

Proiect Informatica Aplicata 4

Belceanu Andra-Maria(321CA), Duca Andrei-
Rares(321CA), Potoceanu Ana-Maria(321CA),
Obreja Ana-Maria(321CA)

Gestionarea datelor interne in cadrul unei companii de curierat

Avand in vedere faptul ca, in prezent, volumul de date a unei companii de curierat este foarte mare, proiectul nostru isi propune crearea unei aplicatii pentru:

1). Imbunatatirea sistemului de organizare pe baza de dosare si documente ce se pot pierde/deteriora foarte usor;

2). Simplificarea modului de lucru a administratorului unei astfel de firme si

eventual, evitarea posibilelor probleme, precum pierderea unui colet, livrarea pachetului gresit etc.

Aceasta aplicatie are rolul de a stoca informatii referitoare la coletele ce ajung la firma, despre persoanele care apeleaza la serviciile firmei (expeditorii si destinatarii), dar si despre angajatii acesteia (curierii). Cateva din conceptele functionarii acestei aplicatii sunt: accesarea anumitor informatii in functie de criteriul dorit, modificarea datelor si vizualizarea acestora.

TABELE

Referitor la proiectarea bazei de date, am realizat 8 tabele initiale, adica: Clienti, Curieri, Colete, Localizari, ConturiAdministratori, respectiv 2 tabele de legatura CurieriZone si ClientiColete.

Astfel, clientii sunt caracterizati de un ID unic, de nume, prenume, telefon si CNP. Apar si informatii legate de adresa acestora, precum Strada+Numar(campul Adresa), Localitate, Judet, Tara si CodPostal, iar daca acesta este trimis de o firma, apare si numele companiei. De mentionat faptul ca apare si campul TipClient pentru a face diferenta intre Expeditori si Destinatari.

Detalii tehnice: - cheia primara ClientID, auto-increment

- CNP-ul este unic

- anumite campuri nu accepta NULL

- Check Constrains: campul TipClient poate sa fie Expeditor sau Destinatar

Urmatorul tabel reprezinta curierii ce ofera informatii precum: ID-ul curierului, numele, prenumele, CNP si telefonul acestuia, dar si date despre salariul si bonusul fiecarui curier angajat.

Detalii tehnice: - cheia primara CurierID, auto-increment

- CNP si telefon unice
- Anumite campuri nu accepta NULL

Coletele sunt caracterizate de ID, denumire, greutate, cost, data expedierii si livrarii, modalitate de plata, dimensiuni si un cod prin care se poate localiza coletul.

Detalii tehnice: - cheia primara ColetID, auto-increment

- Codul este unic
- Anumite campuri nu accepta NULL
- Check Constrains: campul Modalitate Plata poate sa fie Ramburs sau Card

Tabelul Localizari se refera la locul unui colet la un anumit moment de timp, informatii ce ajuta la o supraveghere mai buna a traseului fiecarui colet. In acest tabel identificam: LocalizareID, Cod, Adresa, Localitate si Tara.

Detalii tehnice: - cheia primara LocalizareID, auto-increment

- Anumite campuri nu accepta NULL

Urmeaza tabelul zonelor ce caracterizeaza zonele de actiune a fiecarui curier in parte, oferind informatii precum: ID zona, denumire, Localitate, siruta si tipul UAT.

Detalii tehnice: - cheia primara ZonaID, auto-increment

- Anumite campuri nu accepta NULL
- Campul Siruta este unic

Ultimul este tabelul ConturiAdministratori in care avem AdministratorID cheie primara, NumeUtilizator si Parola.

OBS: - in diagrama bloc mai apar si tabele de legatura

DIAGRAMA BAZEI DE DATE



STRUCTURA TABELELOR

CLIENTI

Column Name	Data Type	Allow Nulls
ClientID	int	<input type="checkbox"/>
Nume	nvarchar(50)	<input type="checkbox"/>
Prenume	nvarchar(50)	<input type="checkbox"/>
CNP	char(13)	<input type="checkbox"/>
Comanie	nvarchar(50)	<input checked="" type="checkbox"/>
Adresa	nvarchar(50)	<input checked="" type="checkbox"/>
Localitate	nvarchar(50)	<input type="checkbox"/>
Judet	nvarchar(50)	<input type="checkbox"/>
Tara	nvarchar(50)	<input type="checkbox"/>
CodPostal	char(10)	<input type="checkbox"/>
Telefon	char(15)	<input type="checkbox"/>
TipClient	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

CURIERI

Column Name	Data Type	Allow Nulls
CurierID	int	<input type="checkbox"/>
Nume	nvarchar(50)	<input type="checkbox"/>
Prenume	nvarchar(50)	<input type="checkbox"/>
CNP	char(13)	<input type="checkbox"/>
Telefon	char(15)	<input type="checkbox"/>
Salariu	decimal(8, 2)	<input type="checkbox"/>
Bonus	decimal(8, 2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

