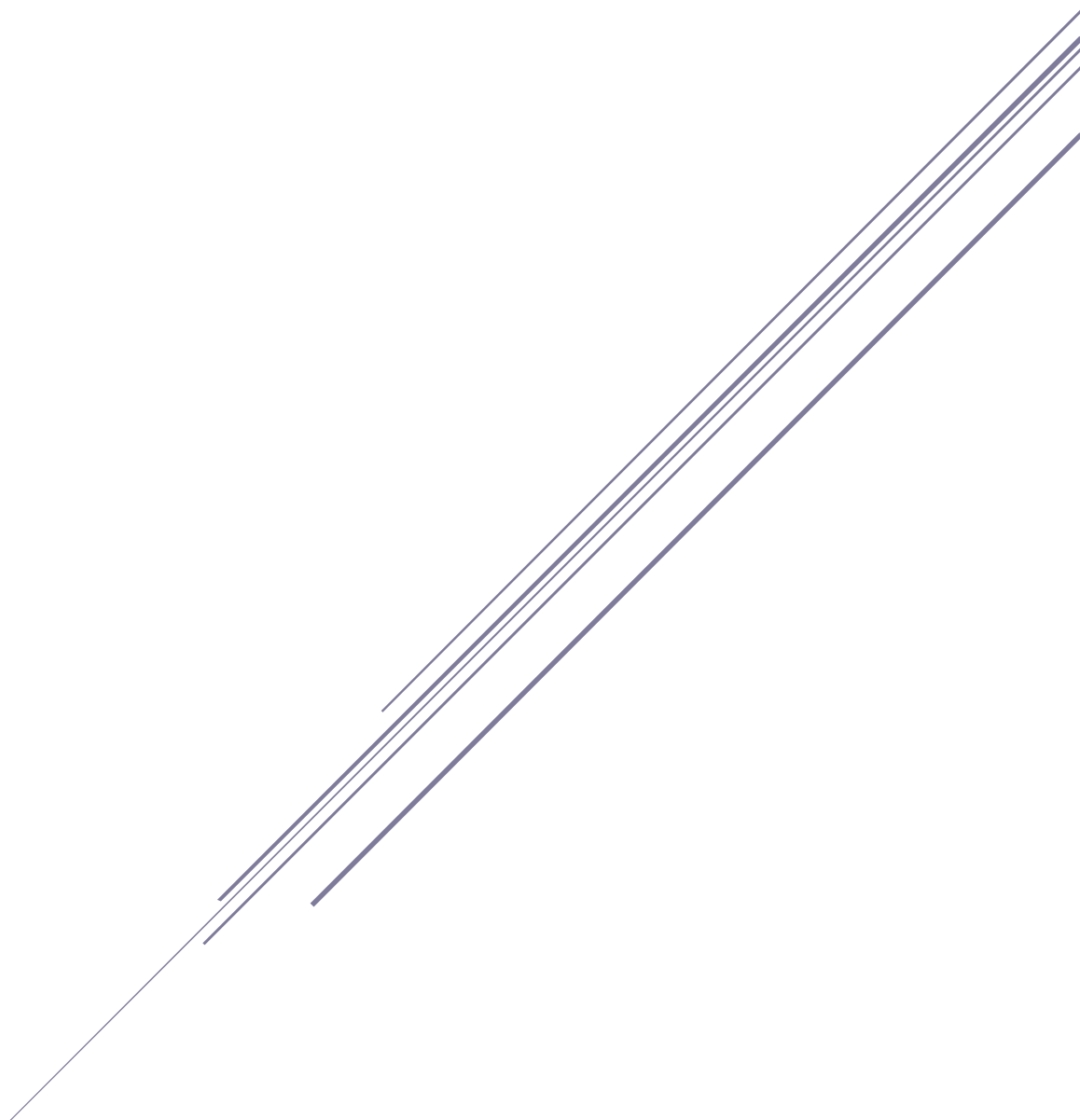


# APLICAȚIE PENTRU EVIDENȚA MANUALELOR ALTERNATIVE ACCEPTATE DE MINISTER/CULEGERI

Sima Maria-Cristina, 331AB



## DEFINIREA CERINȚELOR ȘI PROIECTAREA BAZEI DE DATE

Pentru a putea crea baza de date pentru evidența manualelor și a culegerilor, avem nevoie de 5 tabele principale (Element, Editură, Materie, Clasă, Autor) și unul de legătură (între tabelul Element și Autor, denumit "Element\_Autor").

