

# Proiect Baze de Date

Aplicație pentru gestiunea unei companii care oferă servicii în  
domeniul alimentar : Supermarket

## Cuprins

|  |    |
|--|----|
| 1.Descrierea cerintelor .....                      | 2  |
| 2.Schema relationala .....                         | 3  |
| 3.Constrangeri de integritate .....                | 5  |
| 4.Relatii tabele .....                             | 7  |
| 5.Functionalitate Aplicatie si explicare cod ..... | 8  |
| 6.Explicare interogari.....                        | 10 |

## 1.Descrierea Cerintelor

Supermarket-ul se ocupa cu comercializarea produselor alimentare.

Supermarket-ul are mai multi angajati. Pentru fiecare angajat se retin urmatoarele date: nume, prenume, CNP, data nasterii, adresa ( judet, oras, strada, numar ), sex, salariu.

Supermarket-ul isi structureaza sarcinile prin impartirea pe departamente. Pentru fiecare departament se retin numele si codul departamentului respectiv, descrierea ocupatiei.

Fiecare angajat apartine unui departament. Mai multi angajati pot apartine aceluiasi departament. Un angajat poate fi managerul unui singur departament. Fiecare departament este condus de un manager.

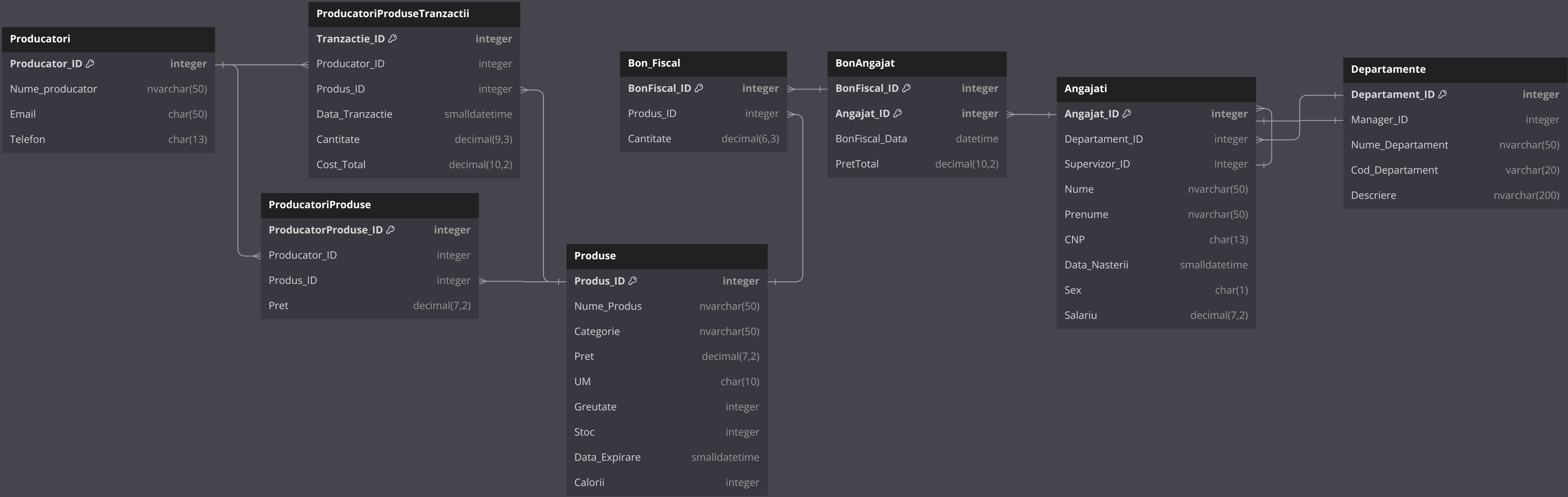
Fiecare angajat este coordonat de un supervizor. Fiecare supervizor poate coordona mai multi angajati. Supervizorul este tot un angajat.

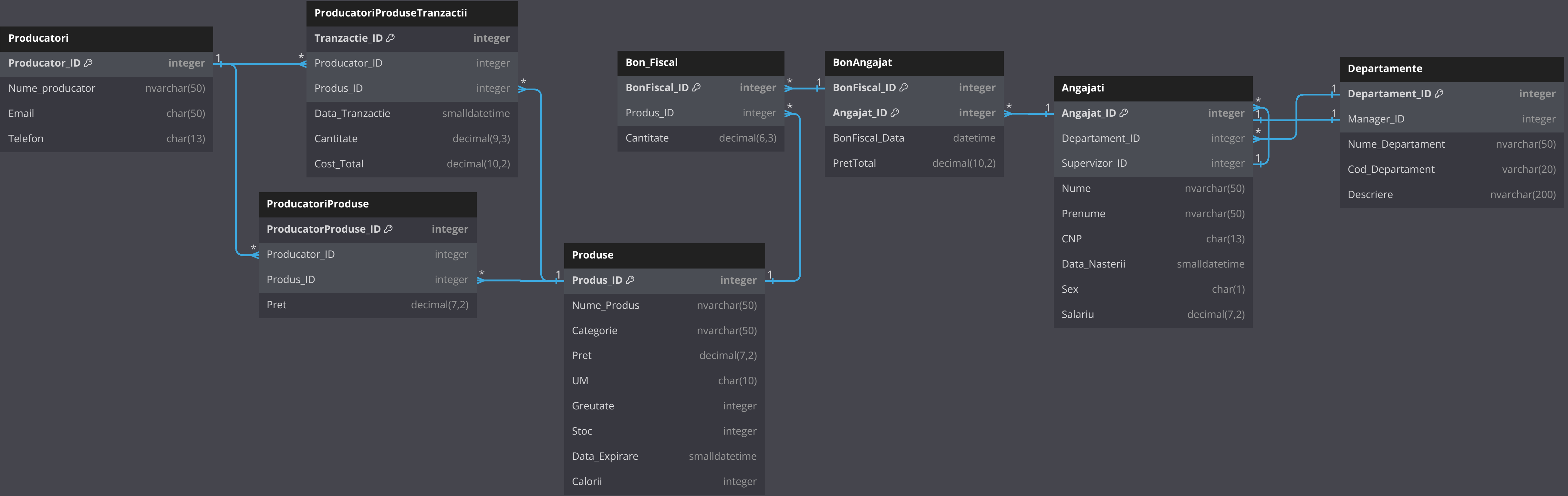
Produsele alimentare comercializate sunt descrise de urmatoarele caracteristici: numele produsului, categoria (fructe/legume/dulciuri/panificatie etc.. ), pretul per unitate, unitatea de masura(bucata/kg), greutatea per unitatea de masura (la bucata difera, iar la kg este 1), data de expirare si valorile nutritionale ( calorii, grasimi, glucide, proteine, sare ).

Produsele sunt comercializate de catre angajati. La vanzarea produselor angajatii elibereaza un bon fiscal. Pe bonul fiscal se noteaza data si pretul total al cumparaturilor efectuate.

Produsele sunt furnizate de catre producatori. Despre producatori se retin numele, email-ul si numarul de telefon. Supermarket-ul doreste sa tina in evidenta fiecare producator impreuna cu produsul si pretul per unitate al produsului pe care il furnizeaza.

Pentru a tine evidenta cheltuielilor/profiturilor se retin datele tranzactiei cu producatorii si anume: data la care a fost efectuata tranzactia, cantitatea achizitionata si costul total pentru tranzactia respectiva.





### 3. Constrangeri de integritate

#### Angajati

| Denumire       | Tip Date      | Not Null | Unique | Primary Key | Check                    | Default | Foreign Key |
|----------------|---------------|----------|--------|-------------|--------------------------|---------|-------------|
| Angajat_ID     | Int           | X        | X      | X           |                          |         |             |
| Departament_ID | Int           | X        |        |             |                          |         | X           |
| Supervizor_ID  | Int           |          |        |             |                          |         | X           |
| Nume           | nvarchar(50)  | X        |        |             |                          |         |             |
| Prenume        | nvarchar(50)  | X        |        |             |                          |         |             |
| CNP            | char(13)      |          | X      |             |                          |         |             |
| Data_Nasterii  | smalldatetime | X        |        |             |                          |         |             |
| Sex            | char(1)       | X        |        |             | ([Sex]='F' OR [Sex]='M') | 'F'     |             |
| Salariu        | decimal(7,2)  | X        |        |             |                          |         |             |

#### BonFiscal

| Denumire     | Tip Date | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|--------------|----------|----------|--------|-------------|-------|---------|-------------|
| BonFiscal_ID | Int      | X        | X      | X           |       |         | X           |
| Produs_ID    | Int      | X        |        | X           |       |         | X           |
| Cantitate    | Int      | X        |        |             |       |         |             |

#### BonAngajat

| Denumire       | Tip Date      | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|----------------|---------------|----------|--------|-------------|-------|---------|-------------|
| BonFiscal_ID   | Int           | X        | X      | X           |       |         |             |
| Angajat_ID     | Int           | X        |        |             |       |         | x           |
| BonFiscal_Data | smalldatetime | X        |        |             |       |         |             |
| PretTotal      | decimal       |          |        |             |       |         |             |

#### Departamente

| Denumire         | Tip Date      | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|------------------|---------------|----------|--------|-------------|-------|---------|-------------|
| Departament_ID   | int           | X        | X      | X           |       |         |             |
| Manager_ID       | Int           | X        | X      |             |       |         | X           |
| Nume_Departament | nvarchar(50)  |          |        |             |       |         |             |
| Cod_Departament  | varchar(20)   | X        | X      |             |       |         |             |
| Descriere        | nvarchar(200) |          |        |             |       |         |             |

#### Producatori

| Denumire        | Tip Date     | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|-----------------|--------------|----------|--------|-------------|-------|---------|-------------|
| Prodicator_ID   | Int          | X        | X      | X           |       |         |             |
| Nume_Prodicator | nvarchar(50) | X        | X      |             |       |         |             |
| Email           | char(50)     | X        | X      |             |       |         |             |
| Telefon         | char(13)     | X        | X      |             |       |         |             |

## ProducatorProduse

| Denumire             | Tip Date     | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|----------------------|--------------|----------|--------|-------------|-------|---------|-------------|
| ProducatorProduse_ID | Int          | X        | X      | X           |       |         |             |
| Produs_ID            | Int          | X        |        |             |       |         | X           |
| Pret                 | decimal(7,2) | X        |        |             |       |         |             |

## ProducatoriProduseTranzactii

| Denumire        | Tip Date      | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|-----------------|---------------|----------|--------|-------------|-------|---------|-------------|
| Tranzactie_ID   | Int           | X        | X      | X           |       |         |             |
| Produs_ID       | Int           | X        |        |             |       |         | X           |
| Data_Tranzactie | Smalldatetime | X        |        |             |       |         |             |
| Cantitate       | Int           | X        |        |             |       |         |             |
| Cost_Total      | decimal(10,2) |          |        |             |       |         |             |

## Produse

| Denumire      | Tip Date      | Not Null | Unique | Primary Key | Check | Default | Foreign Key |
|---------------|---------------|----------|--------|-------------|-------|---------|-------------|
| Produs_ID     | int           | X        | X      | X           |       |         |             |
| Nume_Produs   | nvarchar(50)  |          |        |             |       |         |             |
| Categorie     | nvarchar(50)  |          |        |             |       |         |             |
| Pret          | decimal(7, 2) |          |        |             |       |         |             |
| UM            | char(10)      |          |        |             |       |         |             |
| Greutate      | int           |          |        |             |       |         |             |
| Stoc          | int           |          |        |             |       |         |             |
| Data_Expirare | smalldatetime |          |        |             |       |         |             |
| Calorii       | int           |          |        |             |       |         |             |

## 7. RELATII TABELE

|                              | Producatori | ProducatoriProduse | ProdProduseTranzactii | Produse     | Bon_Fiscal | BonAngajat | Angajati       | Departamente    |
|------------------------------|-------------|--------------------|-----------------------|-------------|------------|------------|----------------|-----------------|
| <b>Producatori</b>           | -           | <b>1:N</b>         | <b>1:N</b>            | <b>N:N*</b> | -          | -          | -              | -               |
| <b>ProducatoriProduse</b>    | <b>1:N</b>  | -                  | -                     | <b>1:N</b>  | -          | -          | -              | -               |
| <b>ProdProduseTranzactii</b> | <b>1:N</b>  | -                  | -                     | <b>1:N</b>  | -          | -          | -              | -               |
| <b>Produse</b>               | <b>N:N</b>  | <b>1:N</b>         | <b>1:N</b>            | -           | <b>1:N</b> | <b>N:N</b> | -              | -               |
| <b>Bon_Fiscal</b>            | -           | -                  | -                     | <b>1:N</b>  | -          | <b>1:N</b> | -              | -               |
| <b>BonAngajat</b>            | -           | -                  | -                     | <b>N:N</b>  | <b>1:N</b> | -          | <b>1:N</b>     | -               |
| <b>Angajati</b>              | -           | -                  | -                     | -           | -          | <b>1:N</b> | <b>1:N</b>     | <b>1:N, 1:1</b> |
| <b>Departamente</b>          | -           | -                  | -                     | -           | -          | -          | <b>1:N,1:1</b> | -               |

\* Intre producatori si produse exista doua relatii de tipul N:N reprezentate prin doua tabele de legatura: ProducatoriProduse, ProdProduseTranzactii

## 5. Functionalitate Aplicatie si explicare cod

Aplicatia este realizata in python cu ajutorul bibliotecilor: tkinter, functools, copy, tkcalendar, pandas, pypyodbc, datetime, copy, os.

Aplicatia a fost gandita initial ca una generala de vizualizare a tuturor tabelelor dintr-o anumita baza de date cu optiunea de a adauga, vizualiza si modifica inregistrari in si din tabele presupunand ca fiecare tabel are o singura cheie primara si ca aceasta va fi pe prima pozitie. Din acest motiv codul python este mai lung decat ar fi fost necesar, initial aplicatia fiind gandita pentru uz general, apoi avand sa aduc modificarile specifice temei ( hardcodarea butoanelor/interogarilor/legaturilor specifice intre tabele, respectiv a modalitatii de afisare a acestora ).

### LOGIN:

Initial am testat aplicatia separat de mediul “SQL Server 2014 Management Studio” pe o baza de date template de pe internet. Din acest motiv am decis implementarea unui sistem simplu de login/register separat de cel din SQL Server. Acest login este deschis pe parcursul aplicatiei permitand deschiderea mai multor instante ( am gandit initial pentru a avea posibilitatea de logare si depanare a utilizatorilor cu drepturi de acces diferite : TODO ).

Aplicatia se lanseaza din functia setupmain() care deseneaza fereastra de Account login, se apeleaza functia main\_account\_screen() pentru a desena butoanele de login/register. De aici se apeleaza ulterior in functie de decizia utilizatorului functiile login() si register() care permit utilizatorului sa se inregistreze si/sau logheze in cont ( nu am implementat o criptare a numelui de utilizator / a parolei ). In cazul in care utilizatorul se conecteaza la un cont existent este apelata functia View():

### IMPLEMENTARE PROPRIU ZISA:

Functia View() este functia principala a aplicatiei ce are rolul de a afisa propriu zis datele rezultate din interogari, in aceasta se populeaza un tabel cu rezultatele primite in urma interogarilor, se seteaza butoanele de selectie pentru tabele, respectiv butoanele pentru a adauga si executa interogari in cadrul aplicatiei. Prin intermediul bibliotecii pypyodbc am putut face rost de numele tabelelor fara a fi nevoie initial sa hardcodez nimic, astfel avand un numar variabil de butoane de selectie pentru fiecare tabel.



Odata apelata functia View() si populat tabelul cu date din baza de date, prin selectia unui rand din tabelul anterior completat se apeleaza functia item\_selected() care face rost de datele din acel rand selectat ( exista optiunea de a copia in clipboard acele date ) si se apeleaza functia modify\_values() care ne deschide o noua fereastră cu optiunea de a modifica sau sterge randul pe care l-am selectat anterior.

Atat modify\_values() cat si View() trateaza problema informatiilor care nu sunt necesare pentru un utilizator: id-urile folosite pentru a face legatura intre tabele. Acestea au fost inlocuite cu informatia pe care o reprezinta, informatie stocata intr-o matrice de mapare declarata la inceputul programului ce contine tabelul catre care face legatura o anumita informatie, cu ce trebuie inlocuita si care ar fi textul formatat ce va trebui aratat utilizatorului. In cazul in care se alege modificarea / stergerea unui rand din tabel se apeleaza functia update\_query() la apasarea celor doua butoane in care se distinge actiunea ceruta, se construiesc interogarea care va executa actiunea propriu zisa si este aplicata asupra bazei de date.