COLETE

Column Name	Data Type	Allow Nulls
ColetID	int	<input type="checkbox"/>
CurierID	int	<input type="checkbox"/>
LocalizareID	int	<input type="checkbox"/>
Denumire	nvarchar(50)	<input type="checkbox"/>
Greutate	int	<input checked="" type="checkbox"/>
Cost	decimal(8, 2)	<input type="checkbox"/>
DataExpedierii	smalldatetime	<input checked="" type="checkbox"/>
DataLivrarii	smalldatetime	<input type="checkbox"/>
ModalitatePlata	char(10)	<input checked="" type="checkbox"/>
Dimensiuni	nvarchar(50)	<input checked="" type="checkbox"/>
Cod	int	<input type="checkbox"/>
		<input type="checkbox"/>

LOCALIZARI

Column Name	Data Type	Allow Nulls
LocalizareID	int	<input type="checkbox"/>
Cod	int	<input checked="" type="checkbox"/>
Adresa	nvarchar(50)	<input type="checkbox"/>
Localitate	nvarchar(50)	<input checked="" type="checkbox"/>
Tara	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ZONE

Column Name	Data Type	Allow Nulls
ZonalID	int	<input type="checkbox"/>
Denumire	nvarchar(50)	<input type="checkbox"/>
Localitate	nvarchar(50)	<input type="checkbox"/>
CodPostal	char(10)	<input type="checkbox"/>
Siruta	char(6)	<input type="checkbox"/>
TipUAT	nvarchar(30)	<input type="checkbox"/>

CONTURI ADMINISTRATORI

Column Name	Data Type	Allow Nulls
AdministratorID	int	<input type="checkbox"/>
NumeUtilizator	nvarchar(50)	<input type="checkbox"/>
Parola	nvarchar(50)	<input type="checkbox"/>

TABEL LEGATURA: CLIENTI-COLETE

Column Name	Data Type	Allow Nulls
ColetID	int	<input type="checkbox"/>
ClientID	int	<input type="checkbox"/>

TABEL LEGATURA: CURIERI-ZONE

Column Name	Data Type	Allow Nulls
CurierID	int	<input type="checkbox"/>
ZonalID	int	<input type="checkbox"/>

RELATII:

	CLIENTI	COLET	CURIER	ZONA	LOCALIZARE	CONTURI
CLIENTI	-	1:N	-	-	-	-
COLET	1:N	-	1:1	-	1:1	-
CURIER	-	1:N	-	1:N	-	-
ZONA	-	-	1:N	-	-	-
LOCALIZARE	-	1:N	-	-	-	-
CONTURI	-	-	-	-	-	-

Clienti: Colet => 1:N + 1:N = N:N (many to many) (deoarece 1 colet are 1 destinatar si 1 expeditor)

Curier: Colet => 1:N + 1:1 = 1:N (one to many)

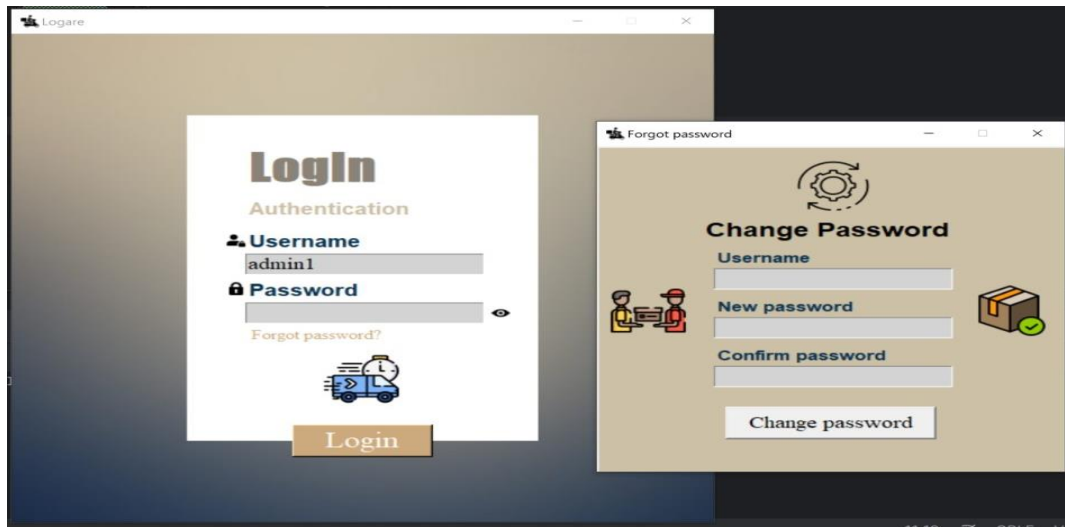
Localizare:Colet => 1:N + 1:1 = 1:N (one to many)

Curier:Zona => 1:N + 1:N = N:N (many to many)

FUNCTIONAREA APLICATIEI

Pentru implementarea interfeței grafice am folosit limbajul de programare Python, iar pentru conectarea și interacțiunea cu baza de date, am utilizat modulul pyodbc. Aplicatia este destinata doar administratorilor unei companii de curierat pentru a gestiona cat mai optim informatiile firmei.