Fiecare manual are cel puțin un autor, o editură de care aparține, un nume, un an al publicației, un număr de pagini, o materie, o clasă pentru care este dedicat (1-12), un semestru pentru care este creat (semestrul 1, semestrul 2 sau integral, pentru tot anul școlar), are un preț și este de un tip anume (manual sau culegere – de aici apărând și denumirea tabelii "Element"). Așadar, tabelul Element va avea următoarele câmpuri: ID Element (cheie primară), ID Autor (cheie externă), ID Editură (cheie externă), ID Clasă (cheie externă), Nume Element, An publicație, ID Materie (cheie externă), Nr Pagini, Semestru, Preț și Tip.

Câmpurile dedicate ID-urilor vor fi de tipul int, cu incrementare automată, Numele Elementului (manualului/culegerii) va fi un nvarchar de maxim 20 caractere, Anul Publicației va fi un nchar de 4 caractere, Nr pagini va fi un int, Semestru va fi un nchar(8), cu constrângerea că acest câmp poate fi doar '1', '2' sau 'integral', iar Tipul va fi un nchar(8) (doar "manual" sau "culegere" fiind cuvintele acceptate).

Fiecare autor va avea un ID (câmp de tipul int, cheie primară), un CNP, un Număr de telefon și un Nume de tip nchar de 20 caractere.

Un autor poate scrie mai multe manual/culegeri, iar un manual poate avea mai mulți autori – așadar, se observă necesitatea creării unei relații de tipul many to many, între tabelul Element și tabelul Autor. Astfel, se va crea tabelul de legătură Element\_Autor.

Va fi nevoie și de tabelul Materie, pentru a stoca ID-ul (cheie primară) și numele fiecărei materii.

De asemenea, va fi nevoie și de tabelul pentru ID-ul (cheie primară) și numele editurilor. De asemenea, este nevoie și de câmpurile Numele Persoană Contact și Nr Telefon Persoana Contact, în cazul în care este nevoie să se refacă stocul de manuale și culegeri.

Este nevoie și de un tabel pentru clasă, deoarece manualele sunt împărțite și în funcție de clasa pentru care sunt scrise. Câmpul "Număr Clasă" va fi constrâns să poată conține doar valori de tip int, între 1 și 12.

## IDENTIFICAREA RELAȚIILOR ÎNTRE TABELE

	<i>Element</i>	<i>Autor</i>	<i>Clasa</i>	<i>Editura</i>	<i>Materie</i>
<i>Element</i>		1:N	1:1	1:1	1:1
<i>Autor</i>	1:N				
<i>Clasa</i>	1:N				
<i>Editura</i>	1:N				
<i>Materie</i>	1:N				