In cazul in care se apasa butonul ADD din fereastră corespunzatoare functiei View() se apeleaza tot functia update\_query(). In acest caz se deschide o fereastră diferita fata de cea corespunzatoare pentru update/delete in care va trebui sa verificam din nou ce date dorim sa afisam in aceasta. La apasarea butonului ADD ROW se apeleaza functia addelement care construiesc interogarea pentru a adauga elementul cu datele respective in tabel si o executa.

In cazul in care se selecteaza 10Interogari/Query, se va apela functia action4choose() cu un parametru in functie de alegerea facuta. Alegerea Query este separata de cerintele temei si mi-a permis sa testez functionalitatea functiei view prin executarea de interogari scrise. Alegerea 10Interogari trateaza cerinta de a avea 6 interogari cu JOIN si 4 interogari cu subcereri. Aceasta alegere deschide o fereastră noua in care sunt listate enunturile prin intermediul unui dropdown. Aceste enunturi sunt corelate cu interogarile stocate in interogariTema. Pentru a parcurge lista de interogari sunt doua optiuni: selectarea cerintei dorite din dropdown list sau parcurgerea lor folosind butoanele next/previous. Pentru ambele optiuni se citește indexul la care se afla cerinta respectiva, se afiseaza pe ecran impreuna cu campul variabil ( sau campurile ) corespunzatoare interogarii respective. La apasarea butonului Execute se inlocuiesc label-urile “placeholder”, respectiv “place2Holder” cu valorile selectate din fereastră si se executa interogarea corespunzatoare. Dupa executarea interogarii se reinitializeaza lista de interogari cu copia acesteia pentru a putea inlocui din nou in cazul altei interogari.

## 6. Explicare interogari

1. Numele produselor vandute cu o cantitate mai mare decat cea [introdusa]

```
SELECT nume_produs AS "Nume Produs" FROM Produse P
INNER Bon_fiscal B
ON P.produs_id=B.produs_id
WHERE B.cantitate > placeHolder
```

2. Numele producatorilor ce aprovizioneaza cu un [anumit produs]

```
Select distinct nume_producator as "Nume Producator" from Producatori P
inner join ProducatoriProduse PP
on P.producator_id=PP.producator_id
inner join Produse Pr
on PP.produs_id=Pr.produs_id
where Pr.nume_produs like '%placeholder%'
```

3. Numarul de bonuri pe care apare un [anumit produs] vandut dupa o [anumita data]

```
select distinct count(*) as "Numar Bonuri" from Bon_Fiscal BF
inner join BonAngajat BA
on BF.bonfiscal_id=BA.bonfiscal_id
inner join Produse Pr
on BF.produs_id=Pr.produs_id
where Pr.nume_produs like '%placeholder%' and Ba.bonfiscal_data >
'place2Holder'
group by Bf.bonfiscal_id
```

4. Numele angajatilor care au vandut un [anumit produs]

```
select distinct concat(A.nume, ' ', A.prenume) as "Nume Angajati" from
Angajati A
inner join BonAngajat BA
on A.angajat_id=BA.angajat_id
inner join Bon_Fiscal BF
on BA.bonfiscal_id=BF.bonfiscal_id
inner join Produse P
on BF.produs_id=P.produs_id
where P.nume_produs like '%placeholder%'
```

5. Pretul platit de supermarket pentru a achizitiona un [anumit produs]

```
select sum(PPT.cost_total) as "Cost Total" from
ProducatoriProduseTranzactii PPT
inner join Produse P
on P.produs_id=PPT.produs_id
where P.ume_produs like '%placeholder%'
```

6. Supervizorii care au mai multi subordonati decat o [anumita valoare]

```
SELECT concat(A.Nume, ' ', A.Prenume) as "Nume Supervizori"
FROM Angajati A INNER JOIN Angajati AA ON
A.Angajat_ID=AA.Supervizor_ID
GROUP BY A.Nume, A.Prenume
HAVING COUNT(A.Angajat_ID) >= placeholder
```

7. Angajatii cu cel mai mare salariu pentru fiecare departament, sortati descrescator

```
SELECT A.Nume, A.Prenume, A.Salariu, D.ume_departament as "Nume
Departament"
FROM Angajati As A
INNER JOIN
( SELECT MAX(AA.Salariu) as SalariuMaxim, DD.Department_ID,
DD.ume_Departament
FROM Angajati as AA
INNER JOIN Departamente As DD on AA.Department_ID =
DD.Department_ID
GROUP BY DD.Department_ID, DD.ume_Departament
) As D on A.Department_ID = D.Department_ID and D.SalariuMaxim =
A.Salariu
ORDER BY A.Salariu DESC
```