Astfel, aplicatia porneste cu login-ul. Administratorul are in baza de date un cont de utilizator si o parola. In cazul in care acesta isi uita parola, are posibilitatea de a o schimba.



In momentul in care ne-am logat, aplicatia se conecteaza la baza de date.

```
1 usage
def login_function(self):

    if self.username_text.get() == "" or self.password_text.get() == "":
        messagebox.showerror( title: "Error", message: "All fields are required", parent=self.root)
    else:
        try:
            conectare = pyodbc.connect(string_de_conectare)
            cursor = conectare.cursor()
            sql_res = cursor.execute("SELECT * FROM ConturiAdministratori")
            sql_res = cursor.fetchone()
            if self.password_text.get() != sql_res[2] or self.username_text.get() != sql_res[1]:
                messagebox.showerror( title: "Error", message: "Invalid Username or Password", parent=self.root)
            else:
                second_root = Toplevel(self.root)
                SecondWindow(second_root)
                self.root.withdraw()
                second_root.mainloop()

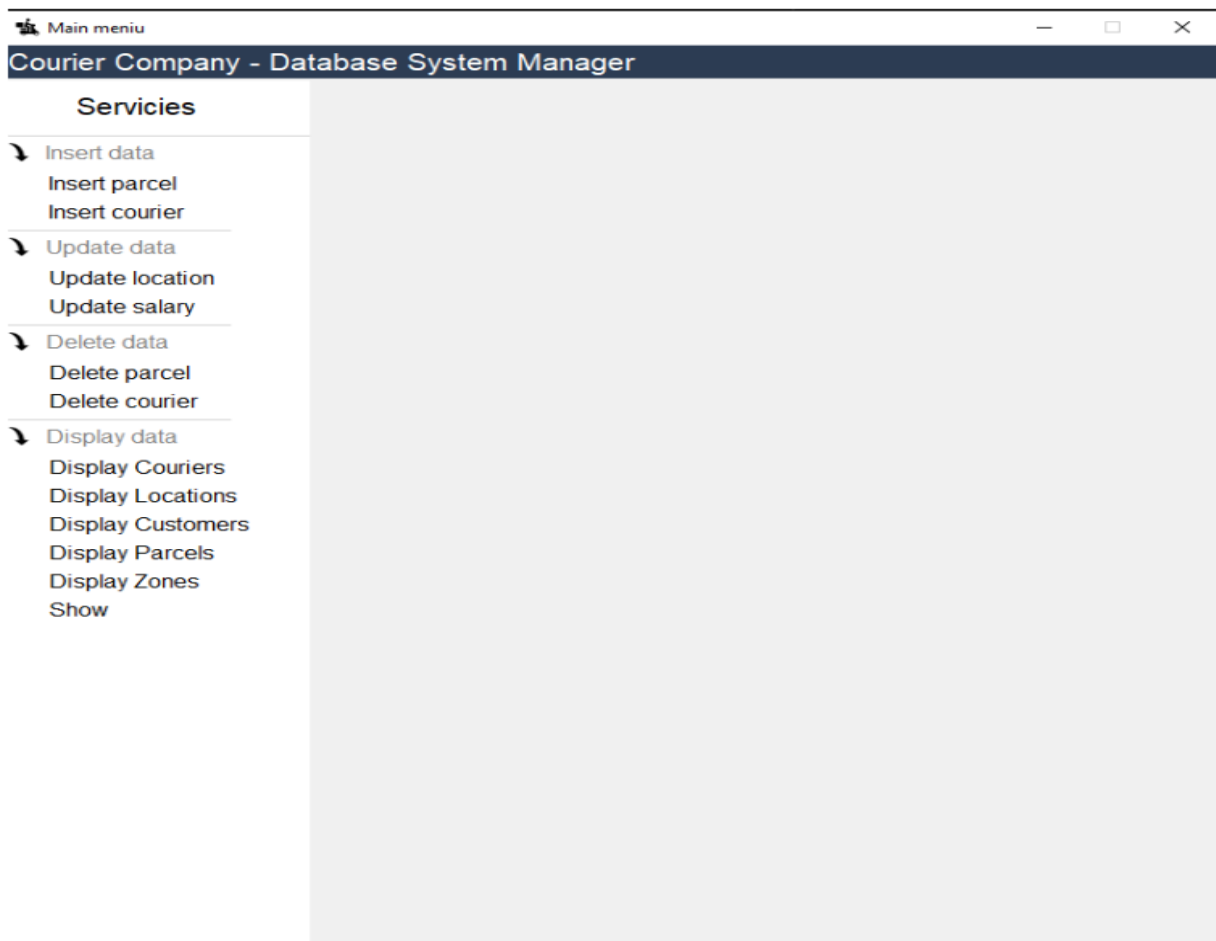
        except Exception as e:
            messagebox.showerror( title: "Error", message: f"Error Due to: {str(e)}", parent=self.root)
            self.root.destroy()

1 usage
def show_function(self):
    if self.password_text.cget("show") == '':
        self.password_text.config(show="*") # hidden password
```


Pentru a reseta parola, am folosit un UPDATE.

```
1 usage
def change_password(self):
    if self.username_id.get() == "" or self.new_password.get() == "" or self.confirm_new_password.get() == "":
        messagebox.showerror( title: "Error", message: "All fields are required!")
    elif self.new_password.get() != self.confirm_new_password.get():
        messagebox.showerror( title: "Error", message: "Confirm password is not identical with new password.", parent=self.root)
    else:
        try:
            conectare = pyodbc.connect(string_de_conectare)
            cursor = conectare.cursor()
            cursor.execute("SELECT NumeUtilizator FROM ConturiAdministratori")
            row = cursor.fetchone()
            if self.username_id.get() != row[0]:
                messagebox.showerror( title: "Error", message: "Please enter a valid username to reset the password.", parent=self.root)
            else:
                cursor.execute( sql: "UPDATE ConturiAdministratori SET Parola = ?", *params: self.new_password.get())
                cursor.commit()
                conectare.close()
                messagebox.showinfo( title: "Success", message: "Your password has been reset.", parent=self.root)
        except Exception as e:
            messagebox.showerror( title: "Error", message: f"Error Due to: {str(e)}", parent=self.root)
            self.root.destroy()
```

Dupa faza de logare si conectare la baza noastra de date, se deschide main menu.



INSERT — am realizat insert pentru clienti, colete si curieri, dar in acelasi timp am inserat si in tabelele aferente de legatura.

Parcel data		Courier data	
Courier	<input type="text"/>	Dispatch Data	<input type="text"/>
Name	<input type="text"/>	Delivery Data	<input type="text"/>
Weight	<input type="text"/>	Payment	<input type="text"/>
Location	<input type="text"/>	Size	<input type="text"/>
Cost	<input type="text"/>	Code Parcel	<input type="text"/>
Client data		Zone details	
First Name	<input type="text"/>	City	<input type="text"/>
Last Name	<input type="text"/>	Distinct	<input type="text"/>
CNP	<input type="text"/>	Country	<input type="text"/>
Company	<input type="text"/>	Zipcode	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Type	<input type="text"/>		
<input type="button" value="INSERT SENDER"/>		<input type="button" value="INSERT RECIPIENT"/>	
<input type="button" value="INSERT PARSEL"/>		<input type="button" value="INSERT ORDER"/>	
		<input type="button" value="INSERT COURIER"/>	

```
# Insert into table colet
cursor.execute("INSERT INTO Colet(CurierID, LocalizareID, Denumire, Greutate, Cost, DataExpedierii, DataLivrarii, ModalitatePlata, Dimensiuni, Cod)"
" VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", self.curier_id, self.location_id,
self.name_entry.get(), self.weight_entry.get(), self.cost_entry.get(),
self.dispatch_data_entry.get(), self.delivery_data_entry.get(),
self.payment_entry.get(), self.size_entry.get(), self.code_parcel_entry.get())
# commit changes
```

```
# insert into table clienti
cursor.execute("INSERT INTO Clienti(Nume, Prenume, CNP, Companie, Adresa, Localitate,"
" Judet, Tara, CodPostal, Telefon, TipClient) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)",
self.client_last_name_entry.get(), self.client_first_name_entry.get(),
self.cnp_entry.get(), self.company_entry.get(), self.address_entry.get(),
self.city_entry.get(), self.distinct_entry.get(), self.country_entry.get(),
self.zipcode_entry.get(), self.phone_entry.get(), self.client_type_entry.get())
```

```
if count == 0:
    cursor.execute("INSERT INTO ClientiColete(ColetID, ClientID) VALUES (?, ?)",
self.colet_id, self.expeditor_id)
    cursor.commit()
    cursor.execute("INSERT INTO ClientiColete(ColetID, ClientID) VALUES (?, ?)",
self.colet_id, self.destinatar_id)
    cursor.commit()
```

```
# insert into table courier
cursor.execute("INSERT INTO Curieri(Nume, Prenume, CNP, Telefon, Salariu, Bonus)"
" VALUES (?, ?, ?, ?, ?, ?)",
self.last_name_courier_entry.get(), self.first_name_courier_entry.get(),
self.cnp_courier_entry.get(), self.phone_courier_entry.get(),
self.salary_courier_entry.get(), self.bonus_entry.get())
```

```
cursor.execute("INSERT INTO CurieriZone(CurierID, ZonaID) VALUES (?, ?)", curier_id, self.zona_id)
cursor.commit()
```

UPDATE — am realizat update pentru locatia la care se afla un colet la un moment dat si pentru salariul unui curier

Update Location

Choose a parcel by Name and Code

Name
Code
Location

UPDATE

Update Salary

Choose the courieri by name and CNP

First name
Last name
CNP
Salary

UPDATE

Astfel, a apasarea butonului UPDATE se executa:

- Pentru locatie Update:

```
cursor.execute("UPDATE Colet SET LocalizareID = ? WHERE Denumire = ? AND Cod = ?",  
              localizare_id, self.name_parsel_entry.get(), self.code_entry.get())  
cursor.commit()
```

- Pentru salariu Update:

```
cursor.execute("UPDATE Curieri SET Salariu = ? WHERE Nume = ? AND Prenume = ? AND CNP = ?",  
              self.salary_entry.get(), self.c_lname_entry.get(), self.c_fname_entry.get(),  
              self.c_cnp_entry.get())  
cursor.commit()
```

DELETE — am realizat delete pentru colete, putand alege 2 criterii de stergere: dupa data si dupa nume si cod si delete pentru curieri dupa nume si cnp.

