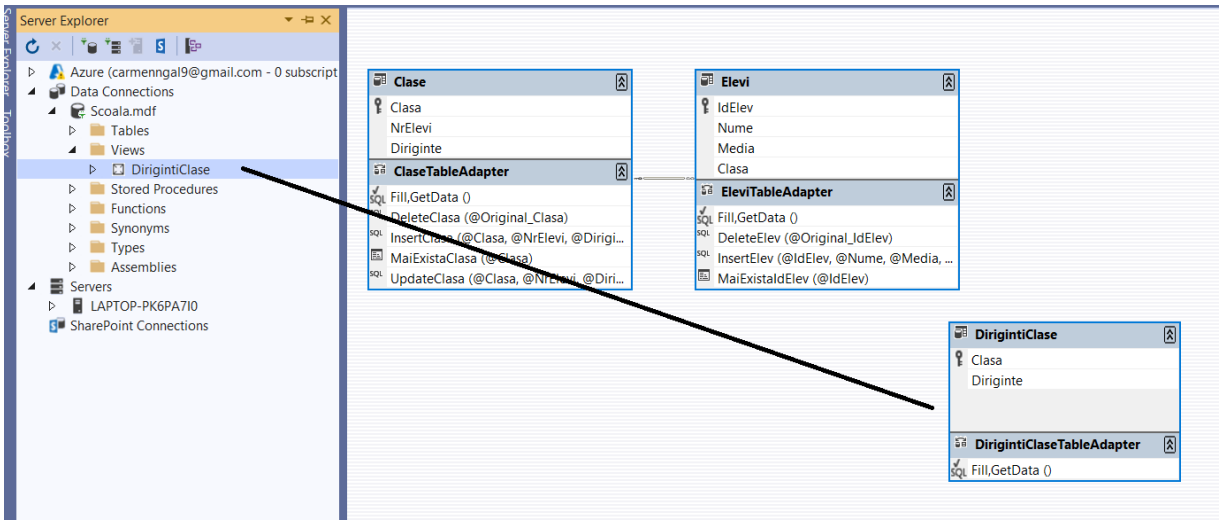
Pentru a construi raportul Diriginti-Clase vom crea un View (o vedere):



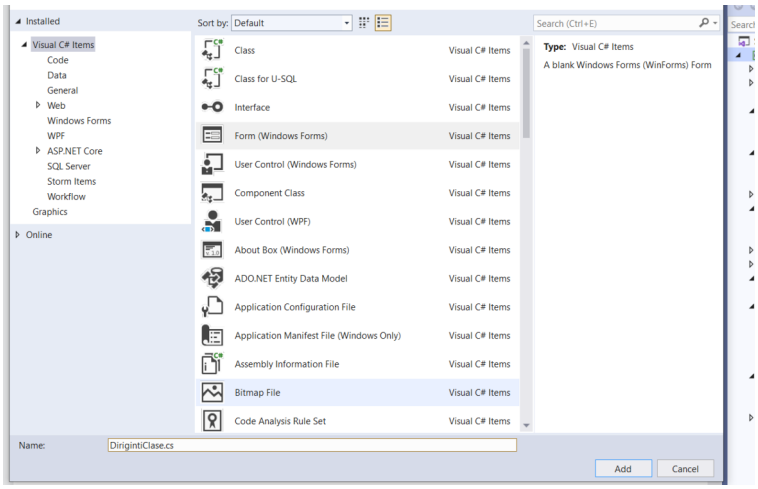
Se scrie codul corespunzator vederii:



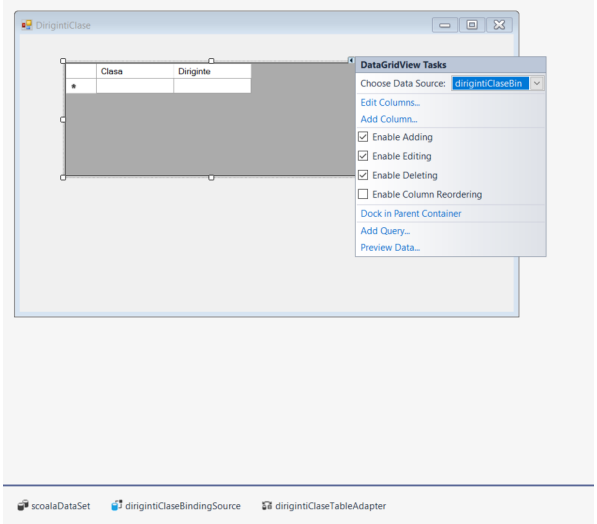
Se actualizeaza DataSet. Prin drag-and-drop se trage vederea in DataSet:



Apoi se creaza un nou Formular:



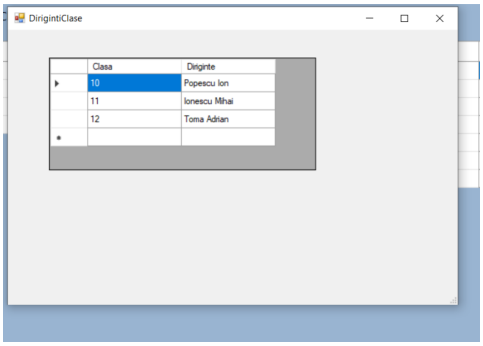
Pe suprafata formularului se adauga un DataGridView, iar la sursa de date se adauga vederea nou creata:



7. Rapoarte

In formularul principal scriem codul la optiunea din meniul Rapoarte - Diriginti-Clase:

```
 private void diriginticlaseToolStripMenuItem_Click(object sender,
EventArgs e)

        {

            DirigintiClase f1 = new DirigintiClase();

            f1.ShowDialog();

        }
```



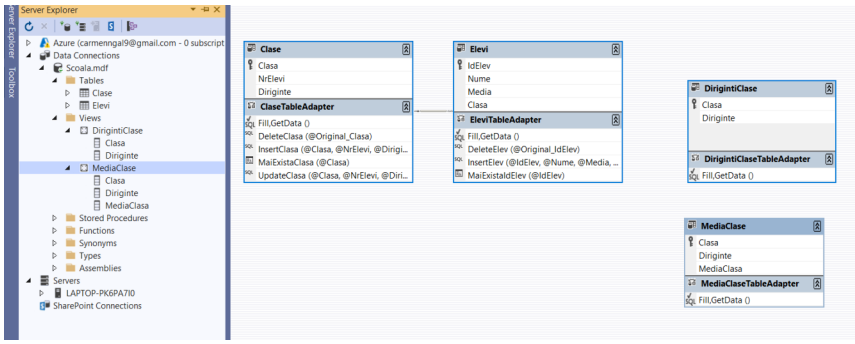
La rulare si selectarea optiunii din meniu se obtine:

Pentru a crea raportul Media pe Clase procedam astfel:

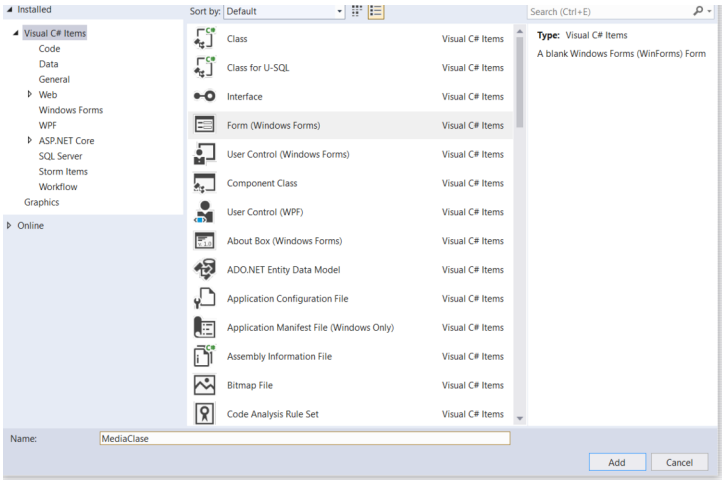
Se creaza o noua vedere:

```sql
CREATE VIEW [dbo].[MediaClase]
    AS SELECT Clase.Clasa, Clase.Diriginte, AVG(Elevi.Media) AS MediaClasa
FROM Clase INNER JOIN Elevi ON Clase.Clasa = Elevi.Clasa
GROUP BY Clase.Clasa, Clase.Diriginte
```

