

PROIECT BAZE DE DATE

Pentru baza de date a unui magazin online de produse cosmetice avem nevoie de urmatoarele tabele: Angajat, Departament, Client, Bon, Produs, Categorie, Comanda(tabela cu care vom rezolva o relatie M:M).

Angajat: AngajatID(PK),Nume,Prenume,DepartamentID(FK),Sex,CNP,Salariu

Departament: DepartamentID(PK),ManagerID(FK),NumeDepartament

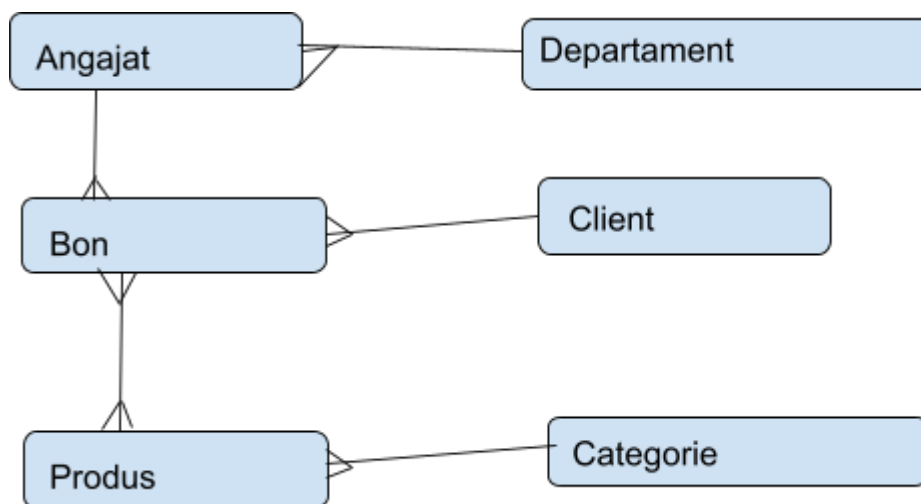
Client: ClientID(PK),Nume,Prenume,Strada,Numar,Oras,Judet,CNP,Sex

Bon: BonID(PK),ClientID(FK),AngajatID(FK),Data

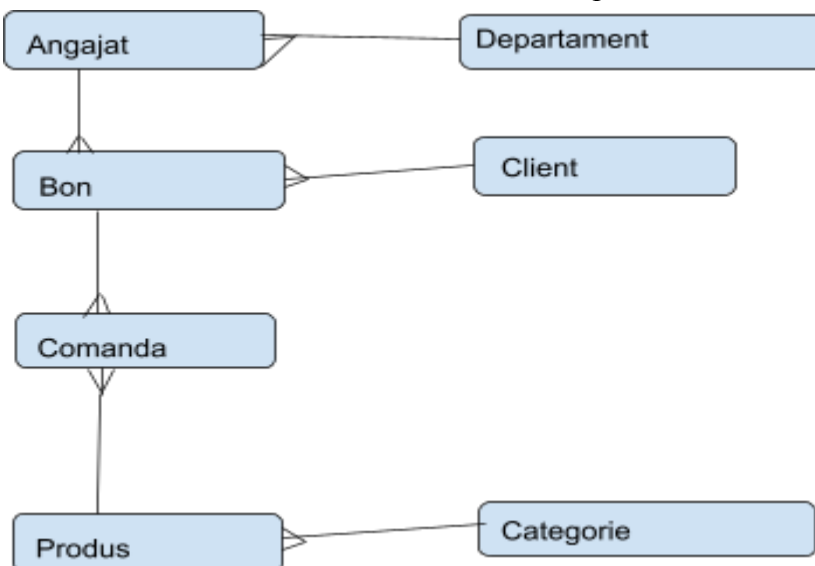
Produs: ProdusID(PK),CategorieID(FK),NumeProdus,Pret

Comanda: ComandaID(PK),BonID(PK),ProdusID(PK),Nr_bucati

Categorie: CategorieID(PK),NumeCategorie



Vom rezolva relatia M:M dintre Bon si Produs prin tabela Comanda.



Descrierea:

Un angajat apartine unui singur departament si astfel vom avea in tabela “Angajat” id-ul departamentului din care face parte precum si alte attribute.

Un departament are mai multi angajati, iar fiecare departament are un manager care este un angajat de-al firmei si astfel vom retine in tabela “Departament” id-ul diecarui manager pentru departamentul pe care il coordoneaza.

Un angajat va emite unul sau mai multe bonuri, iar un bon va fi emis de catre un singur angajat astfel ca vom avea relatie 1:M, iar in tabela “Bon” vom stoca pentru fiecare bon id-ul angajatului care l-a emis.

Bonul are pe el mai multe produse, iar un produs poate fi pe mai multe bonuri (mai multi clienti pot cumpara acelasi produs sau un client poate cumpara de mai multe ori acelasi produs) si astfel avem intre ele o relatie M:M. Aceasta relatie o vom rezolva adaugand tabela asociativa “Comanda” care va contine PK-urile de la celelalte 2 tabele si anume “BonID si ProdusID” precum si un PK “ComandaID”, iar pe langa acestea va mai fi un camp “Nr_bucati” pentru a sti pentru fiecare produs cate bucati sunt. Adaugand aceasta tabela am rezolvat relatia M:M, iar acum avem relatie 1:M intre Bon si Comanda si relatie 1:m intre Produs si Comanda.

Intre Produs si Categorie este o relatie de tip M:1, deoarece un produs poate apartine unei singure categorii, iar o categorie poate contine mai multe produse.

