UNIVERSITATEA TEHNICĂ “GH ASACHI” IAŞI

FACULTATEA AUTOMATICĂ si CALCULATOARE

SPECIALIZAREA CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI

Gestiunea productiei intr-o fabrica de parfumuri

Coordonator,

Prof. Mironeanu Cătălin

Student,

Agapie Oana

Grupa 1306A

Iași, 2023

Titlul proiectului

GESTIUNEA PRODUCTIEI INTR-O FABRICA DE PARFUMURI

Analiza, proiectarea și implementarea unei baze de date care să modeleze gestiunea productiei dintr-o fabrica de parfumuri si totodata o permiterea trimiterii unor comenzi catre fabrica, de catre utilizatori.

Descrierea proiectului

Informatiile de care avem nevoie sunt legate de: distribuitorii ce trimit cereri catre fabrica, cat si despre cererile acestora, iar in ceea ce priveste produsele vom pleca de la ingrediente pana la ambalaj pentru a contura produsul final.

In prima parte ne vom concentra pe distribuitorii ce trimit comenzi catre fabrica si pe cererile acestora. Cererile acestora vor contine informatiile esentiale de care un producator are nevoie pentru a indeplini preferintele unui distribuitor: tipul ambalajului, numarul de bucati .

In a doua parte ne vom ocupa de productia propriu zisa, evidenta stocurilor ingredientelor primare cat si a ambalajelor. Informatiile privind produsele au fost impartite pe 2 categorii ce prezinta totodata si etapele procerii unui produs final: etapa de creare a compozitiei parfumului si cea de ambalare.

Esentele parfumurilor vor fi alcatuite pe baza unor retete ce vor

contine informatii despre ingredientele ce compun parfumul: nume si

procentele in care acestea se afla in compozitia produsului.

Separat vom avea o sectiune cu informatii despre produse: gen/sex,

tipul parfumului si un link catre o poza cu produsul.

Tehnologii folosite pentru Front-end si Back-end

Pentru partea de Front-end

Pentru partea de Front-end am folosit limbajul Python, pachetul cx_oracle pentru conectarea la baza de date si PIL si tkinter pentru partea de grafica.

Pentru partea de Back-end

Am utilizat Oracle SQL Developer, Oracle SQL Developer Data Modeler pentru crearea tabelelor, realizarea modelului logic si modelului relațional.

Descrierea funcțională a aplicației

Principalele funcții ale aplicației sunt:

- Evidența ingredientelor
- Evidența ambalajelor
- Evidenta parfumurilor
- Evidența distribuitorilor
- Evidența cererilor

Structura și relațiile dintre tabele

Entitatile din această aplicație sunt:

- Distribuitor
- Cereri
- Ambalaje
- Esenta

- Ingrediente
- Info

În proiectarea acestei baze de date s-au identificat următoarele tipuri de relații:

1:1 (one-to-one), 1:n (one-to-many), m:n(many-to-many)

Între entitatea Distribuitor si entitatea Cereri se realizeaza o relatie 1:n. Un distribuitor poate da mai multe cereri, dar aceeasi cerere poate fi data doar de un singur distribuitor. Legatura dintre ele doua entitati se face prin campul Id_distribuitor.

Intre entitatea Cereri si entitatea Ambalaje se realizeaza o legatura 1:n. Un ambalaj poate fi continut de mai multe cereri, insa aceeasi cerere poate contine doar un tip de ambalaj. Legatura dintre cele doua entitati se face prin campul Id_ambalaj.