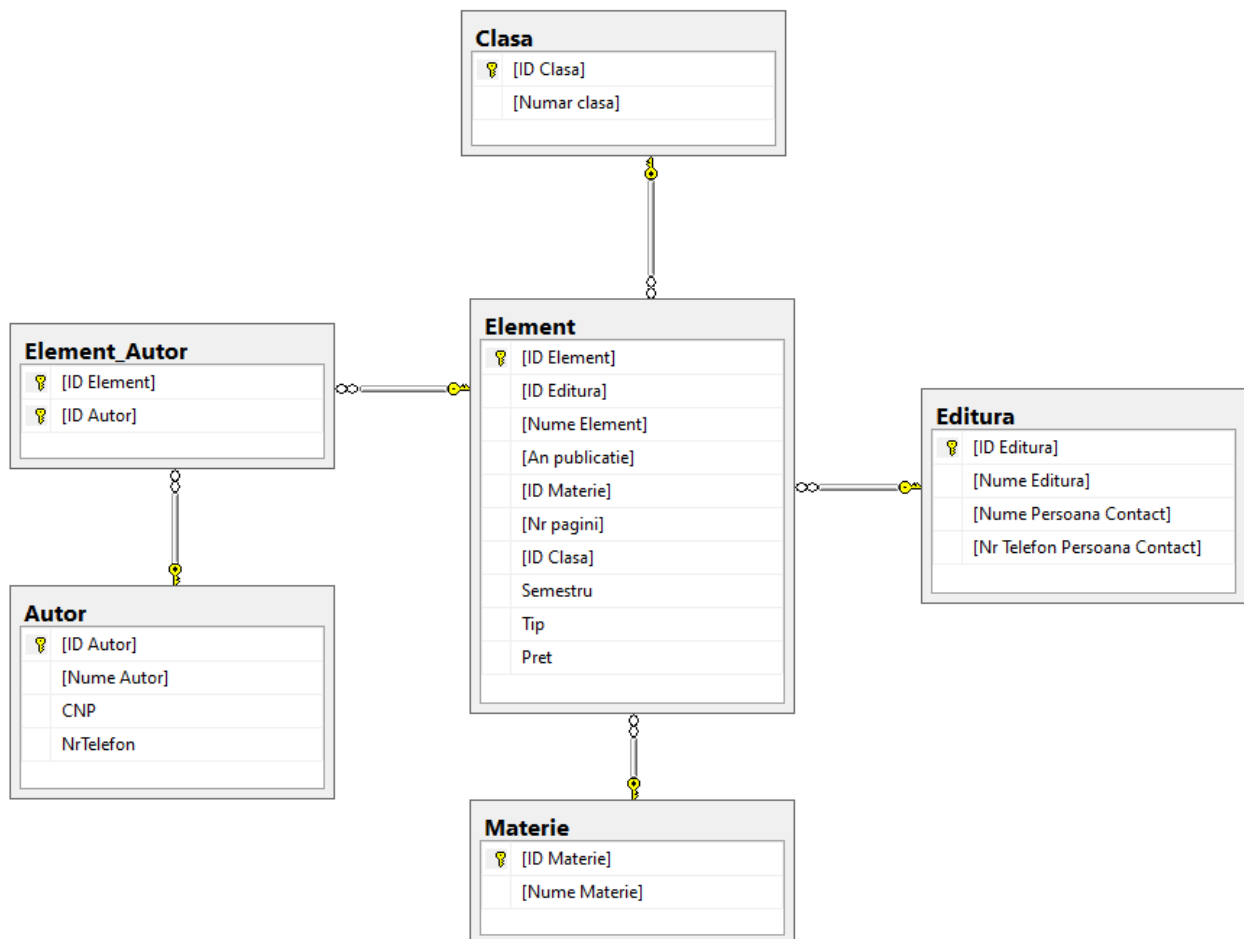
Element – Autor:  $1:N + 1:N = N:N$

Element – Clasa:  $1:1 + 1:N = 1:N$

Element – Editura:  $1:1 + 1:N = 1:N$

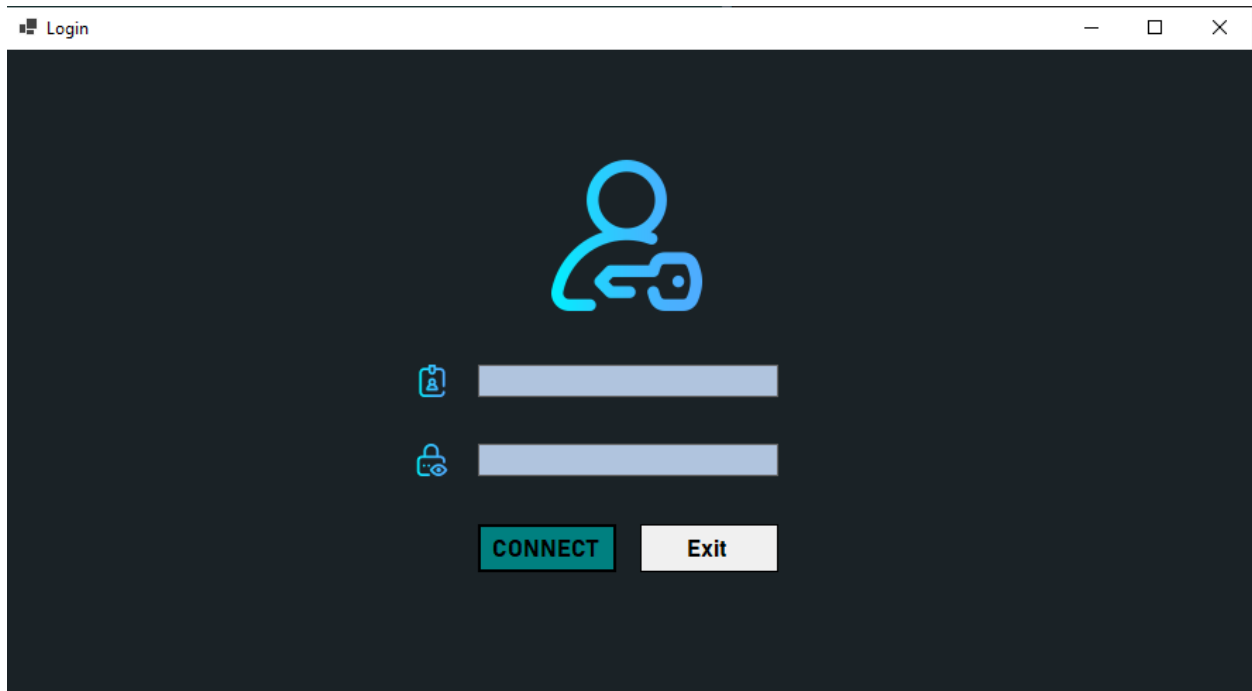
Element – Materie:  $1:1 + 1:N = 1:N$

Diagrama finală are o legătură N:N, între Element și Autor, și 3 legături 1:N, între Element-Editură, Element-Materie și Element-Clasa.



## FUNCȚIONAREA APLICAȚIEI

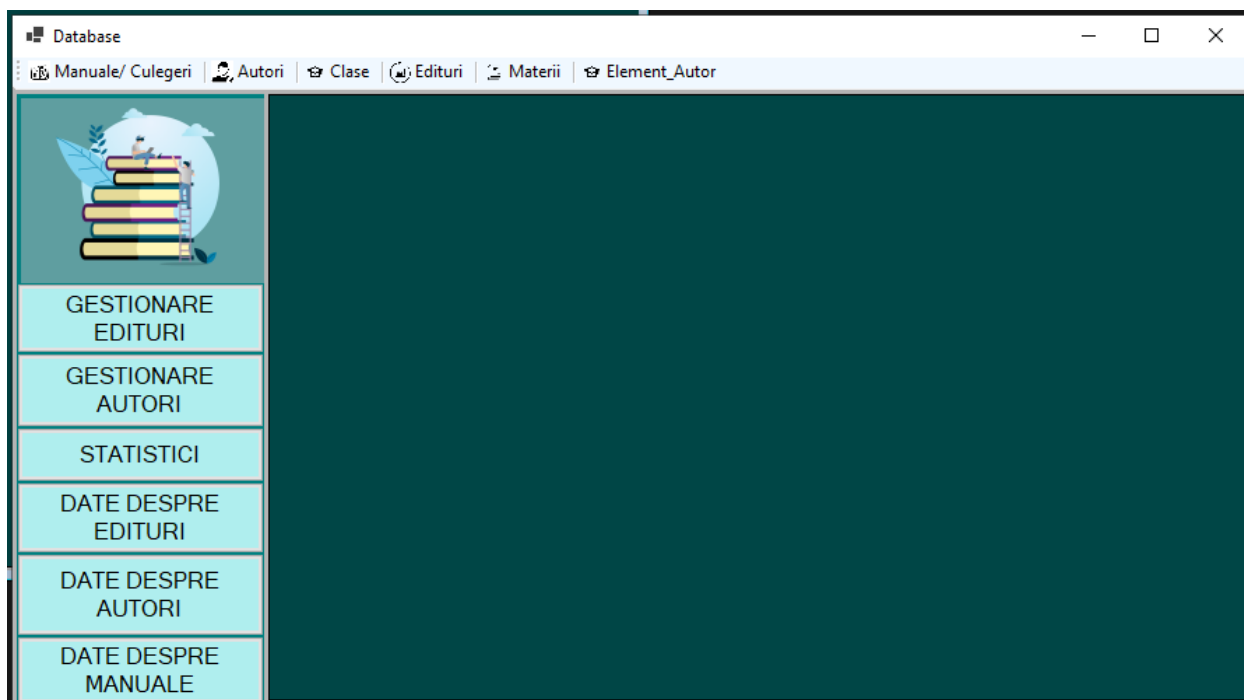
În urma rulării programului, aplicația se deschide cu pagina de Login, care permite utilizatorului, prin introducerea Username-ului "Admin" și a parolei "parola" în cele două TextBox-uri, să se conecteze la baza de date. Dacă, în schimb, sunt introduse alte informații, aplicația va arăta un mesaj de "Invalid login" și nu va permite conectarea la baza de date. Butonul de "Exit" închide aplicația.