Constrangerile pe care le-am impus ar fi ca la toate tabelele ID-urile principale sunt indexate automat, iar in tabelele “Angajat” si “Client” sexul poate fi doar de forma ‘M’ sau ‘F’. De asemenea, CNP-urile sunt de tip UNIQUE KEY acolo unde apar.

Interfata grafica am facut-o in python, utilizand biblioteca pyodbc pentru a ma conecta la serverul sql.

```
server = r'DESKTOP-UM2DH23\SQLEXPRESS'
database = 'Online Cosmetics Shop'
conn = pyodbc.connect('DRIVER={SQL SERVER};SERVER=' + server + ';DATABASE=' + database + ';Trusted_Connection=yes')
```

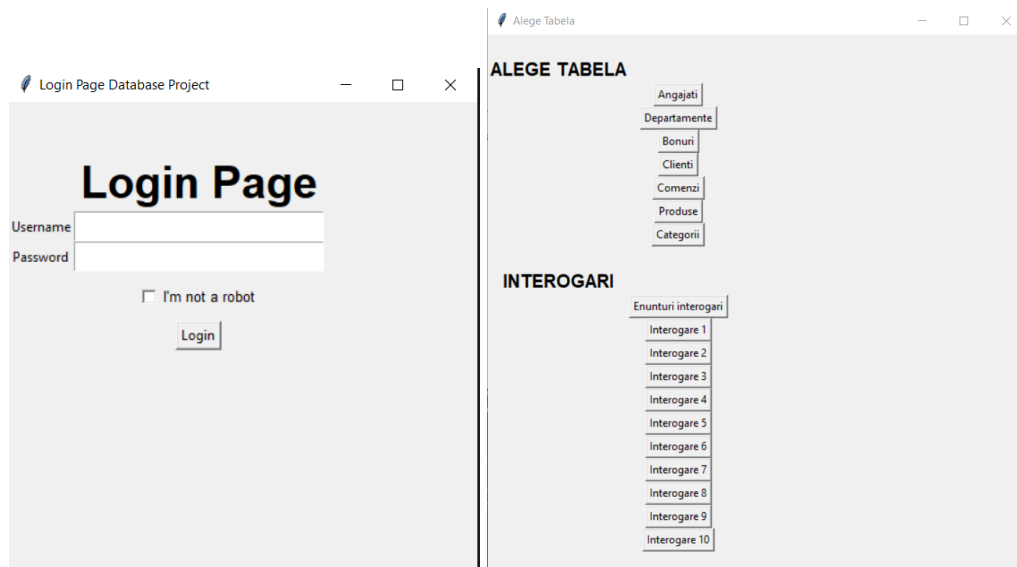
Am facut un formular de login cu utilizator si parola.

```
label=Label(root,text="\nLogin Page", font=('Arial', 30, 'bold'))
label.grid(row=0,column=1)

usernameLabel=Label(root, text="Username")
usernameLabel.grid(row=1,column=0)
username=Entry(root, font=('Arial',15))
username.grid(row=1,column=1)

passwordLabel=Label(root, text="Password")
passwordLabel.grid(row=2,column=0)
password=Entry(root, font=('Arial',15), show="*")
password.grid(row=2,column=1)

checkbox=Checkbutton(root, text="I'm not a robot",font=('Ariel', 10), variable=x1, onvalue=1, offvalue=0, padx=25, pady=10)
checkbox.grid(row=3,column=1)
loginbutton=Button(root, text="Login", command=clicklogin, state=ACTIVE)
loginbutton.grid(row=4, column=1)
```



Dupa autentificare se va putea alege ce tabela se doreste a fi modificata in partea de sus sau se pot observa enunturile si interogarile simple si cele complexe pe care le-am facut in partea de jos selectand-o pe care o dorim.

Tabelele pot fi modificate in functie de fiecare tabela, deoarece unele lucruri nu trebuie modificate sau nu pot fi modificate asa cum isi doreste fiecare.

Spre exemplu:

```
def Adaugaangajat():
    n=nume.get()
    p=prenume.get()
    d=int(departament.get())
    s=sex.get()
    c=str(cnp.get())
    s1 = ""
    for i in c:
        s1 = s1 + str(i)
    print(s1)
    print(type(s1))
    print(c)
    sal=int(salariu.get())

    server = r'DESKTOP-UM2DH23\SQLEXPRESS'
    database = 'Online Cosmetics Shop'
    conn = pyodbc.connect('DRIVER={SQL SERVER};SERVER=' + server + ';DATABASE=' + database + ';Trusted_Connection=yes')
    c=conn.cursor()
    values=[(n,p,d,s,s1,sal)]
    c.executemany("INSERT INTO Angajat(Nume,Prenume,DepartamentID,Sex,CNP,Salariu)VALUES(?,?,?,?,?,?)",values)
    conn.commit()
    c.close()
    conn.close()

    q1="select* from angajat"
    interogare(q1)
```

In codul de mai sus am implementat operatia de adaugare a unui angajat, introducand in interfata detaliile lui, dupa cum puteti observa mai jos.

Tabela Angajat

Nume		Prenume	
Departament		Sex	
CNP		Salariu	

[Adauga](#) [Sterge](#) [Modifica salariu](#) [Modifica departament](#)

ATENTIE! PENTRU STERGERE INTRODUCETI DOAR CNP
ATENTIE! PENTRU MODIFICARE SALARIU INTRODUCETI SALARIU SI CNP
ATENTIE! PENTRU MODIFICARE DEPARTAMENT INTRODUCETI ID-UL DEPARTAMENTULUI SI CNP

ID-uri DEPARTAMENTE:
IT-10
HR-20
MARKETING-30
VANZARI-40

Similar se poate efectua orice operatie se doreste pentru orice tabela, fiecare avand anumite limite.