

Colegiul Național „Mihai Viteazul” Turda

Secția Matematică-Informatică

Intensiv Informatică

Lucrare pentru
atestarea competențelor

Manager's Help

O bază de date utilă pentru cei care
doresc să-și administreze patrimoniul

Candidat:

Boar Victor

Profesor coordonator:

Prof. Pop Daniela

2023

Cuprins

1. Prezentare temei	2
2. Prezentarea limbajului	2
3. Resurse hardware și software necesar	3
4. Manualul utilizatorului	4
5. Manualul programatorului	7
6. Îmbunătățiri	19
7. Bibliografie	19

1. Tema

Am creat această aplicație pentru gestionarea cantitativ- valorică a patrimoniului unei instituții. Interfața aplicației este simplă și intuitivă, putând fi folosită pentru mai multe tipuri de gestiuni. Înregistrarea inventarelor se face în urma creării unui cont în baza numelui, email-ului și parolei. Conectarea la cont se realizează introducând email-ul și parola.

Principalele caracteristici ale aplicației sunt: înregistrarea obiectelor din patrimoniu cu denumire, unitate de măsură, cantitatea acestora, valoarea și data achiziționării.

2. Prezentarea limbajului

Limbajul C# este unul dintre cele mai utilizate limbaje de programare multiparadigmă din lume, fiind clasat în acest moment pe locul 5 mondial, ca nivel de popularitate. Este un limbaj simplu, modern, cu o flexibilitate foarte mare în ceea ce privește dezvoltarea de aplicații și portabilitatea acestora.

Date istorice

Istoria acestui limbaj își are originile la începutul mileniului în care ne aflăm. În iulie 2000 a fost lansată prima distribuție a limbajului C#, deși zvonurile apariției unui limbaj puternic au apărut încă de prin anul precedent. Microsoft a fost corporația care a decis să intre mult mai în forță în lumea dezvoltatorilor, întrucât limbajul Basic își pierduse popularitatea ceea ce a dus la o ușoară scădere a nivelului economic al companiei.

Unul dintre principalele motive care au inițiat crearea acestui limbaj de programare a fost legat de faptul că în anul 1995, cei de la Sun Microsystems au creat limbajul Java, primul limbaj multiparadigmă cu o portabilitate foarte mare. A înregistrat un nivel de popularitate exponențial în primii ani de la lansare, și chiar în zilele noastre, este pe primul loc mondial în ceea ce privește popularitatea. Astfel, Microsoft a decis să creeze un limbaj care să ofere mult mai multe elemente dezvoltatorilor.

Caracteristici

Printre principalele calități ale limbajului, putem distinge:

- modernitate, simplitate, utilitate generală, productivitate mare
- stabilitate în cadrul aplicațiilor complexe, durabilitate
- este un limbaj total orientat pe obiect (orice entitate din acest limbaj este de fapt un obiect)
- oferă suport complet pentru dezvoltarea de componente necesare în medii distribuite, deci este și un limbaj
- orientat către componente

3. Resurse hardware și software necesare

Pentru a crea această aplicație și pentru a o rula, am folosit Visual Studio 2022 enterprise. Pentru a folosi Visual Studio 2022 enterprise calculatorul tău trebuie să aibă următoarele specificații tehnice:

Visual Studio 2022 este acceptat pe următoarele sisteme de operare pe 64 de biți:

- Windows 11 versiunea 21H2 sau mai recentă: Home, Pro, Pro Education, Pro for Workstations, Enterprise și Education
- Windows 10 versiunea 1909 sau mai recentă: Home, Professional, Education și Enterprise.
- Windows Server 2022: Standard și Datacenter.
- Windows Server 2019: Standard și Datacenter.
- Windows Server 2016: Standard și Datacenter.

Notă:

Următoarele nu sunt acceptate:

- Sistemele de operare de 32 de biți și ARM32.
- Windows 11 Home în modul S, Windows Enterprise IoT, Windows 10 IoT Core, ediția Windows 10 Enterprise LTSC, Windows 10 S și Windows 10 Team Edition. În aceste ediții de Windows, puteți utiliza Visual Studio 2022 pentru a construi aplicații care rulează pe acestea.
- Opțiunile Server IoT, Server Core și Minimal Server Interface pentru Windows Server.
- Containerele Windows, cu excepția Visual Studio Build Tools.
- Rularea într-un mediu de mașină virtuală fără un sistem de operare Windows complet.
- Soluțiile de virtualizare a aplicațiilor, cum ar fi Microsoft App-V sau MSIX pentru Windows, sau tehnologiile de virtualizare a aplicațiilor de terțe părți.
- Mai mulți utilizatori simultani care utilizează software-ul pe aceeași mașină, inclusiv mașinile de infrastructură virtuală partajate desktop virtual sau un grup de gazdă Windows Virtual Desktop.

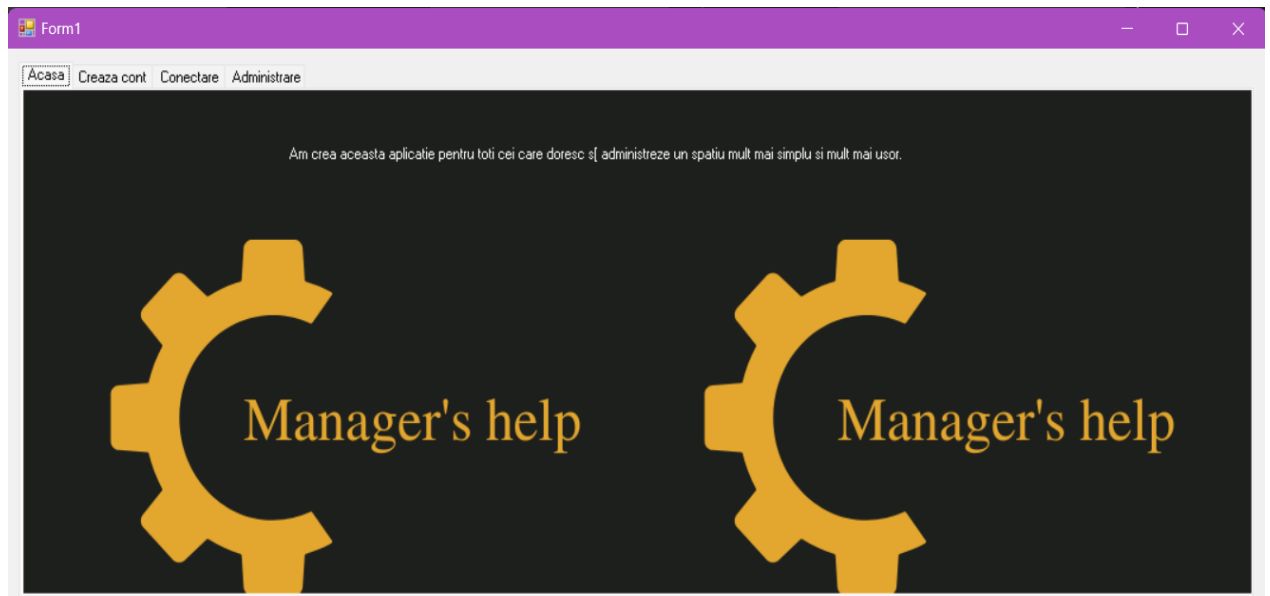
Pentru îndrumare cu privire la îmbunătățirea performanței, consultați Optimizarea performanței Visual Studio.

Procesor ARM64 sau x64; se recomandă un procesor quad-core sau mai performant. Procesoarele ARM 32 nu sunt acceptate. Minim 4 GB de RAM. Mulți factori afectează resursele utilizate; recomandăm 16 GB de RAM pentru soluțiile profesionale tipice. Windows 365: Minim 2 vCPU și 8 GB de RAM. Se recomandă 4 vCPU și 16 GB de RAM. Spațiu pe hard disk: minim 850 MB până la 210 GB de spațiu disponibil, în funcție de caracteristicile instalate; instalările tipice necesită 20-50 GB de spațiu liber. Recomandăm instalarea Windows și Visual Studio pe un solid-state drive (SSD) pentru a crește performanța. Placă video care suportă o rezoluție minimă de afișaj WXGA (1366 x 768); Visual Studio va funcționa cel mai bine la o rezoluție de 1920 x 1080 sau mai mare. Rezoluția minimă presupune că setările de zoom, DPI și scalare a textului sunt setate la 100%. Dacă nu sunt setate la 100%, rezoluția minimă trebuie ajustată corespunzător. De exemplu, dacă setați setarea "Scalare și aspect" de afișare Windows de pe Surface Book, care are un afișaj fizic de 3000 x 2000, la 200%, atunci Visual Studio va vedea o rezoluție logică de ecran de 1500 x 1000, îndeplinind cerința minimă de 1366 x 768.

4. Manualul utilizatorului

Aplicația are patru pagini.

Prima pagină este pagina ACASĂ care permite utilizatorului să vadă motivele pentru care această aplicație a fost creată.



A doua pagină este pagina CREAȘĂ CONT care permite utilizatorului să creeze un cont introducând un nume, un email și o parolă, iar apoi apăsând butonul CREAȘĂ CONT să își creeze propriul cont pe platformă.



A treia pagină este pagina CONECTARE care permite utilizatorului să se conecteze la un cont deja existent introducând emailul și parola, iar apoi apăsând pe butonul CONECTARE sau să se deconecteze de pe contul pe care se află în momentul respectiv apăsând butonul DECONECTARE.

The screenshot shows a web application window titled 'Form1'. It has a navigation bar with links: 'Acasa', 'Creaza cont', 'Conectare', and 'Administrare'. The main area contains a login form with 'Email:' and 'Parola:' labels, input fields, and 'Conectare' and 'Deconecta' buttons. The background is dark with two large yellow gears and the text 'Manager's help' repeated twice.

A patra pagină permite utilizatorului să își administreze propriul spațiu putând adăuga obiecte.

The screenshot shows the 'Administrare' page of the application. It features a form on the left for adding new objects with fields for 'Nr. curent', 'Denumire obiect', 'Cantitate', 'Valoare / buc (lei)', 'Data achizitionarii', 'Data expirarii', and 'Nr inventar'. Below the form are 'Steroe' and 'Adauga' buttons. On the right, a table displays existing objects.

Nr. Ctr.	Nume obiect	Cantitate	Valoare	Data achizitionarii
1	masa	3	100	03.04.2023 00:00:00

introducând numele obiectului, cantitatea, valoarea în lei a obiectului / bucată, data achiziționării, data expirării și numărul de inventar, iar apoi să apese butonul ADAUGĂ

Form1

Acasa Creaza cont Conectare Administrare

Nr. curent: 2
Denumire obiect: Scaun
Cantitate: 3
Valoare / buc (lei): 15
Data achizitionarii: 11.05.2023
Data expirarii: 22.02.2025
Nr inventar: 5

Nr.Ctr. Nume obiect Cantitate Valoare Data achizitionarii

1	masa	3	100	03.04.2023 00:00:00
---	------	---	-----	---------------------

Sterge Adauga

sau să ștergă un obiect selectându-l din căsuța de lângă butonul ștergere numărul curent al obiectului dorit printr-un simplu click, iar apoi apăsând butonul ȘTERGE.

Form1

Acasa Creaza cont Conectare Administrare

Nr. curent:
Denumire obiect:
Cantitate:
Valoare / buc (lei):
Data achizitionarii: 11.05.2023
Data expirarii: 11.05.2023
Nr inventar:

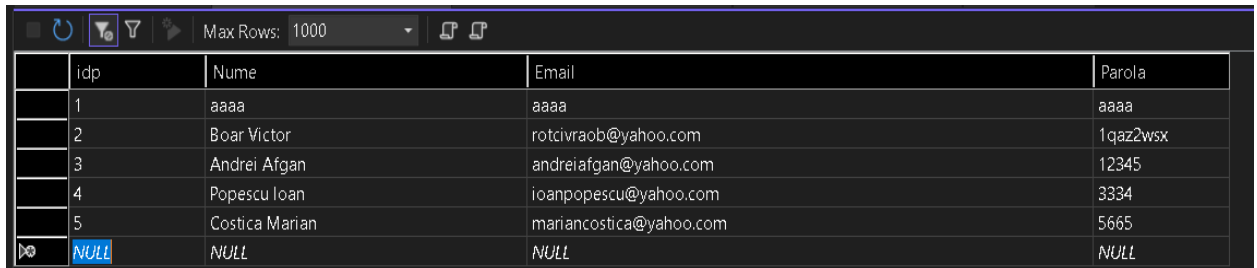
Nr.Ctr. Nume obiect Cantitate Valoare Data achizitionarii

1 Sterge Adauga

5. Manualul programatorului

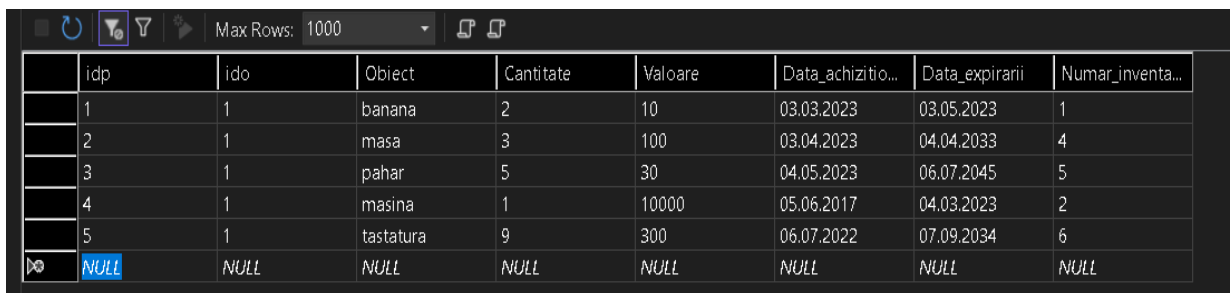
Pentru a realiza această aplicație am creat o bază de date cu două tabele.

Primul tabel se numește PERSOANE și reține informații despre utilizatori precum id-ul, numele, adresa de email și parola contului.



	idp	Nume	Email	Parola
	1	aaaa	aaaa	aaaa
	2	Boar Victor	rotcivraob@yahoo.com	1qaz2wsx
	3	Andrei Afgan	andreiafgan@yahoo.com	12345
	4	Popescu Ioan	ioanpopescu@yahoo.com	3334
	5	Costica Marian	mariancostica@yahoo.com	5665
	NULL	NULL	NULL	NULL

A doua tabelă se numește OBIECTE și reține informații despre obiecte, precum id-ul persoanei care îl deține, id-ul său, numele, cantitatea, valoarea / bucată, data achiziționării, data expirării și numărul de inventar.



	idp	ido	Obiect	Cantitate	Valoare	Data_achizitio...	Data_expirarii	Numar_inventa...
	1	1	banana	2	10	03.03.2023	03.05.2023	1
	2	1	masa	3	100	03.04.2023	04.04.2033	4
	3	1	pahar	5	30	04.05.2023	06.07.2045	5
	4	1	masina	1	10000	05.06.2017	04.03.2023	2
	5	1	tastatura	9	300	06.07.2022	07.09.2034	6
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Între cele două tabele există o relație. Tabela PERSOANE fiind tabela părinte, iar tabela OBIECTE fiind tabela copil.

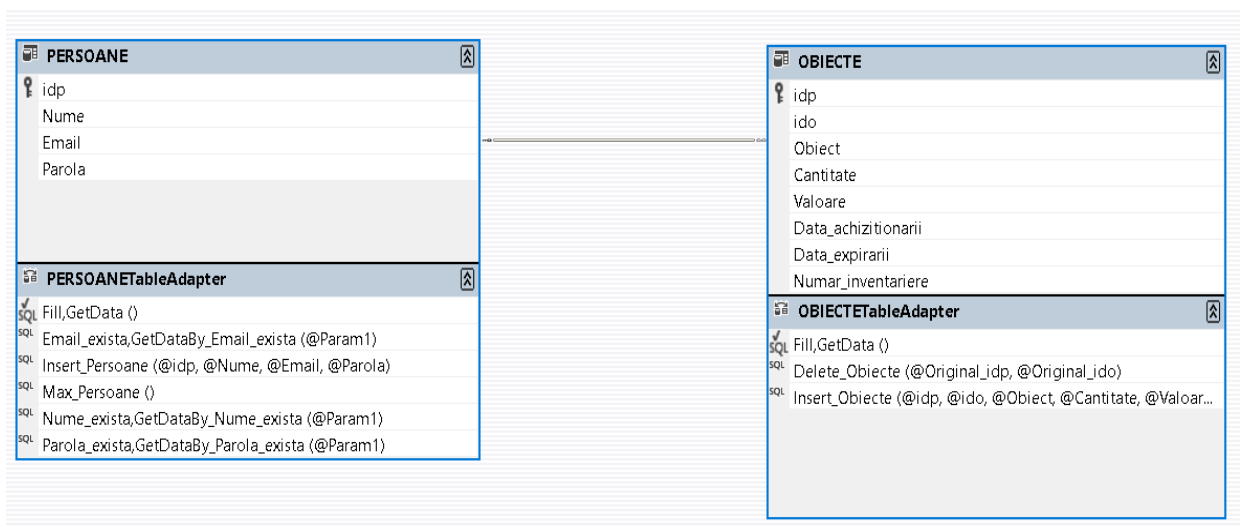
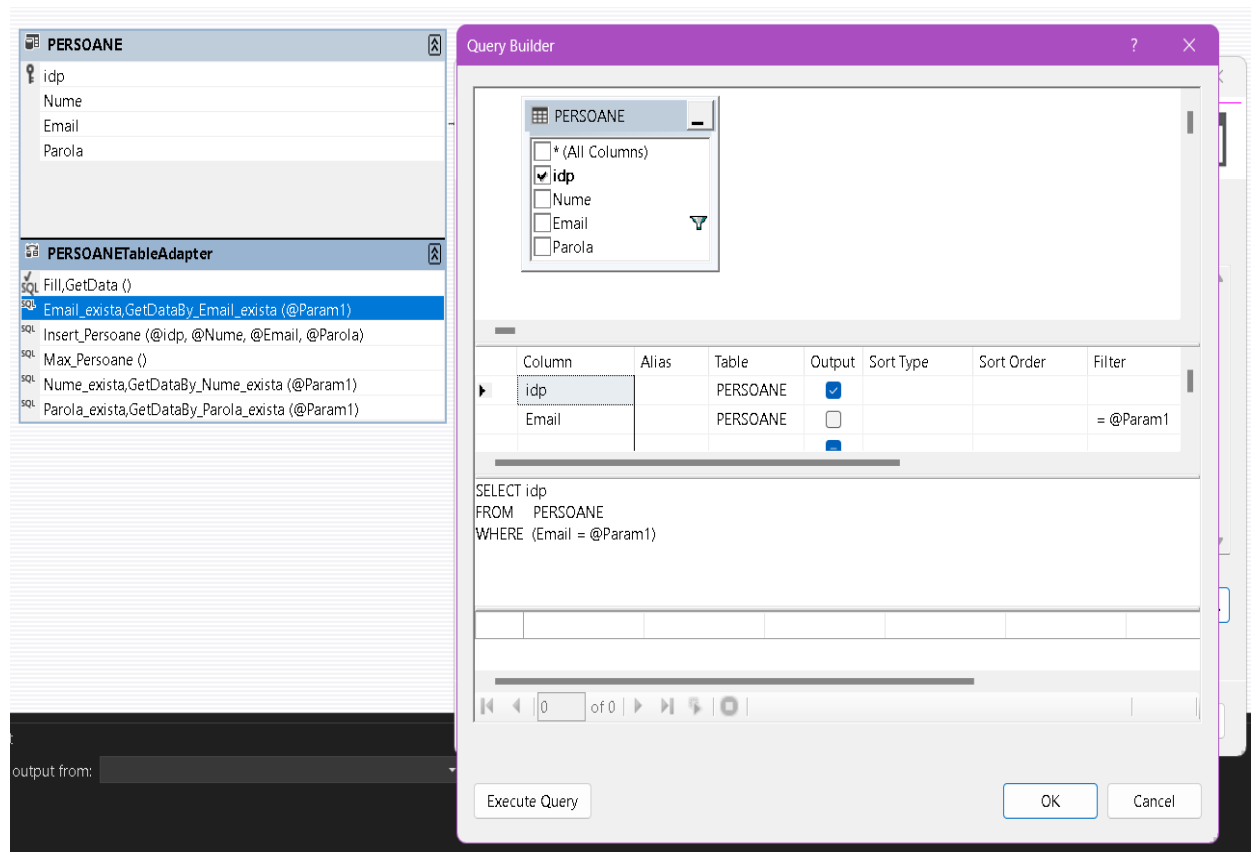


Tabela Persoane conține mai multe query-uri printre care se numără:

Un query pentru a identifica dacă există un anumit email în tabela PEROSANE.



Un query care identifică dacă există un anumit nume în tabela PERSOANE.

The screenshot shows the Query Builder interface. On the left, the 'PERSOANE' table is listed with columns: idp, Nume, Email, and Parola. Below it, the 'PERSOANETableAdapter' is shown with several SQL methods. The 'Nume_exista,GetDataBy_Nume_exista (@Param1)' method is selected. The main area displays the query structure and the resulting SQL statement.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
idp		PERSOANE	<input checked="" type="checkbox"/>			
Nume		PERSOANE	<input type="checkbox"/>			= @Param1

```
SELECT idp
FROM PERSOANE
WHERE (Nume = @Param1)
```

At the bottom, there is an 'Execute Query' button and 'OK' and 'Cancel' buttons.

Un query care identifică dacă există o anumită parolă în tabela PERSOANE.

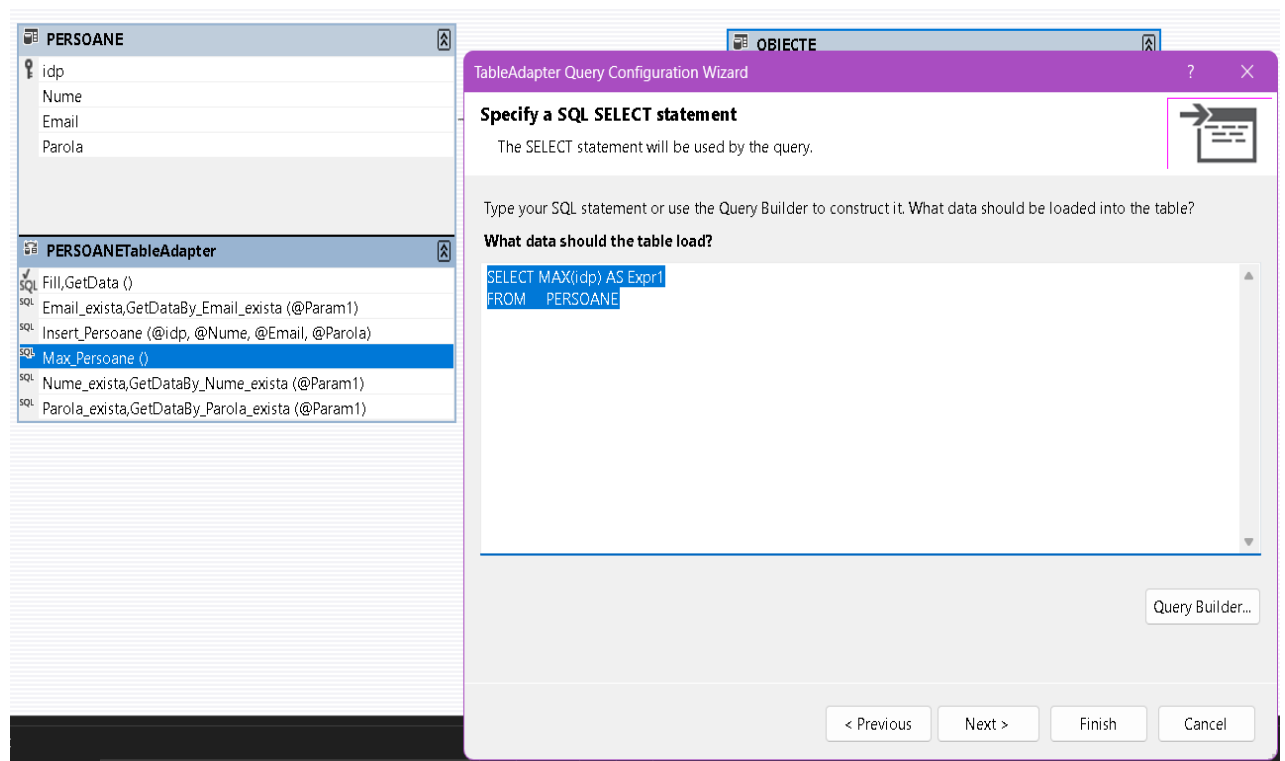
The screenshot shows the Query Builder interface. On the left, the 'PERSOANE' table is listed with columns: idp, Nume, Email, and Parola. Below it, the 'PERSOANETableAdapter' is shown with several SQL methods. The 'Parola_exista,GetDataBy_Parola_exista (@Param1)' method is selected. The main area displays the query structure and the resulting SQL statement.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
idp		PERSOANE	<input checked="" type="checkbox"/>			
Parola		PERSOANE	<input type="checkbox"/>			= @Param1

```
SELECT idp
FROM PERSOANE
WHERE (Parola = @Param1)
```

At the bottom, there is an 'Execute Query' button and 'OK' and 'Cancel' buttons.

Un query care să identifice id-ul amxim existent în tabela PERSOANE.



Un query care să insereze o persoană în tabela PERSOANE.

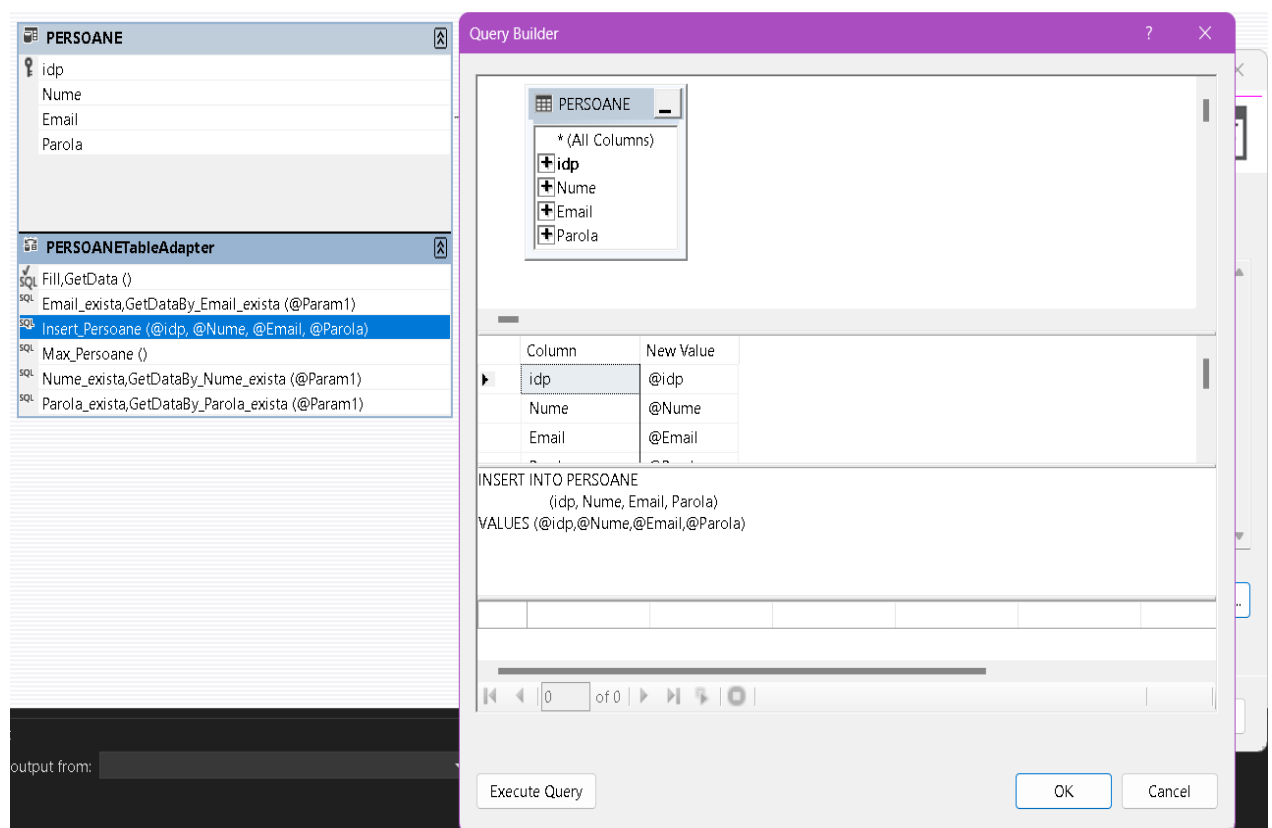
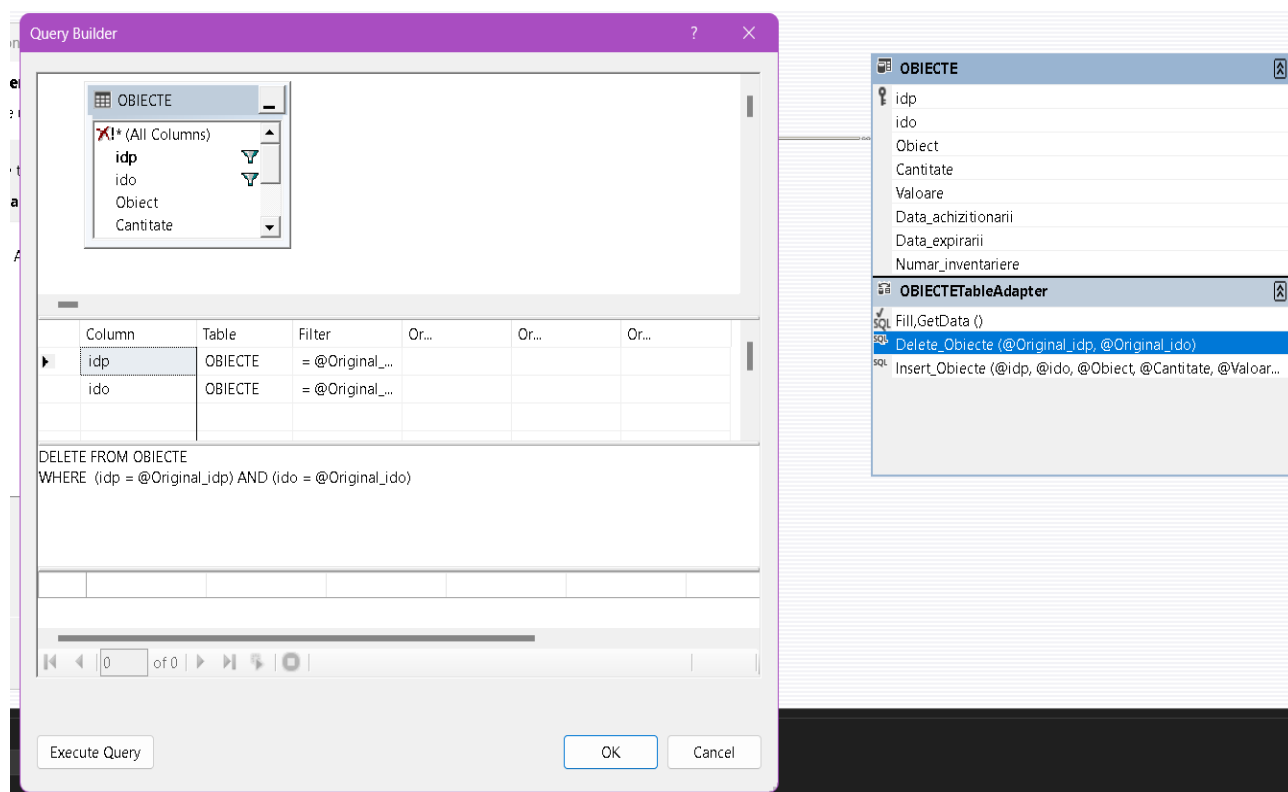
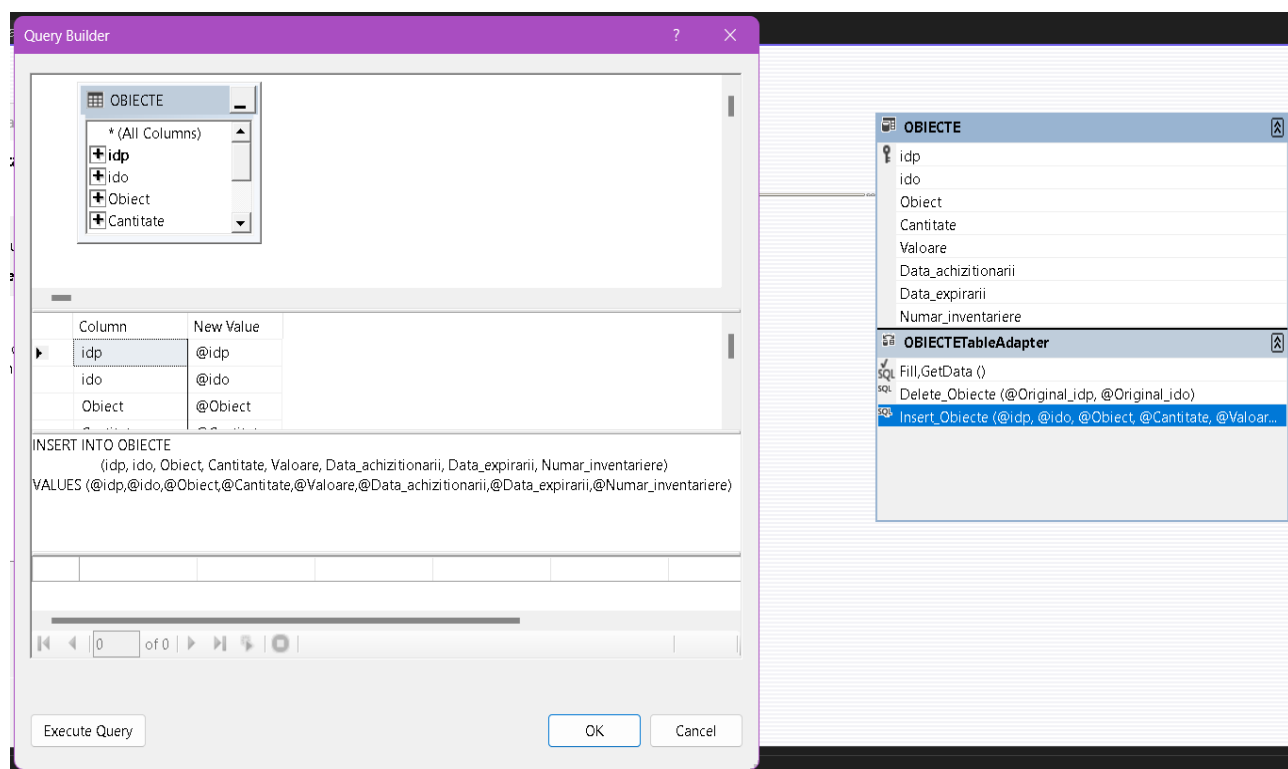


Tabela OBIECTE conține două query-uri:

Un query care să șteargă un obiect din tabela OBIECTE



Un query care să adauge un obiect în tabela OBIECTE



Butonul care determină creerea unui cont nou se folosește de mai multe subprograme pentru a verifica dacă nu cumva vre-un text box este gol, caz în care returnează un mesaj, sau dacă informația dintr-un text box este deja folosită în tabela PERSOANE caz în care din nou returnează un mesaj. În cazul în care totul este bine programul se va folosi de informațiile adăugate de tine în pentru a crea un cont nou de utilizator.

```
private void button_Creaza_cont_nou_Click(object sender, EventArgs e)
{
    if (numetextBox.Text == "")
        MessageBox.Show("Trebuie sa introduci un nume");
    else
        if (nume())
            if (emailtextBox.Text == "")
                MessageBox.Show("Trebuie sa introduci un email");
            else
                if (email())
                    if (parolatextBox.Text == "")
                        MessageBox.Show("Trebuie sa introduci o parola");
                    else
                        if (parola())
                            if (confirmare_parolatextBox.Text != parolatextBox.Text)
                                MessageBox.Show("Re confirmati parola");
                            else
                                {
                                    this.pERSOANETableAdapter.Fill(this.databaseDataSet.PERSOANE);

                                    int mx_idp = int.Parse(pERSOANETableAdapter.Max_Persoane().ToString());

                                    pERSOANETableAdapter.Insert_Persoane(mx_idp + 1, numetextBox.Text, emailtextBox.Text, parolatextBox.Text);

                                    numetextBox.Text = string.Empty;
                                    emailtextBox.Text = string.Empty;
                                    parolatextBox.Text = string.Empty;
                                    confirmare_parolatextBox.Text = string.Empty;

                                    this.pERSOANETableAdapter.Fill(this.databaseDataSet.PERSOANE);
                                }
                        }
                    }
                }
            }
        }
}
```

Butonul Conectare se folosește de mai multe subprograme pentru a verifica dacă nu cumva vre-un text box este gol, caz în care returnează un mesaj, sau dacă informația dintr-un text box este invalidă, caz în care din nou va return un mesaj. În cazul în care toate informațiile sunt valide programul te va conecta în contul tău pentru a putea să îți administrezi lista ta de obiecte.

```
private void button_Conectare_Click(object sender, EventArgs e)
{
    bool ok = false;

    if (textBox_Email.Text == "")
        MessageBox.Show("Trebuie sa introduci un email");
    else
        if (textBox_Parola.Text == "")
            MessageBox.Show("Trebuie sa introduci o parola");
        else
            if (exista_parola() != exista_email() || exista_parola() == -1 || exista_email() == -1)
                MessageBox.Show("Acest cont nu exista");
            else
                if (idp != -1)
                    MessageBox.Show("Esti deja conectat intr-un cont, apasa butonul DECONECTARE pentru a parasi contul curent");
                else
                    ok = true;

    if (ok)
    {
        idp = exista_email();

        textBox_Email.Text = string.Empty;
        textBox_Parola.Text = string.Empty;

        this.oBIECTETableAdapter.Fill(this.databaseDataSet.OBIECTE);

        DataTable obiecte = this.databaseDataSet.OBIECTE;

        this.pERSOANETableAdapter.Fill(this.databaseDataSet.PERSOANE);

        for (int i = 0; i < obiecte.Rows.Count; ++i)
            if (int.Parse(obiecte.Rows[i]["idp"].ToString()) == idp)
            {
                string[] arr = new string[7];
                ListViewItem itm;

                arr[0] = obiecte.Rows[i]["ido"].ToString();
                arr[1] = obiecte.Rows[i]["Obiect"].ToString();
                arr[2] = obiecte.Rows[i]["Cantitate"].ToString();
                arr[3] = obiecte.Rows[i]["Valoare"].ToString();
                arr[4] = obiecte.Rows[i]["Data_achizitionarii"].ToString();
                arr[5] = obiecte.Rows[i]["Data_expirarii"].ToString();
                arr[6] = obiecte.Rows[i]["Numar_inventariere"].ToString();

                itm = new ListViewItem(arr);
                listView_Obiecte.Items.Add(itm);
            }

        this.oBIECTETableAdapter.Fill(this.databaseDataSet.OBIECTE);
        DataTable ob = databaseDataSet.OBIECTE;

        for (int i=0; i<ob.Rows.Count;++i)
            if (int.Parse(ob.Rows[i]["idp"].ToString()) == idp)
                comboBox_Stergere.Items.Add(ob.Rows[i]["ido"].ToString());

        MessageBox.Show("Te-ai conectat cu succes");
    }
}
```

Butonul Deconectare te deconectează total de contul tău pentru a te putea conecta la un alt cont și pentru a putea administra o altă listă.

```
private void button_Deconectare_Click(object sender, EventArgs e)
{
    if (idp != -1)
    {
        idp = -1;
        MessageBox.Show("Te-ai deconectat cu succes");

        listView_Obiecte.Items.Clear();

        this.OBIECTETableAdapter.Fill(this.databaseDataSet.OBIECTE);
        this.PERSOANETableAdapter.Fill(this.databaseDataSet.PERSOANE);

        comboBox_Stergere.Items.Clear();
    }
    else
        MessageBox.Show("Nu esti conectat in nici un cont");
}
```

Butonul Aăugare permite utilizatorului să adauge orice obiect în lista sa și utilizează mai multe subprograme pentru a verifica dacă toate informațiile referitoare la obiect sunt valide sau dacă vre-un text box nu este valid. În cazul în care totul este valid programul adaugă obiectul dorit de utilizator în lista utilizatorului din tabela OBIECTE

```
private void button_Adauga_Click(object sender, EventArgs e)
{
    if (idp != -1)
    {
        if (nr_curent_exista() && denumire_obiec_exista() && cantitate_exista() && valoare_exista() && numar_inventar_exista())
        {
            this.OBIECTTableAdapter.Fill(this.databaseDataSet.OBIECTE);

            DataTable obiecte = this.databaseDataSet.OBIECTE;

            oBIECTTableAdapter.Insert_Obiecte(idp, int.Parse(textBox_Nr_curent.Text), textBox_Denumire_obiect.Text, textBox_Cantitate.
            this.OBIECTTableAdapter.Fill(this.databaseDataSet.OBIECTE);

            comboBox_Stergere.Items.Add(textBox_Denumire_obiect.Text);

            listView_Obiecte.Items.Clear();

            textBox_Denumire_obiect.Text = string.Empty;
            textBox_Cantitate.Text = string.Empty;
            textBox_Valoare.Text = string.Empty;
            dateTimePicker_Data_achizitionarii.ResetText();
            dateTimePicker_Data_expirarii.ResetText();
            textBox_Nr_inventar.Text = string.Empty;

            for (int i = 0; i < obiecte.Rows.Count; ++i)
            {
                if (int.Parse(obiecte.Rows[i]["idp"].ToString()) == idp)
                {
                    string[] arr = new string[7];
                    ListViewItem itm;

                    arr[0] = obiecte.Rows[i]["ido"].ToString();
                    arr[1] = obiecte.Rows[i]["Obiect"].ToString();
                    arr[2] = obiecte.Rows[i]["Cantitate"].ToString();
                    arr[3] = obiecte.Rows[i]["Valoare"].ToString();
                    arr[4] = obiecte.Rows[i]["Data_achizitionarii"].ToString();
                    arr[5] = obiecte.Rows[i]["Data_expirarii"].ToString();
                    arr[6] = obiecte.Rows[i]["Numar_inventariere"].ToString();

                    itm = new ListViewItem(arr);
                    listView_Obiecte.Items.Add(itm);
                }
            }

            this.OBIECTTableAdapter.Fill(this.databaseDataSet.OBIECTE);
        }
        else
        {
            MessageBox.Show("Trebuie sa te conectezi pentru a adauga un obiect");
        }
    }
}
```

1 Reference

No issues found

Butonul Ștergere permite să șteargă orice obiect din lista sa într-un mod foarte simplu și rapid.

1 reference

```
private void button_Stergere_Click(object sender, EventArgs e)
{
    if (comboBox_Stergere.Text != "")
    {
        listView_Obiecte.Items[int.Parse(comboBox_Stergere.Text)-1].Remove();

        comboBox_Stergere.Items.Remove(comboBox_Stergere.Text);
    }
}
```

er's Help

```
int idp = -1;
```

1 reference

```
public bool email()
{
    this.pERSOANETableAdapter.Email_exista(this.databaseDataSet.PERSOANE, emailtextBox.Text);

    DataTable d = this.databaseDataSet.PERSOANE;

    if (d.Rows.Count != 0)
    {
        MessageBox.Show("Acest email este deja folosit");
        return false;
    }
    return true;
}
```

1 reference

```
public bool nume()
{
    this.pERSOANETableAdapter.Nume_exista(this.databaseDataSet.PERSOANE, numetextBox.Text);

    DataTable d = this.databaseDataSet.PERSOANE;

    if (d.Rows.Count != 0)
    {
        MessageBox.Show("Acest nume este deja folosit");
        return false;
    }
    return true;
}
```

1 reference

```
public bool parola()
{
    this.pERSOANETableAdapter.Parola_exista(this.databaseDataSet.PERSOANE, paroltextBox.Text);

    DataTable d = this.databaseDataSet.PERSOANE;

    if (d.Rows.Count != 0)
    {
        MessageBox.Show("Aceasta parola este deja folosita");
        return false;
    }
    return true;
}
```

3 references

```
public int exista_email()
{
    this.pERSOANETableAdapter.Email_exista(this.databaseDataSet.PERSOANE, textBox_Email.Text);

    DataTable d = this.databaseDataSet.PERSOANE;

    if (d.Rows.Count != 0)
        return int.Parse(d.Rows[0]["idp"].ToString());
    else
        return -1;
}
```

Toate subprogramele utilizate pentru ca aplicația să funcționeze complet.



```
Form1.cs [Design] DatabaseDataSet.Designer.cs DatabaseDataSet.xsd*
C# Manager's Help

75 public int exista_parola()
76 {
77     this.pPERSONTableAdapter.Parola_exista(this.databaseDataSet.PERSONAME, textBox_Parola.Text);
78
79     DataTable d = this.databaseDataSet.PERSONAME;
80
81     if (d.Rows.Count != 0)
82         return int.Parse(d.Rows[0]["idp"].ToString());
83     else
84         return -1;
85 }
86
87 1 reference
88 public bool nr_curent_exista()
89 {
90     if (textBox_Nr_curent.Text != "")
91         return true;
92     else
93         MessageBox.Show("Trebuie sa introduci un numar curent mai mare decat ultimul numar curent din list ta");
94
95     return false;
96 }
97 1 reference
98 public bool denumire_obiec_exista()
99 {
100     if (textBox_Denumire_obiect.Text != "")
101         return true;
102     else
103         MessageBox.Show("Trebuie sa introduci un obiect");
104
105     return false;
106 }
107 1 reference
108 public bool cantitate_exista()
109 {
110     if (textBox_Cantitate.Text != "")
111         return true;
112     else
113         MessageBox.Show("Trebuie sa introduci o cantitate");
114
115     return false;
116 }
117 1 reference
118 public bool valoare_exista()
119 {
120     if (textBox_Valoare.Text != "")
121         return true;
122     else
123         MessageBox.Show("Trebuie sa introduci un pret");
124
125     return false;
126 }
127 1 reference
128 public bool numar_inventar_exista()
129 {
130     if (textBox_Nr_inventar.Text != "")
131         return true;
132     else
133         MessageBox.Show("Trebuie sa introduci un numar de invenat");
134
135     return false;
136 }
137 1 reference
138 public Form1()
139 {
140 }
141 }

45 % No issues found
```

6. Îmbunătățiri

Aplicația poate fi îmbunătățită pe mai multe planuri pentru a se alina pe nevoile utilizatorilor. Câteva dintre îmbunătățirile care pot fi aduse aplicației se numără:

Adăugarea unei abilități de a modifica o anumită secțiune a unui obiect fără a fi nevoiți să ștergem tot obiectul.

O interfață mai prietenoasă cu utilizatorul.

Adăugarea unei notificări în momentul în care un obiect expiră sau se apropie de data expirării.

Posibilitate de a sorta obiectele în funcție de numărul lor de inventar.

7. Bibliografie

<https://www.youtube.com/watch?v=PAk31IekYj4&list=PL3TebpYEH0v3URYnJiHGbd0EvlzxJxh4>

<https://www.youtube.com/watch?v=mgtfxtjKoaA&list=PL3TebpYEH0v3URYnJiHGbd0EvlzxJxh4&index=2>

<https://www.youtube.com/watch?v=ZU16HQDz-uM&list=PL3TebpYEH0v3URYnJiHGbd0EvlzxJxh4&index=3>

<https://learn.microsoft.com/en-us/troubleshoot/developer/visualstudio/csharp/language-compilers/create-sql-server-database-programmatically>