După ce are loc conectarea la baza de date, se va deschide o nouă pagină, iar prima (cea cu Login-ul) va fi închisă.

Conectarea la baza de date se realizează astfel:

```
string connetionString;  
connetionString = @"Data Source = localhost\SQLEXPRESS;Initial Catalog = Tema; Integrated Security = True";  
cnn = new SqlConnection(connetionString);  
cnn.Open();
```



Aceasta este pagina principală a aplicației noastre, din care putem realiza diverse operații cu baza de date. Se observă, în partea de sus a aplicației, prezența unui meniu. Prin apăsarea butoanelor prezente pe meniu, utilizatorul poate deschide tabelele și poate accesa conținutul lor.

ID Element	ID Editura	Nume Element	An publicatie	ID Materie	Nr pagini	ID C
1	1	Istorie	2018	9	110	4
2	1	Matematica	2016	2	133	8
3	1	Fizica	2019	4	128	7
4	2	Matematic...	2020	2	187	2
5	2	Matematic...	2020	2	144	2
6	2	Comunicar...	2016	1	87	1
7	2	Comunicar...	2016	1	93	1
8	2	Manual de ...	2013	2	95	1
9	2	Chimie	2015	5	79	10
10	3	EV NAT 2...	2021	2	121	8
11	3	Matematic...	2020	2	132	7
12	3	Caietul ele...	2021	8	56	7
13	3	Informatic...	2019	10	108	5
14	3	Limba rom...	2008	1	146	7
15	3	Limba mod...	2014	3	96	1

Pentru a putea accesa și vizualiza conținutul tabelului Element, am procedat astfel:

```
if (check == true)
{
    SqlDataAdapter sqlData = new SqlDataAdapter(@"Select e.[Nume element], m.[nume materie],
ed.[nume editura], e.semestru, e.tip, e.pret, e.[an publicatie], e.[nr pagini],
e.[id clasa] from Element e
inner join materie m on m.[id materie]=e.[id materie]
inner join editura ed on ed.[id editura]=e.[id editura]", cnn);
    DataTable dtbl = new DataTable();
    sqlData.Fill(dtbl);
    dataGridView1.DataSource = dtbl;
}
```

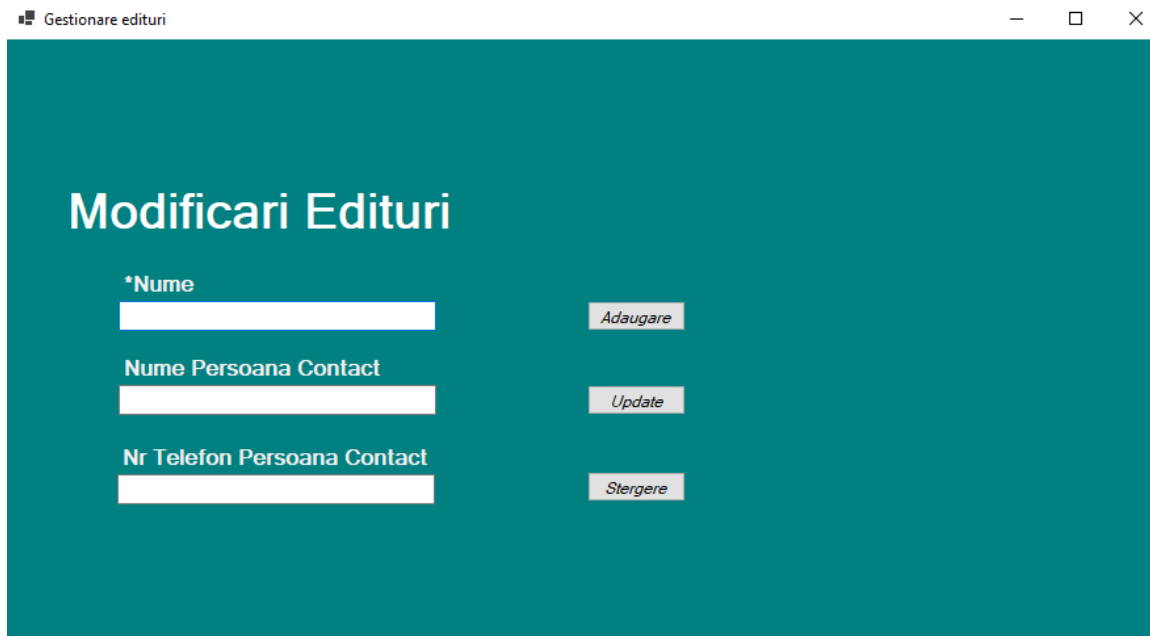
Iar pentru celelalte tabele am procedat astfel:

```
if (check == true)
{
    SqlDataAdapter sqlData = new SqlDataAdapter("Select * from Autor", cnn);
    DataTable dtbl = new DataTable();
    sqlData.Fill(dtbl);
    dataGridView1.DataSource = dtbl;
}
```

În partea stângă a paginii se află următoarele butoane:

## 1. GESTIONARE EDITURI/ GESTIONARE AUTORI

Acest buton permite utilizatorului să intre într-o pagină nouă, de unde poate să adauge o nouă editură (prin apăsarea butonului "Adaugă"), să modifice câmpurile "Nume Persoana Contact" sau "Nr Telefon Persoană Contact" ale unei Edituri cu un anumit Nume (prin apăsarea butonului "Update"), sau să ștergă o Editură (prin apăsarea butonului "Ștergere").



Pentru butonul de Insert, am folosit următorul cod:

```
SqlCommand command = new SqlCommand(@"insert into Editura ([Nume Editura], [Nume Persoana Contact],  
[Nr Telefon Persoana Contact]) values ('" + textBox1.Text + "', '" + textBox2.Text + "', '" +  
" '" + textBox3.Text + "')", cnn);  
  
command.ExecuteNonQuery();
```

Pentru butonul de Update, am folosit următorul cod:

```
SqlCommand command = new SqlCommand(@"update Editura set [Nume Persoana Contact]='" + textBox2.Text+"', " +  
"[Nr Telefon Persoana Contact]='" + textBox3.Text+ "' where [Nume Editura]='" + textBox1.Text + "'", cnn);  
command.ExecuteNonQuery();  
MessageBox.Show("Data has been updated");
```

Pentru butonul de Delete, am folosit următorul cod:

```
SqlCommand command = new SqlCommand("delete from Editura where [Nume Editura]='" + textBox1.Text + "'", cnn);  
command.ExecuteNonQuery();  
MessageBox.Show("Data has been deleted");
```

Și la fel am procedat și pentru tabelul dedicat Autorilor.

## 2. STATISTICI

Când este apăsat butonul de Statistici, se deschide o pagină nouă, care conține cele 6 interogări simple (cele care presupun interogări la două sau mai multe tabele, utilizând join-uri).

The screenshot shows a window titled 'Statistici' with a subtitle 'Statistici privind manualele si culegerile din baza de date'. It contains six search filters arranged in a 2x3 grid:

- Manualele scrise de:** A text input field followed by a 'Cautare' button.
- Numele autorilor care au cel puțin [input] manuale scrise:** A text input field followed by a 'Cautare' button.
- Manualele/culegerile de la editura:** A text input field followed by a 'Cautare' button.
- Materiile cu mai mult de 3 manuale:** A table with columns 'Nume' and 'Materie'. The 'Materie' column is expanded, showing 'Limba romana ...' and 'Matematica ...'.
- Materie:** A text input field followed by a 'Cautare' button.
- Tip:** A text input field followed by a 'Cautare' button.
- Clasa:** A text input field followed by a 'Cautare' button.
- Semestru:** A text input field followed by a 'Cautare' button.

Interogările simple sunt:

a) Permite utilizatorului să caute toate manualele sau culegerile scrise de un anumit autor, introdus de el de la tastatură.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"Select DISTINCT m.[Nume Element], a.[Nume Autor]
From Element m inner join Element_Autor ea on ea.[ID Element] = m.[ID Element]
inner join Autor a on a.[ID Autor] = ea.[ID Autor] where A.[Nume Autor] = '"+textBox6.Text+"'", cnn);
DataTable dtbl = new DataTable();
```

b) Afișează numele autorilor care au cel puțin NR manuale/culegeri scrise, unde NR este un număr introdus de utilizator de la tastatură.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"select a.[Nume Autor] from autor a inner join element_autor ea
on a.[id autor] = ea.[id autor] inner join element e on e.[id element] = ea.[id element]
group by a.[nume autor] having count(e.[id element]) > '"+textBox1.Text+'", cnn);
```



c) Afișează manualele/culegerile de la o anumită editură, introdusă de către utilizator de la tastatură.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"Select e.[Nume Element] from Element e inner join Editura ea  
on e.[ID Editura] = ea.[ID Editura] where ea.[Nume Editura] = '"+textBox7.Text+"'", cnn);  
DataTable dtb1 = new DataTable();
```

d) Afișează materiile cu mai mult de 3 manuale/culegeri.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"select m.[Nume Materie] from materie m  
inner join element e on e.[id materie] = m.[id materie] group by m.[nume materie]  
having count(e.[id element]) > 3", cnn);
```

e) Afișează manualele/culegerile de o anumită materie și de un anumit tip.

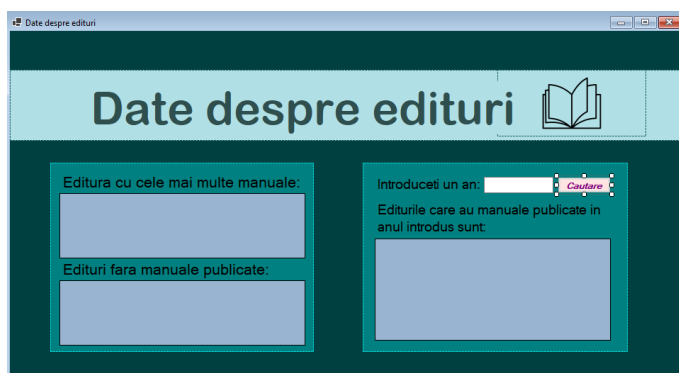
```
SqlDataAdapter sqlData = new SqlDataAdapter(@"select e.[Nume Element], e.pret from element e  
inner join materie m on m.[id materie]=e.[id materie] where ((m.[nume materie]= '"+textBox2.Text+"') +  
" AND (e.Tip = '"+textBox4.Text+"'))", cnn);  
DataTable dtb1 = new DataTable();
```

