

## Tema Semestrială la BD

### Cerinte

- Aplicatia gestioneaza lucrarile dintr-un service auto
- Se pot adauga lucrari noi, clienti noi, masini noi, piese pe stoc etc
- Se poate schimba starea unei lucrari (in desfasurare sau finalizata)
- Se vor putea cauta lucrari pentru a se verifica starea lor si anunta clientul  
daca lucrarea este finalizata
- Se va putea vedea daca un client este nou sau nu cu ajutorul atributului  
"Istoric Lucrari" din tabelul "Clienti", care poate fi NULL sau nu

Pentru realizarea bazei de date de gestiune a lucrarilor, am prelucrat 5 tabele principale si doua de legatura (in ajutorul relatiilor many to many):

- "Angajati"

AngajatID, Nume, Prenume, Data\_Nasterii, Gen, Functie, Telefon

- "Clienti"

ClientID, Nume, Prenume, Telefon, Istoric\_lucrari

- "Masini"

MasinaID, ClientID, Marca, Model, An\_fabricatie,  
Lucrare\_in\_desfasurare

- "Lucrari"

LucrareID, ClientID, MasinaID, Nume\_lucrare, Piese\_necesare, Stare

- "Piese"

PiesaID, Nume\_piesa, Producator, Model, Stoc

- “Lucrari\_piese”

LucrareID, PiesaID, Cantitate

Intre tabelele “Lucrari” si “Piese” exista o relatie N:N, formata cu ajutorul tabelului “Lucrari\_Piese” deoarece un o lucrare poate necesita mai multe piese si o piesa din stoc cu un anume ID poate este folosita la mai multe lucrari.

### **Relatii intre tabele**

#### Cienti – Masini – 1:N

- Un client poate avea mai multe masini cu care a venit la service

#### Cienti – Lucrari – 1:N

- Unui client I se pot face mai multe lucrari in acelasi timp, sau poate avea mai multe lucrari finalizate in istoric

#### Masini – Lucrari – 1:N

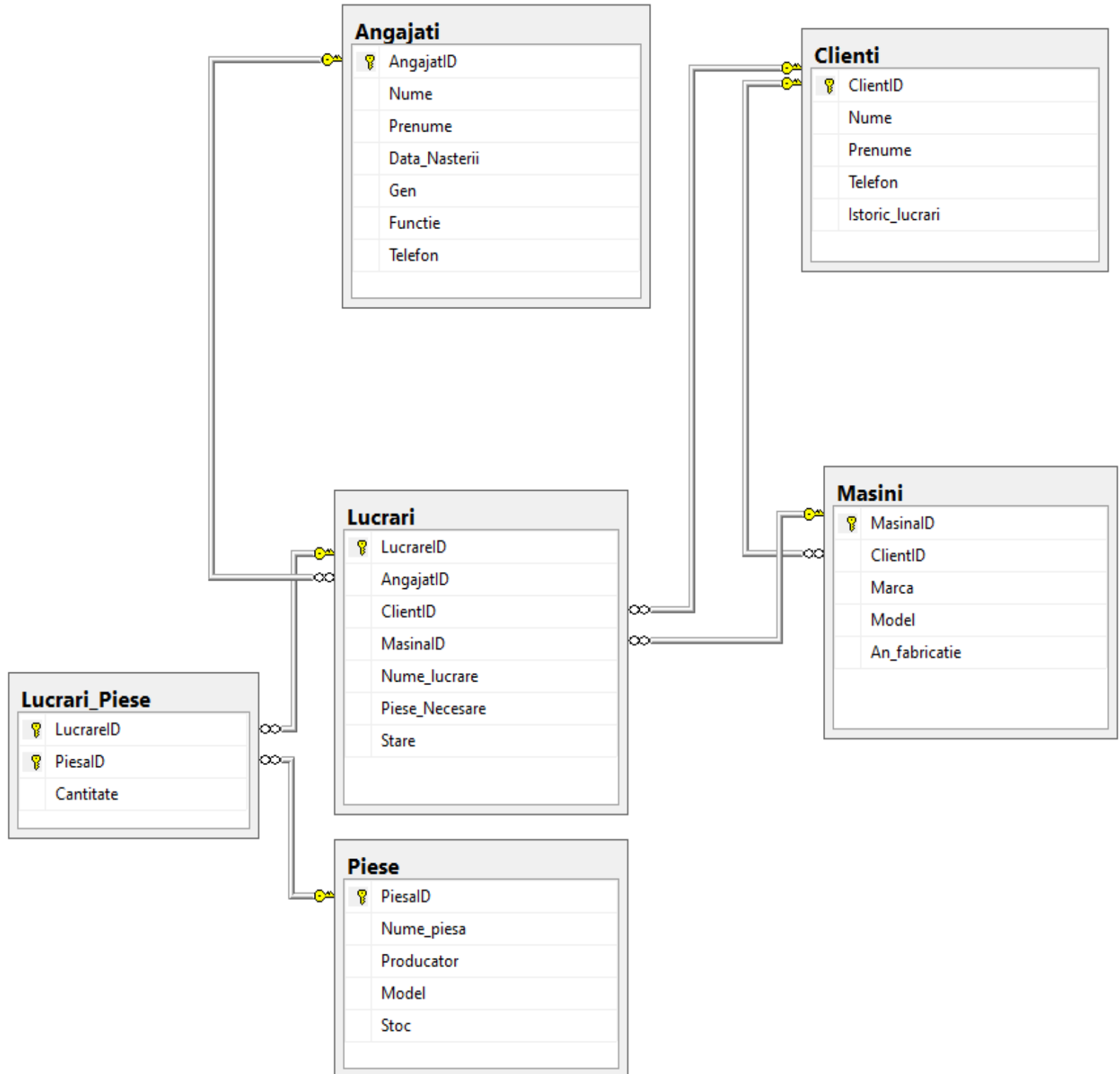
- masina poate avea mai multe lucrari finalizate, sau una in desfasurare si restul finalizate

#### Angajati – Lucrari – 1:N

- un angajat poate avea mai multe lucrari in desfasurare sau finalizate

#### Lucrari – Piese – N:N

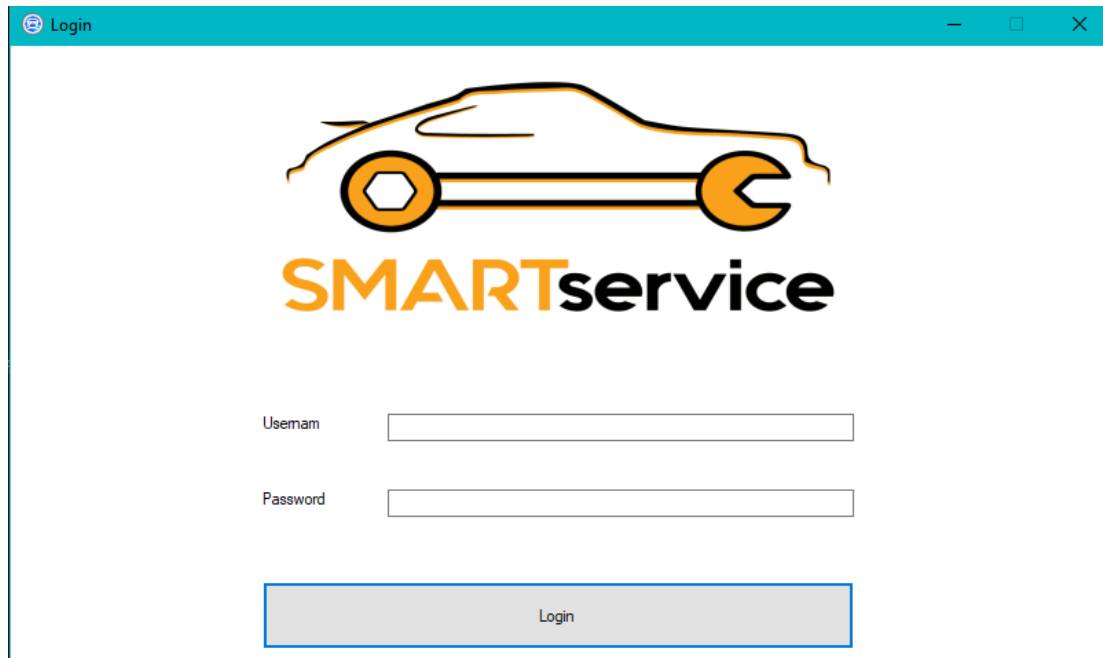
- lucrare poate necesita mai multe piese
- o piesa este folosita la mai multe lucrari



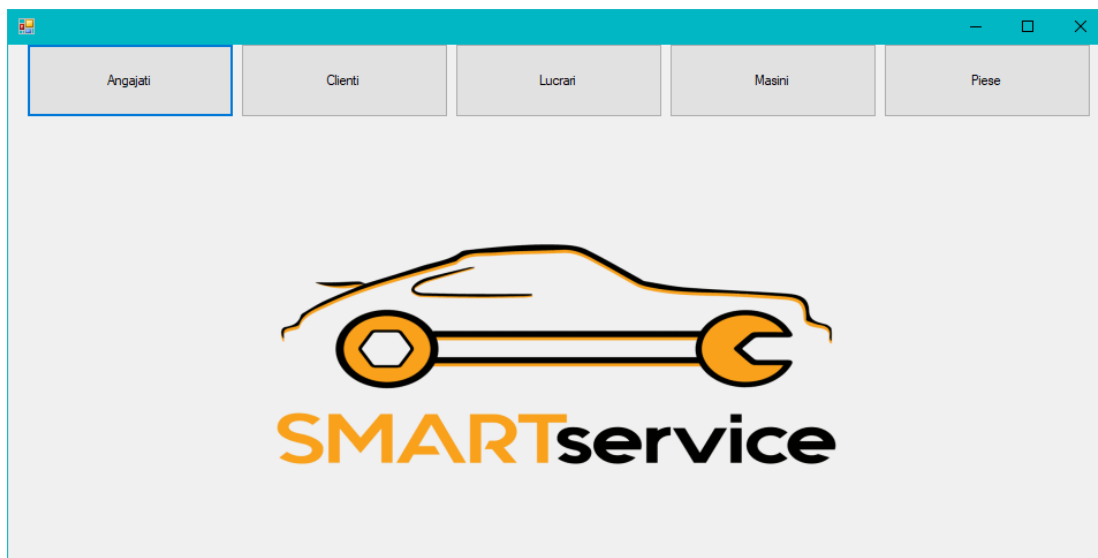
## Interfata Grafica

Interfata aplicatiei a fost realizata in .NET cu ajutorul Windows Forms.

Aplicatia are o pagina de login



Dupa logarea utilizatorului, se prezinta pagina principala de unde se pot alege tabele in care sa se faca anumite operatii.



### Au fost realizate urmatoarele queries:

-- interogare complexa 1  
-- Nr de lucrari ale fiecarui angajat

```
SELECT A.AngajatID, A.Nume, A.Prenume as Nume, (  
  SELECT COUNT(*) FROM Lucrari L  
  WHERE L.AngajatID = A.AngajatID)  
  AS Numar_Lucrari  
FROM Angajati A
```

-- interogare complexa 2 cu parametru variabil  
-- numele si prenumele angajatilor care au lucraza la lucrarea cu starea X

```
SELECT Distinct A.Nume, A.Prenume  
from Angajati A,(  
  select L.Stare, L.AngajatID  
  from Lucrari L  
  where L.Stare='final')  
  as St  
  where A.AngajatID=St.AngajatID
```

-- interogare complexa 3  
-- nr de masini ale fiecarui client  
SELECT C.ClientID, C.Nume, C.Prenume as Nume, (  
 SELECT COUNT(\*) FROM Masini M  
 WHERE M.ClientID = C.ClientID)  
 AS Numar\_Masini  
FROM Clienti C

-- interogare complexa 4  
-- in ce an s-au fabricat cele mai multe masini  
select M.An\_fabricatie, COUNT(M.MasinaID) as nr\_masini  
from Masini M  
group by M.An\_fabricatie  
having COUNT( M.MasinaID)>=(  
 select top 1 COUNT(M1.MasinaID)  
 from Masini M1  
 group by M1.An\_fabricatie  
 order by COUNT(M1.MasinaID) DESC  
)

-- interogare simpla 1  
-- numele si prenumele clientilor care au masini de marca X  
select C.Nume, C.Prenume, M.Marca  
from Clienti C inner join Masini M on C.ClientID=M.ClientID  
where M.Marca='Audi'  
group by C.Nume, C.Prenume, M.Marca

Croitoru Robert-Valentin

332AB

```
-- interogara simpla 2
-- numele si prenumele clientilor care au lucrari nefinalizate
select C.Nume, C.Prenume, L.Stare
from Clienti C inner join Lucrari L on C.ClientID=L.ClientID
where L.Stare='nefinal'
group by C.Nume, C.Prenume, L.Stare

-- interogare complexa 3
-- numele si prenumele angajatilor care nu si-au terminat lucrarile
select A.Nume, A.Prenume, L.Stare
from Angajati A inner join Lucrari L on A.AngajatID=L.AngajatID
where L.Stare='nefinal'
group by A.Nume, A.Prenume, L.Stare

-- interogare simpla 4
-- lucrarile care sunt desfasurate doar de mecanici

select L.LucrareID, L.Nume_Lucrare, A.Functie
from Lucrari L inner join Angajati A on A.AngajatID=L.AngajatID
where A.Functie='mecanic'
group by L.LucrareID, L.Nume_lucrare, A.Functie

INSERT INTO Lucrari_Piese(LucrareID, PiesaID)
SELECT L.LucrareID, P.PiesaID
FROM Lucrari L, Piese P
WHERE L.Piese_Necesare=P.Nume_piesa
```

## Portiuni de cod unde s-au folosit queries:

```
dt = new DataTable();
con.Open();
adpt = new SqlDataAdapter("SELECT C.ClientID, C.Nume, C.Prenume as Nume, ( SELECT COUNT(*) FROM Masini M WHERE M.ClientID = C.ClientID) AS Numar_Masini FROM Clienti C", con)
adpt.Fill(dt);
dataGridView1.DataSource = dt;
con.Close();
```

```
try
{
    con.Open();
    cmd = new SqlCommand("delete from Clienti where ClientID='" + ID + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Clientul a fost sters!");

    con.Close();
    Refresh();
    display();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

### Complex 1

```
dt = new DataTable();
con.Open();
adpt = new SqlDataAdapter("select M.An_fabricatie, COUNT(M.MasinaID) as nr_masini from Masini M group by M.An_fabricatie having COUNT(M.MasinaID) >= ( select top 1 COUNT(M1.MasinaID) from Masini M1 group by M1.An_fabricatie order by COUNT(M1.MasinaID) DESC )", con);

adpt.Fill(dt);
dataGridView1.DataSource = dt;
con.Close();
```

Croitoru Robert-Valentin

332AB

Complex 2

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("SELECT C.ClientID, C.Nume, C.Prenume as Nume,
( SELECT COUNT(*) FROM Masini M WHERE M.ClientID = C.ClientID) AS Numar_Masini FROM
Clienti C", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```

Complex 3

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("SELECT Distinct A.Nume, A.Prenume from
Angajati A, (select L.Stare, L.AngajatID from Lucrari L where L.Stare = 'final') as St
where A.AngajatID = St.AngajatID", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```

Complex 4

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("SELECT A.AngajatID, A.Nume, A.Prenume as Nume,
(SELECT COUNT(*) FROM Lucrari L WHERE L.AngajatID = A.AngajatID) AS Numar_Lucrari FROM
Angajati A ", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```

Simple 1

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("select C.Nume, C.Prenume, M.Marca from Clienti
C inner join Masini M on C.ClientID = M.ClientID where M.Marca = 'Audi' group by C.Nume,
C.Prenume, M.Marca", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```

Simple 2

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("select A.Nume, A.Prenume, L.Stare from
Angajati A inner join Lucrari L on A.AngajatID = L.AngajatID where L.Stare = 'nefinal'
group by A.Nume, A.Prenume, L.Stare", con);
```



Croitoru Robert-Valentin

332AB

```
adpt.Fill(dt);
dataGridView1.DataSource = dt;
con.Close();
```

Simple 3

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("select C.Nume, C.Prenume, L.Stare from Clienti
C inner join Lucrari L on C.ClientID = L.ClientID where L.Stare = 'nefinal' group by
C.Nume, C.Prenume, L.Stare", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```

Simple 4

```
dt = new DataTable();
    con.Open();
    adpt = new SqlDataAdapter("select L.LucrareID, L.Nume_Lucrare, A.Functie,
A.Nume, A.Prenume from Lucrari L inner join Angajati A on A.AngajatID = L.AngajatID where
A.Functie = 'mecanic' group by L.LucrareID, L.Nume_lucrare, A.Functie, A.Nume,
A.Prenume", con);

    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
```