8. Cei mai tineri angajati din fiecare departament. Ordonati dupa data nasterii.

```
SELECT A.Nume, A.Prenume, B.ume_Departament as "Nume
Departament", A.data_nasterii as "Data Nasterii"
FROM Angajati A
INNER JOIN (SELECT Department_ID, MAX(data_nasterii) AS
data_nasterii
FROM Angajati
GROUP BY Department_ID) C
ON A.Department_ID = C.Department_ID AND
A.data_nasterii=C.data_nasterii
INNER JOIN Departamente B ON A.Department_ID = B.Department_ID
```

9. Departamentele care au mai putini angajati decat un [anumit numar]

```
SELECT D.ume_departament "Nume departament"  
FROM Departamente D  
WHERE (SELECT count(*) FROM angajati a WHERE  
a.Department_ID=D. Departament_ID) < placeHolder
```

10. Primii [n] angajati in functie de salariul castigat

```
SELECT TOP placeHolder concat(A.ume, ' ', A.prenume) as "Angajat",  
D.ume_departament as "Departament", A.salariu  
FROM angajati A  
INNER JOIN Departamente D  
ON A.departament_id=D.departament_id  
ORDER BY salariu DESC
```

11. In ce an s-au nascut cei mai multi angajati? Afisati anul si numarul angajatilor.

```
SELECT year(data_nasterii) as "Anul Nasterii", count(*) as "Nr Angajati"  
FROM angajati  
GROUP BY year(data_nasterii)  
HAVING count (*) IN (SELECT TOP 1 count(*)  
FROM angajati  
GROUP BY year(data_nasterii)  
ORDER BY 1 DESC)
```

12. Angajatii care au un salariu mai mare decat media salariilor, sortati crescator

```
SELECT concat(Nume, ' ', Prenume), salariu FROM Angajati  
WHERE SALARIU > (  
SELECT AVG(Salariu) FROM Angajati)  
ORDER BY salariu asc
```

13. Produsele care depasesc media caloriilor din categoria lor.

```
SELECT P.ume_produs, P.categorie, P.calorii FROM Produse P  
WHERE P.calorii > (  
SELECT AVG(Q.calorii)  
FROM Produse Q  
where Q.categorie = P.categorie)
```

14. Producatorii celor mai scumpe produse din fiecare categorie.

```
Select Pr.ume_producator as "Producator", P.ume_produ as "Produ",  
P.categorie, P.pret from Producatori Pr  
inner join ProducatoriProduce PP  
on Pr.producator_id=PP.producator_id  
inner join Produce P  
on PP.produ_id=P.produ_id  
inner join (  
SELECT categorie, MAX(pret) as pret_maxim FROM produse  
GROUP BY categorie ) AS preturi_maxime  
ON P.categorie = preturi_maxime.categorie and P.pret =  
preturi_maxime.pret_maxim
```

15. Profitul realizat pentru fiecare produs (vanzare - aprovizionare)

```
Select PQ.ume_produ as "Produ", -PPT.cost_total + (Select  
sum(BF.cantitate)*P.pret FROM bon_fiscal BF  
inner join Produce P  
on P.produ_id = Bf.produ_id  
where P.produ_id=PPT.produ_id  
GROUP BY Bf.produ_id, P.pret) as "Profit Actual" FROM  
ProducatoriProduceTranzactii PPT  
inner join Produce PQ  
on PPT.produ_id = PQ.produ_id
```

16. Angajatii cu salariul mai mare decat salariul maxim dintr-un [anumit departament].

```
SELECT concat(A.ume, ' ', A.prenume) as "Angajat", D.ume_departament  
as "Nume departament", A.salariu  
FROM angajati A  
inner join Departamente D  
on A.departament_id=D.departament_id  
WHERE salariu > ALL  
(SELECT AA.salariu  
FROM angajati AA  
inner JOIN Departamente DD  
ON AA.departament_id = DD.departament_id  
WHERE DD.ume_departament like '%placeholder%')
```