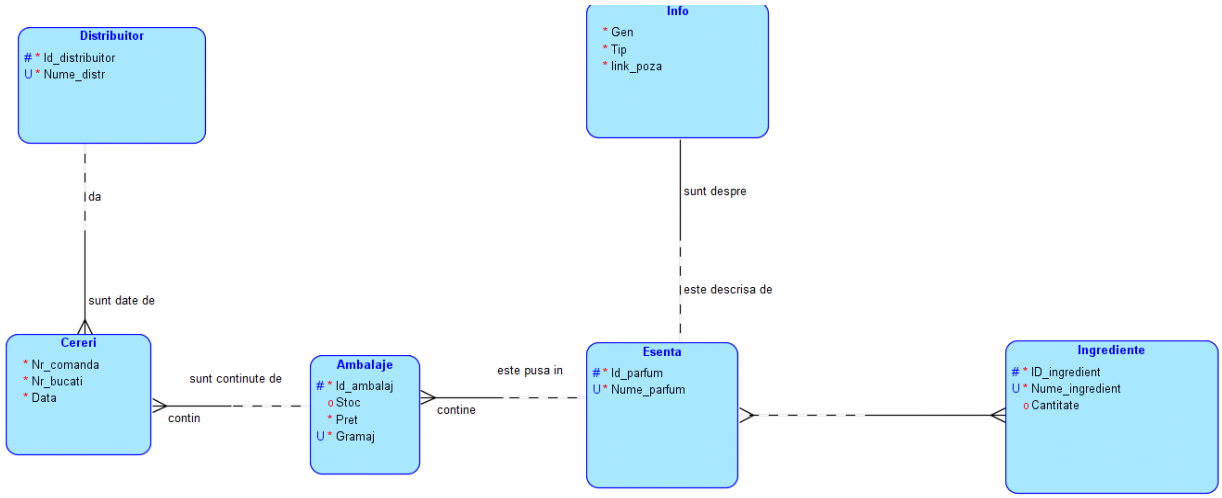
Intre entitatea Ambalaje si entitatea Esenta se realizeaza o legatura 1:n. O esenta poate contine mai multe ambalaje in functie de gramajul parfumului ce urmeaza a fi rezultat, insa un ambalaj poate contine o singura esenta. Legatura dintre cele doua entitati se face prin campul Id_parfum.

Intre entitatea Esenta si entitatea Ingrediente se realizeaza o legatura m:n. O esenta poate contine mai multe ingrediente si un ingredient poate fi continut de mai multe esente. Corespondenta dintre cele doua entitati se face pe baza unei alte entitati Formula ce se formeaza pe relatia m:n. Legatura dintre Esenta si Formula se face prin campul Id_parfum, iar intre Formula si Ingrediente prin campul

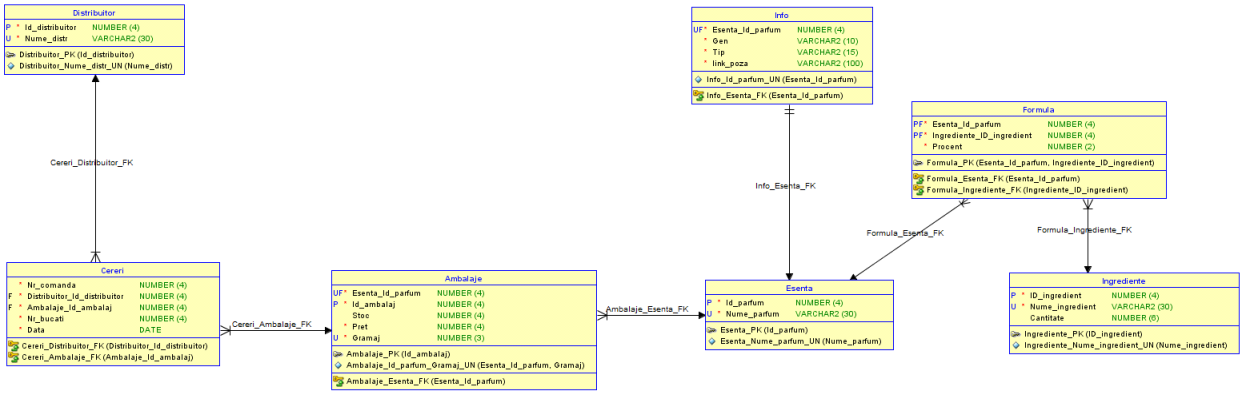
Id_ingredient.

Intre entitatea Info si entitatea Esenta se realizeaza o relatie 1:1. O esenta este descrisa de o singura serie de informatii, iar o serie de informatii pot descrie o singura esenta.Legatura dintre cele doua entitati se face prin campul Id_parfum.

Modelul logic



Modelul relațional



Descrierea constrângerilor

-Constrangerile primary key se gasesc aproape in toate entitatile: pentru

nume distribuitor, nume parfum si nume ingredient deoarece acestea nu pot fi gasite de mai multe ori la nivelul entitatilor distribuitor, esenta, respectiv ingrediente. Deasemenea aceleasi atribute contin si constrangeri de tip check pentru a contine doar litere, nu pot exista nume in care sa gasim cifre sau alte

caractere.

-Avem o contrangere unique la nivelul entitatii Ambalaje intre atributele

Id_parfum si gramaj, deoarece nu putem avea mai multe ambalaje pentru acelasi set de valori(Id_parfum,gramaj).

-Constrangerea de tip check mai este folosita si in entitatea Info, unde

atributul link ar trebui sa aiba sablonul unei adrese URL.

-Constrângerile de tip null se găsesc pe marea majoritate din atributele din tabele.

-Deasemenea am adaugat o constrangere unique atributului id_parfum, din entitatea Info deoarece nu putem avea mai multe seturi de informatii pentru acelasi parfum.

Descrierea modalității de conectare la baza de date din aplicație

Pentru conectarea la baza de date am utilizat biblioteca cx_oracle din Python.Conectarea a necesitat includerea unui folder cu diferite fișiere descărcat de la companie.

```
#conectare
user = 'bd001'
password = 'oana2106*'
cx_Oracle.init_oracle_client(
   lib_dir=r"C:\Users\Oana\Documents\an 3\tema_bd\instantclient-basic-windows.x64-21.7.0.0.0dbru\instantclient_21_7")
dsn_tns = cx_Oracle.makedsn('bd-dc.cs.tuiasi.ro', '1539', service_name='orcl')
conn = cx_Oracle.connect(user=user, password=password, dsn=dsn_tns)
cursor = conn.cursor()
```

Aspect interfată

Specificatii tehnice

Interfata grafica da posibilitatea clientului de a-si adauga produse in cos, de a trimite o comanda spre fabrica si deasemenea de a anula o comanda(daca aceasta a fost data de catre el).

In partea din stanga a interfetei avem o serie de campuri ce vor constitui comanda ce va fi trimisa la fabrica, iar in partea dreapta este afisat cosul consumatorului actualizat dupa fiecare adaugare a unui produs si totodata si totalul de plata ce trebuie achitat de client.

```
def adauga_in_cos():
    if(verificare_stocuri()==0):
        print("s-a adaugat in cos\n")
        cos.pret.append(client.pret)
        cos.coduri_parfum.append(client.cod_parfum)
        cos.cantitate.append(client.nr_bucati)
        cos.gramaj.append(client.gramaj)
        cos.nume_parfumuri.append(client.nume_parfum)
        total=0
        for i in range (0,len(cos.nume_parfumuri)):
            total=total+int(cos.cantitate[i])*int(cos.pret[i])
            inaltime=i/2+3.1
            Label(win, text=_str(cos.nume_parfumuri[i])+ "---->"+str(cos.cantitate[i])+"x"+ str(cos.pret[i]) + "   RON").place(relx=6.75/ 10,rely=inaltime / 10)
        cos.set_total(total)
        text =Label(win, text="TOTAL:").place(
        relx = 6.75 / 10, rely = 7.5 / 10)
        Label(win,
              text=str(cos.total)+"   RON").place(
            relx=_7.75/ 10, rely=7.5 / 10)
```

Pentru a da o comanda clientul trebuie sa-si completeze numele sau, numele parfumului, gramajul acestuia si numarul de bucati.
Numele parfumurilor vor fi extrase din tabela esenta:

```
#nume parfum
clicked = StringVar(win)
clicked.set("Select an perfume")
Label(win, text="Nume parfum").place(relx=1 / 10, rely=3.5 / 10)
cursor.execute('select nume_parfum from esenta')
row = cursor.fetchall()
nume_parfum = OptionMenu(win,clicked,*row,command=evaluare_nume_parfum)
nume_parfum.place(relx=2.5 / 10, rely=3.4 / 10)
```

Pentru numele selectat de client se vor extrage din tabela ambalaj doar gramajele disponibile pentru acel parfum:

```
    cursor.execute('select gramaj from ambalaje where esenta_id_parfum=\'' + str(client.cod_parfum) + '\'')
    rows = cursor.fetchall()
    vect=[]
    for i in rows:
        i = str(i)
        i=i[1:len(i)-2]
        vect.append(i)
    if not vect:
        vect.append("nu avem niciun produs disponibil")
    client.set_gramaje_disponibile(vect)
    create_gramaje()

def create_gramaje():
    clicked1.set("select a weight")
    gramaj = OptionMenu(win, clicked1, *client.gramaj_disponibil, command=evaluare_gramaj)
    gramaj.place(relx=2.5 / 10, rely=4.5 / 10)


def evaluare_gramaj(event):
    gramaj=clicked1.get()
    client.set_gramaj(gramaj)
    adauga_pret()
    client.gramaj_disponibil.clear()
```

Odata cu apasarea butonului de adaugare in cos se va face o verificare a stocurilor pentru a avea stocuri disponibile pentru realizarea cererii. Daca nu exista

cantitatea necesara pentru a produce parfumurile acestea nu vor fi adaugate in cos.

```
def verificare_stocuri():
        print(cos.stocuri_ambalaje)
        if client.cod_ambalaj in cos.coduri_ambalaje:
            print("se afla")
            index = cos.coduri_ambalaje.index(client.cod_ambalaj)
            print(index)
            print(cos.stocuri_ambalaje)
            if(int(client.nr_bucati)>int(cos.stocuri_ambalaje[index])):
                return 1
            cos.stocuri_ambalaje[index] =int(cos.stocuri_ambalaje[index]) - int(client.nr_bucati)
            ingrediente = []
            cursor.execute(
                'select ingrediente_id_ingredient from formula where esenta_id_parfum=' + str(client.cod_parfum) + '')
            rows = cursor.fetchall()
            for i in rows:
                i = str(i)
                i = i[1:len(i) - 2]
                i = int(i)
                ingrediente.append(i)

            cursor.execute(
                  'select procent from formula where esenta_id_parfum=' + str(client.cod_parfum) + '')
            rows = cursor.fetchall()
            procente = []
for i in rows:
    i = str(i)
    i = i[1:len(i) - 2]
    i=int(i)
    procente.append(i)
for i in range(0, len(ingrediente)):
    if ingrediente[i] in cos.id_ingredient:
        index = cos.id_ingredient.index(ingrediente[i])
        if float(client.nr_bucati) * float(client.gramaj) * float(float(procente[i])/100) > float(
                cos.stocuri_ingrediente[index]):
            return 1
        cos.stocuri_ingrediente[index] = float(cos.stocuri_ingrediente[index]) - float(client.nr_bucati) * float(
            client.gramaj) * float(float(procente[i])/100)
    else:
        cursor.execute(
            'select cantitate from ingrediente where id_ingredient=' + str(ingrediente[i]) + '')
        rows = cursor.fetchall()
        rows = str(rows)
        rows = rows[2:len(rows) - 3]
        cantitate = rows
        if float(client.nr_bucati)*float(client.gramaj)*float(float(procente[i])/100)>float(cantitate):
            return 1
        cos.stocuri_ingrediente.append(
            float(cantitate) - float(client.nr_bucati) * float(client.gramaj) * float(float(procente[i])/100))
        cos.id_ingredient.append(ingrediente[i])
```

```
else:
    print("produsul nu se afla deja in cos")
    cursor.execute(
        'select stoc from ambalaje where id_ambalaj=' + str(client.cod_ambalaj) + '')
    stoc_ambalaj_total = cursor.fetchall()
    stoc_ambalaj_total = str(stoc_ambalaj_total)
    stoc_ambalaj_total = stoc_ambalaj_total[2:len(stoc_ambalaj_total) - 3]
    if (int(client.nr_bucati) > int(stoc_ambalaj_total)):
        return 1
    cos.coduri_ambalaje.append(client.cod_ambalaj)
    cos.stocuri_ambalaje.append(int(stoc_ambalaj_total) - int(client.nr_bucati))

    ingrediente = []
    cursor.execute(
        'select ingrediente_id_ingredient from formula where esenta_id_parfum=' + str(client.cod_parfum) + '')
    rows = cursor.fetchall()
    for i in rows:
        i = str(i)
        i = i[1:len(i) - 2]
        i = float(i)
        ingrediente.append(i)

    cursor.execute(
        'select procent from formula where esenta_id_parfum=' + str(client.cod_parfum) + '')
    rows = cursor.fetchall()
    procente = []
```

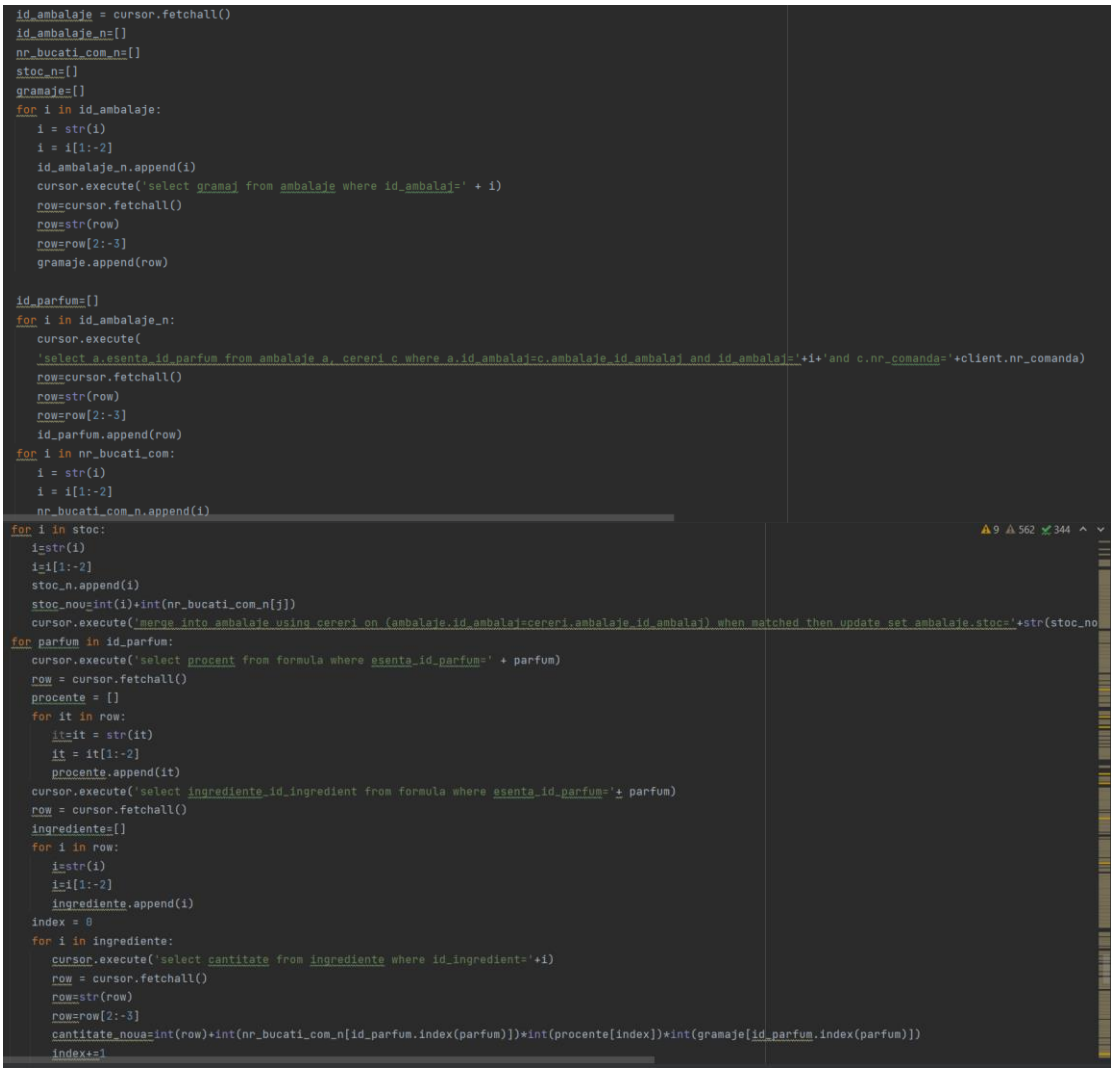
```
procente = []
for i in rows:
    i = str(i)
    i = i[1:len(i) - 2]
    procente.append(i)
for i in range (0,len(ingrediente)):
    if ingrediente[i] in cos.id_ingredient_:
        index = cos.id_ingredient.index(ingrediente[i])
        if float(client.nr_bucati)*float(client.gramaj)*float(float(procente[i])/100)>float(cos.stocuri_ingrediente[index]):
            return 1
        cos.stocuri_ingrediente[index]=float(cos.stocuri_ingrediente[index])-float(client.nr_bucati)*float(client.gramaj)*float(float(procente[i])/100)
    else:
        cursor.execute(
            'select cantitate from ingrediente where id_ingredient=' + str(ingrediente[i]) + '')
        rows = cursor.fetchall()
        rows = str(rows)
        rows = rows[2:len(rows) - 3]
        cantitate=rows
        if float(client.nr_bucati)*float(client.gramaj)*float(float(procente[i])/100)>float(cantitate):
            return 1
        cos.stocuri_ingrediente.append(float(cantitate)-float(client.nr_bucati)*float(client.gramaj)*float(float(procente[i])/100))
        cos.id_ingredient.append(ingrediente[i])
```

Butonul "Send the order" executa adaugarea comenzii in tabela cu comenzi din baza de date si actualizarea stocurilor(scaderea ambalajelor si ingredientelor necesare comenzii).

```
def send_order():
    cursor.execute(
        'select id_distribuitor from distribuitor where nume_distr=\'' + str(client.nume_distribuitor) + '\'')
    rows = cursor.fetchall()
    #daca exista numele il prelucram luam id ul
    if rows:
        rows = str(rows)
        rows = rows[2:len(rows) - 3]
    #daca nu exista numele il adaugam in distribuitori si luam id_ul lui
    else:
        query = 'insert into distribuitor (id_distribuitor,nume_distr) values (null,\''+str(client.nume_distribuitor) +'\')'
        cursor.execute(query)
        conn.commit()
        cursor.execute(
            'select id_distribuitor from distribuitor where nume_distr=\'' + str(client.nume_distribuitor) + '\'')
        rows = cursor.fetchall()
        rows = str(rows)
        rows = rows[2:len(rows) - 3]
    cursor.execute(
        'select max(nr_comanda) from cereri')
    nr_comanda = cursor.fetchall()
    nr_comanda=str(nr_comanda)
    nr_comanda=nr_comanda[2:len(nr_comanda)-3]
    for i in range (0,len(cos.coduri_ambalaje)):
        query = 'insert into cereri (nr_comanda,distribuitor_id_distribuitor,ambalaje_id_ambalaj,nr_bucati,data)values (' +nr_comanda+1+','+ str(
            rows) + ',' + cos.coduri_ambalaje[i] + ',' + cos.cantitate[i] + ',sysdate)'
        cursor.execute(query)
        query = 'update ambalaje set stoc=' + str(cos.stocuri_ambalaje[i]) + 'where id_ambalaj=' + str(
            cos.coduri_ambalaje[i]) + ''
        cursor.execute(query)
    for j in range(0, len(cos.stocuri_ingrediente)):
        query = 'update ingrediente set cantitate=' + str(
            cos.stocuri_ingrediente[j]) + 'where id_ingredient=' + str(
            cos.id_ingredient[j]) + ''
        cursor.execute(query)
    conn.commit()
```

Butonul "Cancel the order" poate fi utilizat pentru a anula comanda cu numarul completat in campul "Nr comanda" cu conditia ca utilizatorul ce doreste sa o anuleze sa fie cel care a trimis-o. In momentul indeplinirii acestei conditii se v-a sterge comanda sin tabela cu comenzi si se vor restabili stocurile conform adaugarii ingredientelor ce nu vor mai fi folosite.

```
def anulareComanda():
    query='select nr_comanda from cereri'
    cursor.execute(query)
    conn.commit()
    nr = cursor.fetchall()
    nr_comenzi=[]
    for i in nr:
        i=str(i)
        i=i[1:-2]
        nr_comenzi.append(i)
    if client.nr_comanda in nr_comenzi:
        cursor.execute('select nr_comanda from cereri c, distribuitor d where c.distribuitor_id_distribuitor=d.id_distribuitor and nume_distr=\''+client.nume_distribuitor+
        nr=cursor.fetchall()
        nr_n=[]
        for i in nr:
            i=str(i)
            i=i[1:-2]
            nr_n.append(i)
        if client.nr_comanda in nr_n:
            cursor.execute('select a.stoc from ambalaje a, cereri c where a.id_ambalaj=c.ambalaje_id_ambalaj and c.nr_comanda='+client.nr_comanda)
            conn.commit()
            stoc = cursor.fetchall()
            cursor.execute(
                'select c.nr_bucati from ambalaje a, cereri c where a.id_ambalaj=c.ambalaje_id_ambalaj and c.nr_comanda=' + client.nr_comanda)
            conn.commit()
            nr_bucati_com=cursor.fetchall()
            cursor.execute(
                'select c.ambalaje_id_ambalaj from ambalaje a, cereri c where a.id_ambalaj=c.ambalaje_id_ambalaj and c.nr_comanda=' + client.nr_comanda)
            conn.commit()
```



```
id_ambalaje = cursor.fetchall()
id_ambalaje_n=[]
nr_bucati_com_n=[]
stoc_n=[]
gramaje=[]
for i in id_ambalaje:
    i = str(i)
    i = i[1:-2]
    id_ambalaje_n.append(i)
    cursor.execute('select gramaj from ambalaje where id_ambalaj=' + i)
    row=cursor.fetchall()
    row=str(row)
    row=row[2:-3]
    gramaje.append(row)

id_parfum=[]
for i in id_ambalaje_n:
    cursor.execute(
    'select a.esenta_id_parfum from ambalaje a, cereri c where a.id_ambalaj=c.ambalaje_id_ambalaj and id_ambalaj='+i+'and c.nr_comanda='+client.nr_comanda)
    row=cursor.fetchall()
    row=str(row)
    row=row[2:-3]
    id_parfum.append(row)
for i in nr_bucati_com:
    i = str(i)
    i = i[1:-2]
    nr_bucati_com_n.append(i)
```

```
for i in stoc:                                                                                    9  562  344
    i=str(i)
    i=i[1:-2]
    stoc_n.append(i)
    stoc_nou=int(i)+int(nr_bucati_com_n[j])
    cursor.execute('merge into ambalaje using cereri on (ambalaje.id_ambalaj=cereri.ambalaje_id_ambalaj) when matched then update set ambalaje.stoc='+str(stoc_no
for parfum in id_parfum:
    cursor.execute('select procent from formula where esenta_id_parfum=' + parfum)
    row = cursor.fetchall()
    procente = []
    for it in row:
        it=it = str(it)
        it = it[1:-2]
        procente.append(it)
    cursor.execute('select ingrediente_id_ingredient from formula where esenta_id_parfum='+ parfum)
    row = cursor.fetchall()
    ingrediente=[]
    for i in row:
        i=str(i)
        i=i[1:-2]
        ingrediente.append(i)
    index = 0
    for i in ingrediente:
        cursor.execute('select cantitate from ingrediente where id_ingredient='+i)
        row = cursor.fetchall()
        row=str(row)
        row=row[2:-3]
        cantitate_noua=int(row)+int(nr_bucati_com_n[id_parfum.index(parfum)])*int(procente[index])*int(gramaje[id_parfum.index(parfum)])
        index+=1
```

Descrierea tranzactiei

Tranzatia a fost realizata in momentul in care vom da o comanda.