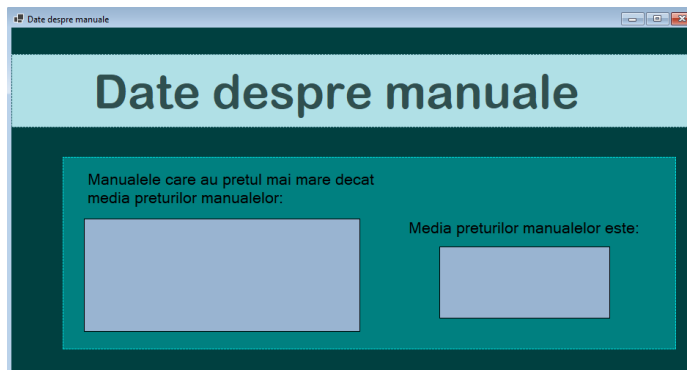
f) Afișează manualele/culegerile de o anumită clasă și de un anumit semestru.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"select e.[Nume Element], e.pret from element e  
inner join clasa c on c.[id clasa]=e.[id clasa] where ((c.[numar clasa]= '" + textBox5.Text + "') +  
" AND (e.Semestru = '" + textBox3.Text + "')", cnn);  
DataTable dtb1 = new DataTable();
```

### 3. DATE DESPRE EDITURI/AUTORI/MANUALE

Următoarele 3 butoane furnizează informații despre edituri, autori și manuale/culegeri, folosindu-se de interogări complexe.





Interogările complexe:

a) Afișează editura cu cele mai multe manuale/culegeri publicate.

```
cnn.Open();
SqlDataAdapter sqlData = new SqlDataAdapter(@"select ed.[nume editura] from editura ed
inner join element e on ed.[id editura] = e.[id editura] group by ed.[nume editura]
having COUNT(ed.[id editura]) >= (select top 1 count(m.[id editura]) from element m
group by m.[id editura] order by count(m.[id editura]) DESC)", cnn);
```

b) Afișează editurile fără manuale/culegeri publicate.

```
sqlData = new SqlDataAdapter(@"select ed.[id editura], ed.[nume editura] from editura ed
where ed.[id editura] NOT IN(select e.[id editura] from element e inner join editura ed2
on e.[id editura] = ed2.[id editura])", cnn);
```

c) Afișează editurile care au manuale publicate în anul introdus de la tastatură de către utilizator.

```
SqlDataAdapter sqlData = new SqlDataAdapter(@"select ed.[id editura], ed.[nume editura]
from editura ed inner join element e on ed.[id editura] = e.[id editura]
where ed.[id editura] = any (select m.[id editura] from element m
where m.[an publicatie] = '"+textBox1.Text+"' ) group by ed.[id editura], ed.[nume editura]", cnn);
```

d) Afișează autorii care nu au scris/publicat niciun manual/ nicio culegere.

```
cnn.Open();  
SqlDataAdapter sqlData = new SqlDataAdapter(@"select a.[Nume Autor] from autor a  
where a.[id autor] not in (select b.[id autor] from autor b  
inner join element_autor ea on ea.[ID autor] = b.[ID autor]  
inner join element e on e.[id element] = ea.[id element]  
where ((b.[id autor] = ea.[id autor]) AND(ea.[id element] = e.[id element])))", cnn);  
DataTable dtbl = new DataTable();
```

e) Afișează manualele/culegerile care au prețul mai mare decât media prețurilor tuturor manualelor/culegerilor.

```
cnn.Open();  
SqlDataAdapter sqlData = new SqlDataAdapter(@"select e.[id element], e.[nume element], e.pret  
from element e group by e.[id element], e.[nume element], e.pret  
having e.pret > (select avg(pret) from element)", cnn);  
DataTable dtbl = new DataTable();
```