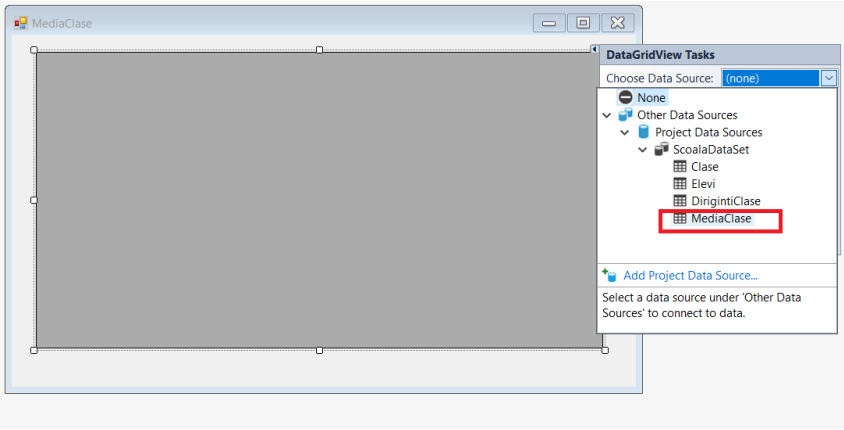
Se adaugă vederea în DataSet prin Drag-and-Drop:

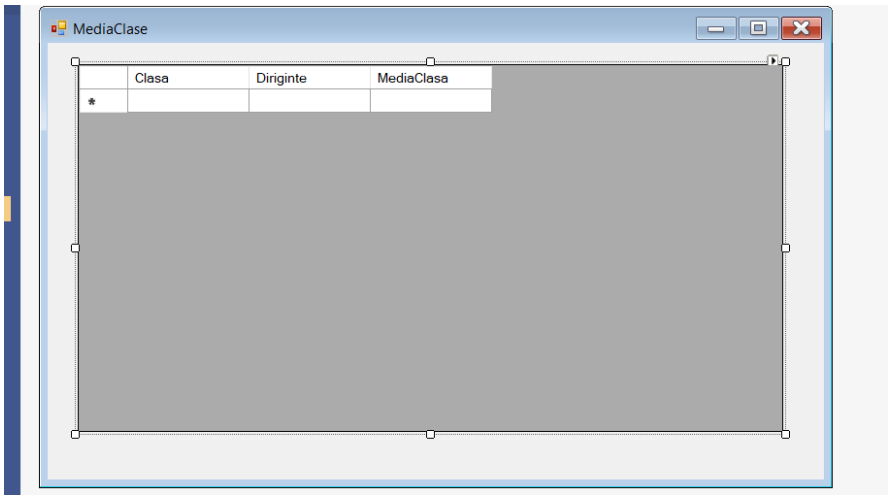


Se adauga un nou formular MediaClase:

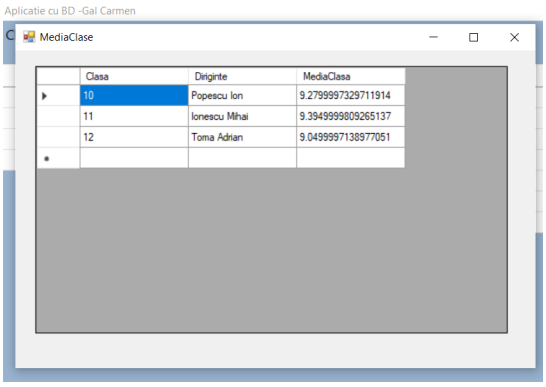


Se adauga DataGridView la formular și se adaugă vederea la sursa de date:





Se introduce codul in formularul principal la optiunea din meniu:

```
 private void mediaPeClaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            MediaClase f1 = new MediaClase();
            f1.Show();
        }
```

Rezultatul rularii: