

## Proiect Baze de Date

Aplicație pentru gestiunea unei companii care oferă servicii de mentenanță pentru echipamente industriale

### Cuprins

1.	Descrierea Cerințelor .....	2
2.	Structura Bazei de Date .....	4
2.1.	Schema Relațională .....	4
2.2.	Tipuri de date și Constrângeri de Integritate .....	5
2.2.1.	User .....	5
2.2.2.	Angajat .....	5
2.2.3.	Tichet .....	5
2.2.4.	User_Echip .....	6
2.2.5.	Echipament .....	6
2.2.6.	Client .....	6
2.2.7.	Luna_Echip .....	7
2.2.8.	Luna .....	7
2.2.9.	ListaOper .....	7
2.2.10.	LisaOper_Oper .....	7
2.2.11.	Operatie .....	7
2.2.12.	ListaPiese .....	8
2.2.13.	ListaPiese_Piese .....	8
2.2.14.	Piesa .....	8
3.	Descrierea aplicatiei .....	9
4.	Interogari .....	13
4.1.	Interogari simple .....	13
4.2.	Interogari complexe .....	15

## 1. Descrierea Cerințelor

Compania se ocupă cu prestarea de servicii de mentenanță, pentru echipamente industriale, mai multor clienți. Un client este caracterizat prin denumire, stradă și număr.

Un client are cel puțin un echipament la care se face mentenanță. Un echipament are câmpurile: cod, denumire, tip, preventivă.

Aplicația se utilizează pe bază de user, deci fiecare user are un profil angajat, dar legăturile se fac către user, deoarece avem nevoie de nivel acces. Un user are câmpurile: username, parolă și nivel acces, iar un angajat are câmpurile: nume, prenume, CNP, stradă, număr, sex, data nașterii, salariu.

Fiecare echipament are cel puțin un responsabil, care este un user, iar fiecare user poate lucra la cel puțin un echipament, reținându-se data începerii mentenanței unui user la un echipament.

Se poate emite un ticket cu datele specifice: dată, tip intervenție, categorie, descriere, status și termen limită. Un user poate emite mai multe tickete și un echipament poate avea mai multe tickete care se referă la el.

Fiecare echipament are o listă de operații și una de piese, fiecare cu o denumire și versiune. O listă de operații are mai multe operații, fiecare cu câmpurile: denumire, instrucțiune, scule speciale, aplicativitate preventivă, aplicativitate corectivă și durată.

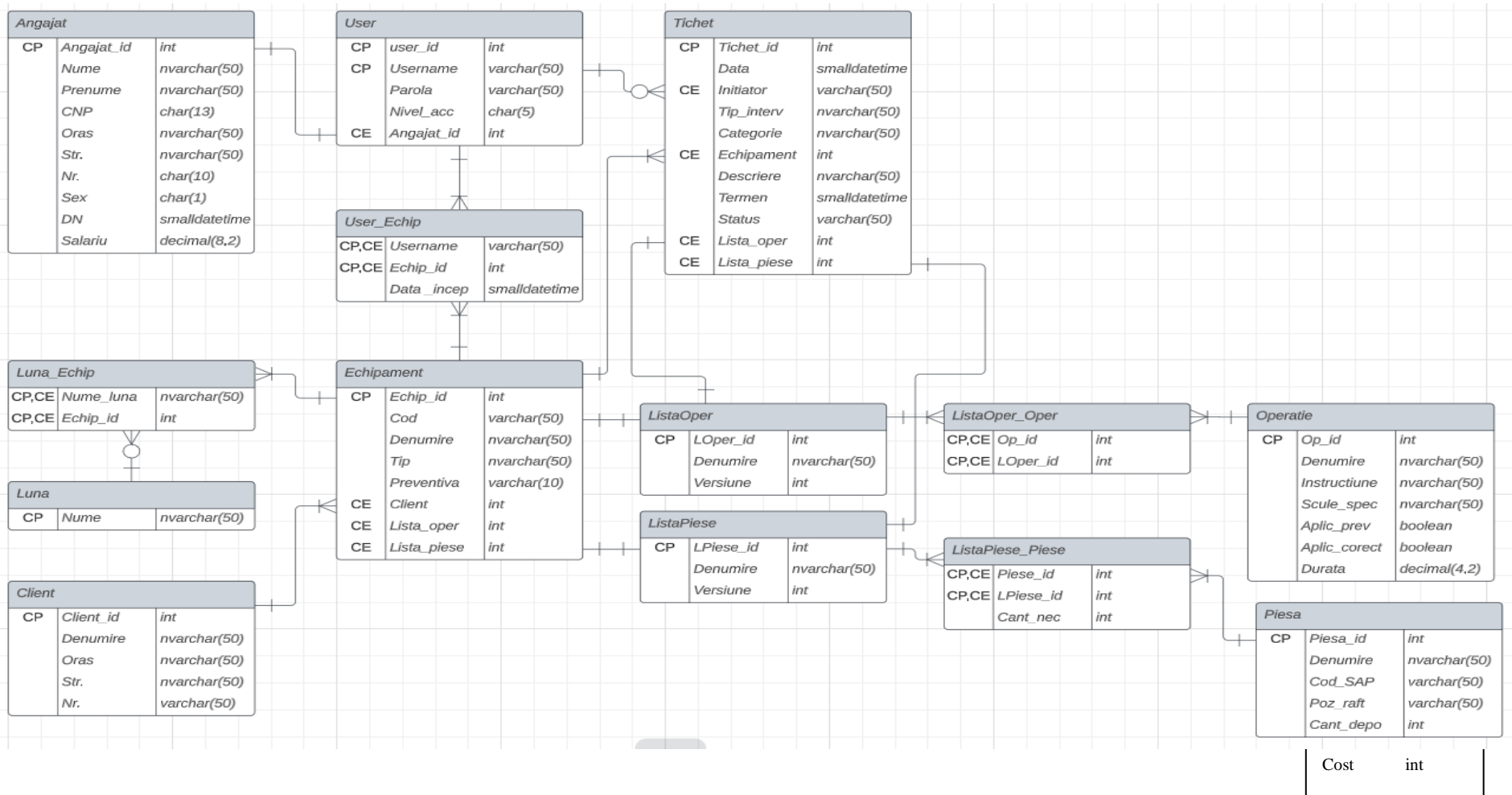
Fiecare listă de piese are mai multe piese, fiecare cu câmpurile: denumire, cod\_SAP, poziție raft, cost și cantitate în depozit, memorându-se, de asemenea, cantitatea necesară din fiecare piesă pentru fiecare listă de piese.

Fiecare ticket are, de asemenea, o listă de piese și o listă de operații, fiind de fapt, un tabel al tuturor acțiunilor ce s-au desfășurat sau se desfășoară în companie. Ținem seama astfel de piesele și operațiile exacte care s-au utilizat/realizat la fiecare operație.

Fiecare echipament are cel puțin o lună în care se face mentenanța, dar într-o lună se poate face mentenanța la mai multe echipamente, deci avem și un tabel al lunilor, fiecare intrare având doar atributul nume.

## 2. Structura Bazei de Date

### 2.1. Schema Relațională



## 2.2. Tipuri de date și Constrângeri de Integritate

### 2.2.1. User

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
user_id	X	X	X			
username	X	X	X			
parola	X					
nivel_acc						
angajat_id	X	X				X

### 2.2.2. Angajat

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
angajat_id	X	X	X			
nume	X					
prenume	X					
cnp		X				
oras						
strada						
numar						
sex	X			([Sex]='F' OR [Sex]='M')	F	
data_n	X					
salariu					0	

### 2.2.3. Tichet

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
tichet_id	X	X	X			
data	X					
initiator	X					X
tip_interv	X					
categorie						
echipament	X					X

descriere						
termen	X					
status	X					
lista_oper						X
lista_piese						X

#### 2.2.4. User\_Echip

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
username	X	X	X			X
echip_id	X	X	X			X
data_incep	X					

#### 2.2.5. Echipament

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
echip_id	X	X	X			
cod	X	X				
denumire	X					
tip						
preventiva						
client	X					X
lista_oper	X					X
lista_piese	X					X

#### 2.2.6. Client

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
client_id	X	X	X			
denumire	X					
oras						
strada						
numar						

### 2.2.7. Luna\_Echip

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
nume_luna	X	X	X			X
echip_id	X	X	X			X

### 2.2.8. Luna

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
nume	X	X	X	pot verifica toate luniile		

### 2.2.9. ListaOper

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
loper_id	X	X	X			
denumire	X					
versiune	X				1	

### 2.2.10. LisaOper\_Oper

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
operatie_id	X	X	X			X
loper_id	X	X	X			X

### 2.2.11. Operatie

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
operatie_id	X	X	X			
denumire	X					
instructiune						
scule_spec						
aplic_prev	X					
aplic_corect	X					

durata	X			([durata]>0)	0	
--------	---	--	--	--------------	---	--

### 2.2.12. ListaPiese

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
lpiese_id	X	X	X			
denumire	X					
versiune	X				1	

### 2.2.13. ListaPiese\_Piesa

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
piesa_id	X	X	X			X
lpiesa_id	X	X	X			X
cant_nec	X			([cant_nec]>0)	1	

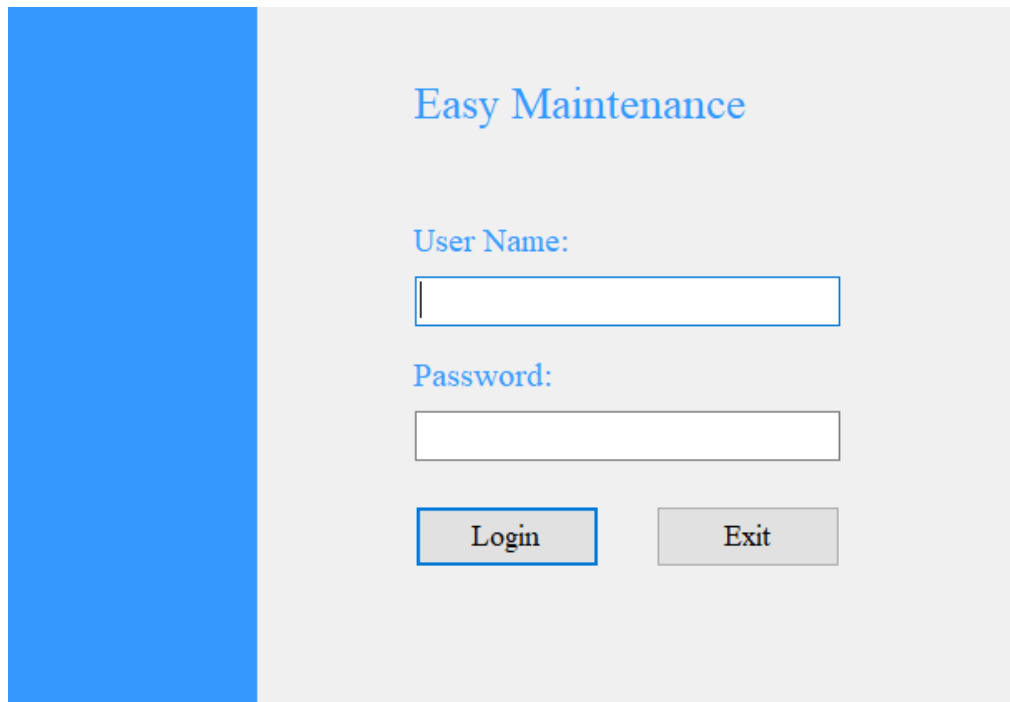
### 2.2.14. Piesa

Denumire	Not Null	Unique	Primary Key	Check	Default	Foreign Key
piesa_id	X	X	X			
denumire	X					
cod_sap	X	X				
poz_raft						
cant_depo				([cant_depo]>0)		
cost	x					



### 3. Descrierea aplicatiei

Avem o aplicatie destul de simpla. Prima parte este partea de logare.



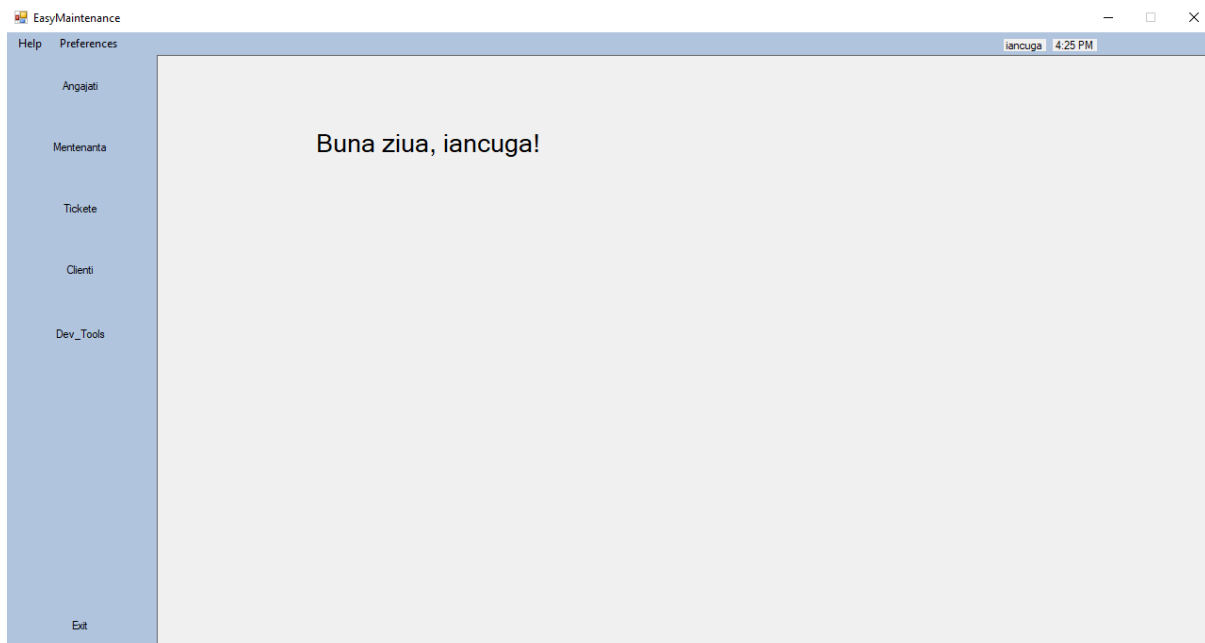
Easy Maintenance

User Name:

Password:

Login Exit

Daca ne-am conectat cu succes, ajungem in meniul principal unde este ilustrat username-ul cu care ne-am conectat si ora la care ne-am logat in cont.



Avem din acest punct, mai multe butoane pe partea stanga a meniului, care o sa ne deschida o alta parte a aplicatiei.

The screenshot shows the 'EasyMaintenance' application window. The left sidebar has a menu with 'Angajati' (Employees) selected. The main area displays a form for adding or editing employees. It includes input fields for 'username' and 'nivel acces' (access level), a 'Reset' button, and a table of existing employees. The table has columns 'username' and 'nivel\_acc'. The data in the table is as follows:

username	nivel_acc
admin	R
lancuga	RW
banescua	R
vasiles	R
stani	RW

Below the table is a large grey rectangular area. To the right of the table are buttons: 'Adaugare' (Add), 'Modificare' (Modify), 'Stergere' (Delete), and 'Cautare' (Search). Further right are input fields for 'Echipamente in mentenanta' (Equipment in maintenance), 'Incepere activitate' (Start activity), and 'Primul client' (First client). Below these are 'Operatii preferate' (Preferred operations) with a 'Denumire' (Name) input field and a 'Numar aplicari' (Number of applications) input field. The top right of the window shows the user 'lancuga' and the time '4:25 PM'.

Avem aici partea de angajati unde avem cateva statistici pentru fiecare client.

The screenshot shows the 'EasyMaintenance' application window with the 'Mentenananta' (Maintenance) section selected in the sidebar. The main area displays a table with the following columns: 'Echipamente' (Equipment), 'Piese' (Parts), 'Lista piese' (List of parts), 'Operatii' (Operations), and 'Lista Operatii' (List of operations). The table is currently empty.

Partea de mentenanta este cea mai importanta si are alte 5 submeniuri destul de similare intre ele.

denumire	cod_sap	poz_raft	cant_depo	cost
Filtru ulei 100x10	11	A1	65	55.50
Filtru ulei 100x20	22	A2	25	75.50
Termocuplu	33	C1	30	3.00
Surub 5x12	44	D10	124	1.50
Filtru motorina	55	A6	10	121.50
Filtru ulei 100x10	66	A3	10	46.88
Pulita 10x10	100	Z4	158	0.40
Saiba 10x10	355	Z3	160	0.60
test	101	test	100	150.00

Avem aici administrarea pieselor. Este o interfata destul de simpla, deci nu am foarte multe de explicat.

denumire	versiune
Kit intretinere IC Tester IE200	1
Kit intretinere Statie Calibrare HL	2
Kit intretinere Presa Hidraulica	5
Kit intretinere Impimanta 3D	2
Kit intretinere Cabina Universala LP H3	2
Kit reparatie IC Tester IE200	1

Avem aici partea de liste de piese care este putin mai complicata, deoarece avem mai multe actiuni pe care putem sa le luam pentru administrarea lor. Avem si partea de statistici in coltul din dreapta jos. Administrarea de operatii si de lista de operatii sunt foarte similare.

The screenshot shows the 'EasyMaintenance' application interface. On the left is a sidebar with navigation links: Angajati, Mentenanta, Tickete (highlighted), Clienti, and Dev\_Tools. The main area displays a table of tickets with columns: data, initiator, tip\_interv, categorie, echipament, descriere, termen, status, lista\_oper, and lista\_piese. The table contains five rows of data. Below the table is a form for creating a new ticket, with fields for Data, Initiator, Tip interventie, Categorie, Descriere, Termen, Denumire echipament, Client, Status, Lista piese, and Lista operatii.

data	initiator	tip_interv	categorie	echipament	descriere	termen	status	lista_oper	lista_piese
11/20/2023	iancuga	Corectiva	Defect electronic	1	Rezistentă arsă	11/30/2023	Activ	6	
11/21/2023	iancuga	Corectiva	Defect Mecanic	1	Filtru rupt	11/30/2023	Activ	7	
11/22/2023	iancuga	Corectiva	Defect electronic	1	Condensator ars	11/30/2023	Activ	8	
11/20/2022	ianescova	Preventiva		4		11/30/2022	Inactiv	1	6
10/5/2021	vasiles	Preventiva		7		10/30/2022	Inactiv	2	7

Data: 11/20/2023 12:00:00 AM  
Initiator: iancuga  
Tip interventie: Corectiva  
Categorie: Defect electronic  
Descriere: Rezistentă arsă  
Termen: 11/30/2023 12:00:00  
Denumire echipament: IC Tester IE200  
Client: Steinell  
Status: Activ  
Lista piese:  
Lista operatii: Plan corectie IC Tester

Avem aici partea de tickete cu preluare automata de date din tabelele corespunzatoare.

The screenshot shows the 'EasyMaintenance' application interface, specifically the 'Dev\_Tools' section. It features a sidebar with navigation links: Angajati, Mentenanta, Tickete, Clienti, and Dev\_Tools (highlighted). The main area contains a section for executing SQL queries, with buttons for 'Open Connection', 'Close Connection', and 'Execute Query'. The 'Connection status' is displayed as 'NOT GOOD'. Below the buttons is a large text area for entering the query, and a 'Reset Query' button.

Open Connection Close Connection Connection status: NOT GOOD  
Query:   
Execute Query  
Reset Query

Avem si o sectiune unde putem introduce automat comenzi SQL pentru a vedea raspunsul de la server.

## 4. Interogari

### 4.1. Interogari simple

Partea de logare:

```
String query = "SELECT * FROM User1 WHERE username = '" +  
textUsername1.Text + "' AND parola = '" + textPassword1.Text + "'";
```

Citim aici username-ul si parola pentru a incerca conectarea la aplicatie.

Partea de mainmenu nu are select-uri.

Partea de clienti:

```
String query = "SELECT * FROM Client";  
String query1 = "SELECT COUNT(*) FROM Echipament WHERE client  
= '" + idBox + "'";
```

```
String query = "SELECT * FROM Client WHERE denumire LIKE '" +  
denumire.Text + "%" + "' AND oras LIKE '" + oras.Text + "%" + "'";
```

Caut toti clientii dupa denumire si oras.

```
String query2 = "SELECT COUNT(*) FROM Tichet INNER JOIN  
Echipament ON Tichet.echipament = Echipament.echip_id WHERE  
Echipament.client = '" + idBox + "'";
```

Aflam aici numarul de tichete pe care le are un anumit client

```
String query3 = "SELECT COUNT(*) FROM Tichet INNER JOIN  
Echipament ON Tichet.echipament = Echipament.echip_id WHERE  
Echipament.client = '" + idBox + "' AND Tichet.status LIKE 'Activ'";
```

Aflam aici numarul de tichete active pe care le are un anumit client

```
String query4 = "SELECT TOP 1 User_Echip.data_incep FROM  
Echipament INNER JOIN User_Echip ON User_Echip.echip_id =  
Echipament.echip_id WHERE Echipament.client = '" + idBox + "'  
ORDER BY User_Echip.data_incep ASC";
```

Aflam aici data la care un primul angajat a inceput sa lucreze la un client

Partea de angajati(user):

```
String query1 = "SELECT COUNT(*) FROM User_Echip  
WHERE User_Echip.username LIKE '" + username.Text + "'";
```

Aflam aici numarul de echipamente in mentenanta

```
String query2 = "SELECT TOP 1 User_Echip.data_incep FROM  
User_Echip WHERE User_Echip.username LIKE '" + username.Text + "  
"ORDER BY User_Echip.data_incep ASC";
```

Aflam aici data cand un angajat si-a inceput activitatea

```
String query3 = "SELECT TOP 1 Client.Denumire FROM Client INNER  
JOIN Echipament ON Echipament.client=Client.client_id INNER JOIN  
User_Echip ON User_Echip.echip_id = Echipament.echip_id WHERE  
User_Echip.username LIKE '" + username.Text + "' ORDER BY  
User_Echip.data_incep ASC";
```

Aflam aici primul client la care a lucrat un angajat

Partea de Lista\_Piese:

```
query = "SELECT Piese.denumire, Piese.cod_sap, Piese.poz_raft,  
Piese.cant_depo, Piese.cost, ListaPiese_Piese.cant_nec FROM  
ListaPiese_Piese JOIN Piese ON ListaPiese_Piese.piesa_id =  
Piese.piesa_id WHERE ListaPiese_Piese.lpiesa_id = '" + idBoxLista +  
"'"
```

Aflam aici piesele care apartin unei anumite liste

```
String query1 = "SELECT ISNULL(Suma,0) FROM (SELECT  
SUM(ListaPiese_Piese.cant_nec*Piese.cost) AS Suma FROM  
ListaPiese_Piese JOIN Piese ON ListaPiese_Piese.piesa_id =  
Piese.piesa_id WHERE ListaPiese_Piese.lpiesa_id = '" + idBoxLista + "')  
AS R";
```

Aflam aici suma costurilor pieselor dintr-o anumita lista.

```
String query2 = "SELECT ISNULL(Maxim,0) FROM (SELECT  
MAX(ListaPiese_Piese.cant_nec*Piese.cost) AS Maxim FROM  
ListaPiese_Piese JOIN Piese ON ListaPiese_Piese.piesa_id =
```

```
Piese.piesa_id WHERE ListaPiese_Piese.lpiesa_id = "" + idBoxLista + "")  
AS R";
```

Aflam aici pretul maxim dintr-o anumita lista.

```
query = "SELECT Operatie.denumire, Operatie.instructiune,  
Operatie.scule_spec, Operatie.aplic_prev, Operatie.aplic_corect,  
Operatie.durata FROM ListaOper_Oper JOIN Operatie ON  
ListaOper_Oper.operatie_id = Operatie.operatie_id WHERE  
ListaOper_Oper.loper_id = "" + idBox + """;
```

Aflam aici operatiile care fac parte dintr-o anumita lista de operatii.

```
String query1 = "SELECT ISNULL(Suma,0) FROM (SELECT  
SUM(Operatie.durata) AS Suma FROM ListaOper_Oper JOIN Operatie  
ON ListaOper_Oper.operatie_id = Operatie.operatie_id WHERE  
ListaOper_Oper.loper_id = "" + idBox + "") AS R";
```

Aflam aici suma duratelor operatiilor dintr-o lista de operatii.

Partea de Tichete:

```
String query4 = "SELECT Client.denumire FROM Echipament INNER  
JOIN Client ON Echipament.client = Client.client_id WHERE  
Echipament.echip_id = "" + idBox + """;
```

Aflam aici denumirea clientului pe care un anumit echipament in are.

## 4.2. Interogari complexe

Partea de Clienti:

```
String query5 = "SELECT Luna_Echip.num_e_luna,  
Echipament.denumire FROM Luna_Echip JOIN Echipament ON  
Luna_Echip.echip_id = Echipament.echip_id WHERE Echipament.client  
= "" + idBox + "" AND Luna_Echip.num_e_luna IN (SELECT TOP 1  
LE1.num_e_luna FROM Luna L1 JOIN Luna_Echip LE1 ON L1.num_e_luna =  
LE1.num_e_luna JOIN Echipament E1 ON LE1.echip_id = E1.echip_id  
WHERE E1.client = "" + idBox + "" AND L1.index_luna >= "" + thisMonth  
+ "" ORDER BY L1.index_luna);
```

Aflam aici echipamentele la care trebuie sa se faca mentenanta, ale unui anumit client din urmatoarea luna din anul curent, si evident, si luna.

### Partea de Angajati(user):

```
String query4 = "SELECT R.denumire, R.NumarOp1 FROM (SELECT  
Operatie.denumire, COUNT(*) NumarOp1 FROM Tichet JOIN  
ListaOper_Oper ON lista_oper = loper_id JOIN Operatie ON  
ListaOper_Oper.operatie_id = Operatie.operatie_id WHERE initiator  
LIKE '" + username.Text + "' GROUP BY Operatie.denumire) AS R  
WHERE R.NumarOp1 = ( SELECT R1.NumarOp FROM (SELECT TOP  
1 O1.denumire, COUNT(*) AS NumarOp FROM Tichet T1 JOIN  
ListaOper_Oper LOO1 ON T1.lista_oper = LOO1.loper_id JOIN Operatie  
O1 ON LOO1.operatie_id = O1.operatie_id WHERE initiator LIKE '" +  
username.Text + "' GROUP BY O1.denumire ORDER BY NumarOp  
DESC) AS R1)";
```

Am aflat aici operatiile care apar de cele mai multe ori in tichetele unui anumit user si numarul de aparitii ale acestora.

### Partea de Lista\_Piese:

```
String query3 = "SELECT ISNULL(Medie,0) FROM (SELECT  
AVG(ListaPiese_Piese.cant_nec*Piese.cost) AS Medie FROM  
ListaPiese_Piese JOIN Piese ON ListaPiese_Piese.piesa_id =  
Piese.piesa_id WHERE ListaPiese_Piese.lpiesa_id = '" + idBoxLista + "')  
AS R";
```

Aflam media costurilor dintr-o lista de piese.

```
String query = "SELECT * FROM Piese WHERE Piese.piesa_id NOT IN  
( SELECT P1.piesa_id FROM Piese P1 INNER JOIN ListaPiese_Piese  
LPP1 ON P1.piesa_id = LPP1.piesa_id WHERE LPP1.lpiesa_id = '" +  
idBoxLista + "')";
```

Aflam aici lista pieselor care nu sunt deja intr-o anumita lista.

### Partea de Lista\_Operatii:

```
String query2 = "SELECT ISNULL(Maxim,0) FROM (SELECT  
Max(Operatie.durata) AS Maxim FROM ListaOper_Oper JOIN Operatie  
ON ListaOper_Oper.operatie_id = Operatie.operatie_id WHERE  
ListaOper_Oper.loper_id = '" + idBox + "') AS R";
```

Aflam aici cea mai mare durata pe care o are o operatie dintr-o anumita lista.