Delete Courier	Delete parcel
<div>Delete a courier by Name and CNP</div> <div>First name <input type="text"/></div> <div>Last name <input type="text"/></div> <div>CNP <input type="text"/></div> <div>DELETE</div>	<div>Delete a row by Name and Code</div> <div>Name <input type="text"/></div> <div>Code <input type="text"/></div> <div>DELETE</div> <div>Delete by DATE</div> <div>Date <input type="text"/></div> <div>DELETE</div>

Atunci cand dormi sa stergem colete dupa data, tastam data ceruta si dupa apasarea butonului DELETE se vor sterge toate coletele ce au data livrarii am veche ca data introdusa de noi. Trebuie specificat ca informatia se sterge automat si din tabelul de legatura.

```
cursor.execute("SELECT ColetID FROM Colet WHERE DataLivrarii < ?", self.date_entry.get())
for row in cursor.fetchall():
    self.colet_id = row[0]
    cursor.execute("DELETE FROM ClientiColete WHERE ColetID = ?", self.colet_id)
    cursor.commit()

cursor.execute("DELETE FROM Colet WHERE DataLivrarii < ?", self.date_entry.get())
cursor.commit()
conectare.close()
```

Daca dormi sa stergem dupa nume si cod, introducem datele si aparam pe delete.

```
cursor.execute("SELECT ColetID FROM Colet WHERE Denumire = ? and Cod = ?", self.name_parsel_entry.get(),
               self.id_entry.get())
self.colet_id = cursor.fetchone()[0]

cursor.execute("DELETE FROM ClientiColete WHERE ColetID = ?", self.colet_id)

cursor.execute("DELETE FROM Colet WHERE Denumire = ? AND Cod = ?", self.name_parsel_entry.get(),
               self.id_entry.get())
cursor.commit()
```

Introducem datele despre curierul pe care dorim sa il stergem.

```
cursor.execute("DELETE FROM CurieriZone WHERE CurierID = ?", self.curier_id)
cursor.commit()

cursor.execute("DELETE FROM Colet WHERE CurierID = ?", self.curier_id)
cursor.commit()

cursor.execute("DELETE FROM Curieri WHERE CNP = ? AND Nume = ? AND Prenume = ?",
               int(self.c_cnp_entry.get()), self.c_lname_entry.get(), self.c_fname_entry.get())
cursor.commit()

conectare.close()
```

Pentru a putea realiza aceste lucruri, este nevoie de afisarea anumitor date.

De ex, daca dorim sa stergem un colet dupa nume, atunci trebuie sa avem o lista de colete. Daca dorim sa dam update la locatia unui colet, ne trebuie inca o lista de locatii. Astfel, am implementat cateva butoane care sa afiseze anumite informatii despre fiecare entitate in parte.

Couriers						
CourierID	Last Name	First Name	CNP	Phone	Salary	Bonus
1	Georgescu	Dan	1923456789012	0733123456	3000.00	500.00
2	Dumitru	Alina	2912345678901	0744123457	3200.00	300.00
3	Ionescu	Alexandru	3901234567890	0721123458	2800.00	400.00
4	Popa	Andreea	4902345678901	0752123459	3100.00	350.00
5	Marinescu	Gabriel	5903456789012	0733123460	3000.00	300.00
6	Vasilescu	Mihaela	6904567890123	0744123461	3300.00	450.00
7	Antonescu	Cristian	7905678901234	0715123462	2900.00	380.00
8	Stanciu	Ana	8906789012345	0756123463	3200.00	420.00
9	Gheorghe	Bianca	9907890123456	0712123464	3100.00	380.00
10	lordache	Radu	1108901234567	0745123465	3000.00	320.00

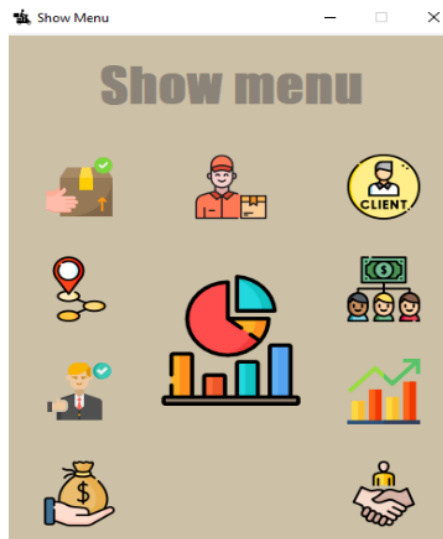
Locations				
LocationID	Code	Address	City	Country
1	1	Strada Victoriei nr.111	Timisoara	Romania
2	2	Strada Pacii nr.10	Brasov	Romania
3	3	Bulevardul Unirii nr.25	Cluj-Napoca	Romania
4	4	Calea Dorobantilor nr.12	Iasi	Romania
5	5	Aleea Rozelor nr.5	Braila	Romania
6	6	Strada Mihai Eminescu nr.1	Constanta	Romania
7	7	Bdul 1 Decembrie nr.8	Oradea	Romania
8	8	Calea Victoriei nr.20	Bucuresti	Romania

Customers				
ClientID	Last Name	First Name	CNP	Client type
1	Popescu	Ion	1980210123456	Expedito
2	Ionescu	Maria	2970321234567	Destinatar
3	Georgescu	Ana	3950412345678	Expedito
4	Radulescu	Doru	4960502345678	Destinatar
5	Mihai	Elena	5940603456789	Expedito
6	Dumitrescu	Marian	6930714567890	Destinatar
7	Stefanescu	Raluca	7920805678901	Expedito
8	Constantinescu	Adrian	8900916789012	Destinatar
9	Marin	Cristina	9911007890123	Expedito
10	Cristea	Marius	1091121012344	Destinatar

Parcels				
ColetID	Name	Code	DataLivrarii	LocalizareID
1	Pachet docu	1001	2023-01-16 12:00:00	1
2	Pachet stand	1002	2023-01-17 14:00:00	2
3	Pachet1	12345	2023-01-10 00:00:00	1
4	Pachet2	67390	2023-12-20 00:00:00	2
5	Pachet3	21436	2023-01-15 00:00:00	3
6	Pachet4	78901	2023-12-20 00:00:00	4
7	Pachet5	34567	2023-01-20 00:00:00	5
8	Pachet6	89012	2023-01-22 00:00:00	6
9	Pachet7	45678	2023-01-25 00:00:00	7
10	Pachet8	12956	2023-01-28 00:00:00	8

Areas		
ZonaID	Name	City
1	Zona A	Bucuresti
2	Zona B	Bucuresti
3	Zona C	Bucuresti
4	Zona A	Cluj-Napoca
5	Zona B	Cluj-Napoca
6	Zona C	Cluj-Napoca
7	Zona A	Constanta
8	Zona B	Constanta
9	Zona C	Constanta
10	Zona A	Ploiesti

SHOW MENU – INTEROGARI SIMPLE SI COMPLEXE



⇒ Apasarea acestui buton o sa deschida o fereastră in care se pot introduce informatii legate de comanda ce se doreste a fi afisata.



Astfel, pentru realizarea acestei functionalitati am implementat interogari simple ce vor afisa informatii despre comanda: coletul (Denumire si Cod), Curierul, Expeditorul si Destinatarul.

--1. Afisare Colet, Curier, Expeditor pentru un colet dat

```
SELECT Col.ColetID, Col.Denumire, Col.Cod, C.Nume + ' ' + C.Prenume  
AS Curier, Exped.Nume + ' ' + Exped.Prenume AS Expeditor  
FROM Colet Col INNER JOIN Curieri C ON Col.CurierID = C.CurierID  
INNER JOIN ClientiColete CC ON Col.ColetID = CC.ColetID  
INNER JOIN Clienti Exped ON CC.ClientID = Exped.ClientID  
AND Exped.TipClient = 'Expeditor'  
WHERE Col.Cod = 96822 AND Col.Denumire = 'Masa'
```

--2. Afisare Destinatari pentru un colet dat

```
SELECT Col.ColetID, Col.Denumire, Col.Cod, Dest.Nume + ' ' + Dest.Prenume  
AS Destinatar FROM Colet Col  
INNER JOIN ClientiColete CC ON Col.ColetID = CC.ColetID  
INNER JOIN Clienti Dest ON CC.ClientID = Dest.ClientID  
AND Dest.TipClient = 'Destinatar'  
WHERE Col.Cod = 96822 AND Col.Denumire = 'Masa'
```




Prin apasarea acestui buton, se va deschide o fereastră unde se vor putea introduce informatii despre o zona. Astfel, se vor afisa datele despre curierii care lucreaza in aria de acoperire a zonei respective.

Show couriers for a zone

Choose a zone

Location

Name



Zona Zona A din Localitatea Bucuresti

Name	CNP	Phone
Georgescu Dan	1923456789012	0733123456
Ionescu Alexandr	3901234567890	0721123458
Gheorghe Bianca	9907890123456	0712123464

Pentru implementarea acestei functionalitati am folosit interogarea simpla:

--3. Afisare curieri pentru fiecare zona

```
SELECT Z.ZonaID, Z.Denumire, Z.Localitate, C.Nume + ' ' + C.Prenume  
AS Curier, C.CNP, C.Telefon FROM Zone Z  
INNER JOIN CurieriZone CZ ON Z.ZonaID = CZ.ZonaID  
INNER JOIN Curieri C ON CZ.CurierID = C.CurierID  
WHERE Z.Denumire = 'Zona de jos' AND Z.Localitate = 'Onesti'
```





Prin apasarea acestui buton, se va deschide o fereasta unde se vor introduce informatii despre clientul pentru care dorim informatii despre toate coletele salte.

Show a customer's packages

Choose a client by name

First Name:

Last Name:

Clientul: Cristea Marius
Numar colete: 1

ID	Name	Code	Type	Status
0	Pachet10	23456	Colet primit	Nelivrat
1	Pachet4	78901	Colet primit	Nelivrat

Prin apasarea acestui buton, se va deschide o fereasta unde se vor introduce informatii despre clientul pentru care dorim informatii despre toate coletele salte. Pentru implementarea acestei functionalitati am folosit 2 interogari simple: 1 pentru colete livrate si una pentru nelivrate.

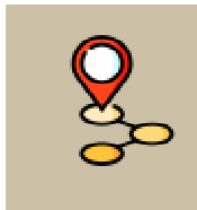
--4. Afisare colete livrate si nelivrate pentru un client

--NELIVRATE

```
SELECT Cl.TipClient, COUNT(Col.ColetID), Col.Denumire, Col.Cod FROM Clienti Cl
INNER JOIN ClientiColete CC ON Cl.ClientID = CC.ClientID
LEFT JOIN Colet Col ON CC.ColetID = Col.ColetID
WHERE Col.DataLivrarii > GETDATE() AND Cl.Nume = 'Badea' AND Cl.Prenume =
'Andrei' GROUP BY Cl.TipClient, Col.Denumire, Col.Cod ORDER BY Col.Denumire;
```

--LIVRATE

```
SELECT Cl.TipClient, COUNT(Col.ColetID), Col.Denumire, Col.Cod FROM Clienti Cl
INNER JOIN ClientiColete CC ON Cl.ClientID = CC.ClientID
LEFT JOIN Colet Col ON CC.ColetID = Col.ColetID
WHERE Col.DataLivrarii < GETDATE() AND Cl.Nume = 'Badea' AND Cl.Prenume =
'Andrei' GROUP BY Cl.TipClient, Col.Denumire, Col.Cod ORDER BY Col.Denumire;
```




Acest buton are ca functionalitate afisarea locatiei unui pachet. Se introduc numele si codul pachetului si se vor afisa locatia curenta, data curenta, data livrarii, data expedierii si curierul.

Show location for a parcel


Choose a parcel

Name

Code



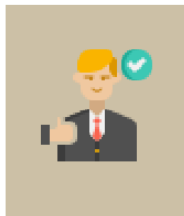
The parcel Pachet standard 1002 is located in Brasov
Current data: 2023-12-19
Data of dispatch: 2023-01-16
Data of delivery: 2023-01-17
Courier & phone: Dumitru Alina 0744123457



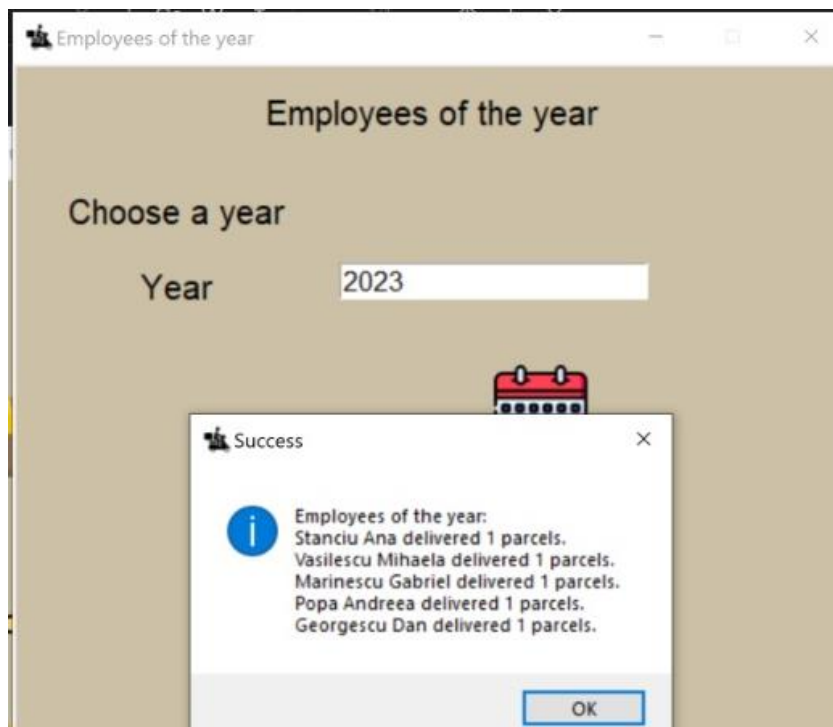
Pentru implementarea acestei functionalitati am folosit 1 interogare simpla:

--5. Afisare localizare colet, data expedierii, data livrarii si curierul (Nume + telefon)

```
SELECT L.Localitate, GETDATE(), Col.DataExpedierii, Col.DataLivrarii,  
C.Nume + ' ' + C.Prenume AS Curier, C.Telefon FROM Localizari L  
INNER JOIN Colet Col ON L.LocalizareID = Col.LocalizareID  
INNER JOIN Curieri C ON Col.CurierID = C.CurierID  
WHERE Col.Denumire = 'Laptop' AND Col.Cod = 11938
```



Acest buton deschide o fereastră unde se poate introduce un anumit an, apoi se afiseaza angajatul anului. De ex: daca introducem 2021, se vor afisa angajatii care au cele mai multe colete livrate din 2020 pana in 2021.



Pentru implementarea acestei functionalitati, am folosit interogarea complexa:

--1. Angajatul anului: Afisare angajat cu cele mai multe colete dintr-un anumit an

```
SELECT C.Nume + ' ' + C.Prenume AS Curier, COUNT(Col.ColetID) AS NumarColete
FROM Curieri C LEFT JOIN Colet Col ON C.CurierID = Col.CurierID
GROUP BY C.CurierID, C.Nume, C.Prenume
HAVING COUNT(Col.ColetID) = (SELECT TOP 1 COUNT(Col.ColetID) FROM Colet Col2
GROUP BY Col2.DataLivrarii
HAVING Col2.DataLivrarii BETWEEN DATEADD(Year,-1,GETDATE()) AND GETDATE())
ORDER BY COUNT(Col.ColetID) DESC)
```



Acest buton afiseaza anul in care s-au vandut cele mai multe colete si numarul acestora.

Success

×



The most productive year was 2021, we delivered 9 packages.

OK

Pentru implementare, am realizat interogarea complexa:

--2. Afisarea anului cu cele mai multe colete livrate

```
SELECT Year(Col.DataLivrarii) AS An, COUNT(*) AS NumarColete FROM Colet Col
WHERE Year(Col.DataLivrarii) = (SELECT TOP 1 Year(Col2.DataLivrarii) FROM Colet Col2
GROUP BY Year(Col2.DataLivrarii)
ORDER BY COUNT(*) DESC) GROUP BY Year(Col.DataLivrarii)
```



Acest buton afiseaza comenzile cele mai scumpe efectuate de firma de curierat.

Expensive orders

Most expensive orders

Name	Code	Cost
Pachet1	12345	50.00
Pachet3	21436	70.00
Pachet5	34567	60.00
Pachet6	89012	80.00
Pachet8	12956	90.00
Pachet9	67890	50.00

Pentru implementare am realizat:

--3. Afisare colete care costa mai mult decat media costurilor comenzilor

```
SELECT Col.Denumire, Col.Cod, Col.Cost FROM Colet Col
WHERE Col.Cost > ALL (SELECT AVG(C.Cost) FROM Colet C)
```



Acest buton deschide o pagina unde se poate introduce o locatie. Se vor afisa clientii/clientul cel mai loial al companiei, care a apelat de cele mai multe ori la serviciile acesteia. Criteriul de loialitate este: clientul sa fie din localitatea aleasa de administrator si sa aibe comenzi de peste 250 de lei. (se iau in considerare doar comenzile in care el a fost expeditor)

Loyal customers

Choose a location

Location

Error

No orders were placed during that period.

OK

Pentru implementare, am realizat interogarea:

--4. Afisare clienti fideli din Bucuresti/dintr-o localitate

```
SELECT Cl.ClientID, Cl.Nume, Cl.Prenume, Cl.CNP, C.Denumire, C.Cod, C.Cost FROM
Clienti Cl INNER JOIN ClientiColete CC ON Cl.ClientID = CC.ClientID
INNER JOIN Colet C ON CC.ColetID = C.ColetID
WHERE Cl.ClientID IN (SELECT Cl2.ClientID FROM Clienti Cl2 WHERE Cl2.Localitate =
'Bucuresti' AND Cl2.TipClient = 'Expeditor' AND Cl.ClientID = Cl2.ClientID)
GROUP BY Cl.ClientID, Cl.Nume, Cl.Prenume, Cl.CNP, CC.ColetID, C.Cost, C.Denumire,
C.Cod
HAVING C.Cost > 250;
```

Ce a contribuit fiecare?

In cadrul acestui proiect, am impartit in mod egal aproximativ toate cerintele, deoarece nu ne descurcam la un nivel ridicat niciunul , atat cu limbajul Python, cat si cu bazele de date. Insa, lucrând ca o echipa, am reusit sa ducem ideea pana la capat.

Ce dificultati ati intampinat?

Lucrul cu bazele de date a reprezentat cea mai grea piedica, deoarece nu am mai lucrat niciunul inca din liceu. Facand putin research si amintindu-ne cateva lucruri mai importante, am rezolvat problemele aparute.