

UNIVERSITATEA DIN BUCUREȘTI  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT**  
**BAZE DE DATE**

PROFESOR COORDONATOR:

VASILE SILVIU LAURENȚIU

STUDENT:

SCHMIDT ROBERT-EDUARD

BUCUREȘTI

2023

UNIVERSITATEA DIN BUCUREȘTI  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

## **LANȚ DE MAGAZINE**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU LAURENȚIU

STUDENT:

SCHMIDT ROBERT-EDUARD

BUCUREȘTI  
2023

Prezentarea modelului din viața reală: .....	4
Reguli.....	4
Franciza.....	4
Magazin.....	4
Raion .....	4
Produs .....	4
Oferta .....	4
Istoric_Oferta .....	4
Angajat.....	4
Post.....	4
Contract.....	4
Diagrama entitate-relație.....	5
Descrierea entităților, atributelor și cheilor .....	5
Franciza.....	5
Magazin.....	5
Raion .....	5
Produs .....	6
Produs_Raion.....	6
Oferta .....	6
Istoric_Oferte .....	6
Angajat.....	6
Post.....	6
Contract.....	7
Descrierea relațiilor și a cardinalităților.....	7
Franciza conține magazine - relație one-to-many .....	7
Magazinul conține raioane - relație one-to-many .....	7
Raioanele conțin produse - relație many-to-many .....	7
Produsele au oferte - relație many-to-many .....	7
Angajatul lucrează ca un post la un magazin - relație de tip 3 .....	7
Diagrama conceptuală.....	8
Constrângeri de integritate.....	8
Scheme relaționale: .....	9
Crearea tabelelor .....	10
Tabelul franciza .....	10
Tabelul magazin.....	10
Tabelul angajat.....	10
Tabelul post.....	10
Tabelul contract .....	11
Tabelul raion .....	11
Tabelul oferta .....	11
Tabelul produs .....	12

Tabelul produs_raion .....	12
Tabelul istoric_oferte .....	12
Inserarea datelor în baza de date .....	13

## **Prezentarea modelului din viața reală:**

### **Reguli**

#### **Franciza**

- Franciza poate avea 0 sau mai multe magazine.

#### **Magazin**

- Un magazin poate avea 0 sau mai multe raioane și 0 sau mai multe contracte.
- Un magazin trebuie să aparțină de o franciză.
- Un magazin trebuie să fie prezent în contractul angajatului.

#### **Raion**

- Un raion poate avea 0 sau mai mult produse.
- Un raion trebuie să aparțină de un magazin.

#### **Produs**

- Un produs poate aparține în 0 sau mai multe raioane.
- Un produs poate avea 0 sau mai multe oferte aplicată.
- Prețul trebuie să fie pozitiv.

#### **Oferta**

- O ofertă se poate aplica la 0 sau mai multe produse.
- Data de final trebuie să fie mai mare decât data de început.
- Procentajul ofertei trebuie să fie un număr între 0 și 1.

#### **Istoric\_Oferta**

- Ofertele aplicate produselor sunt păstrate în acest tabel.

#### **Angajat**

- Un angajat poate să aibă unul sau mai multe contracte în care să fie și un post și un magazin.

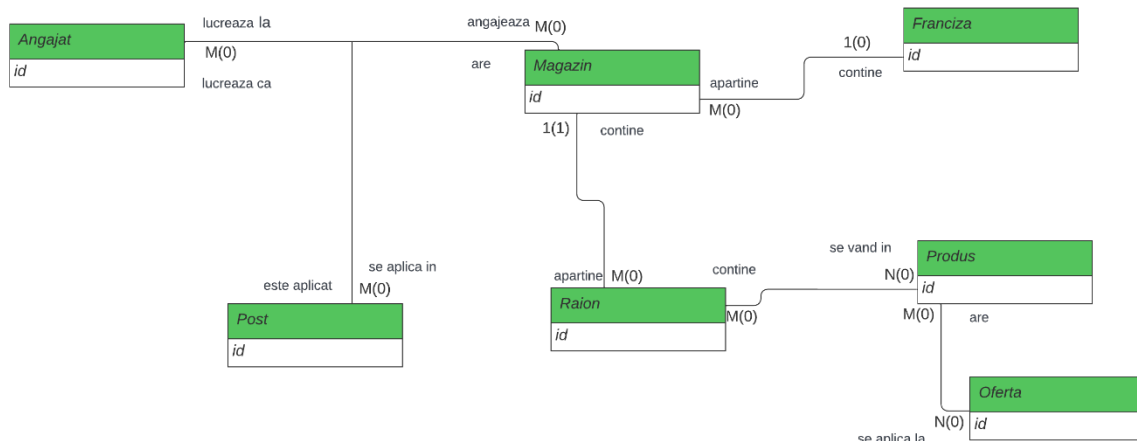
#### **Post**

- Un post apare în contractele angajaților.
- Salariul trebuie să fie mai mare decât 0.

#### **Contract**

- Un contract se face între un magazin și un angajat, cu un post.
- Un contract are o data de început și o dată de final, care reprezintă perioada de muncă a angajatului, iar data de final trebuie să fie mai mare decât cea de început.
- Un contract trebuie să aibă un angajat.

## Diagrama entitate-relație



## Descrierea entităților, atributelor și cheilor

### Franciza

**Descriere** - diferitele francize operate de investitori

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea francizei
- locație - tip de dată varchar(100), locația francizei (țară, regiune, oraș, după caz)
- numeDetinator - tip de dată varchar(30), numele investitorului care deține franciza

### Magazin

**Descriere** - magazinele care operează sub francize

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea magazinului
- adresă - tip de dată varchar(100), adresa la care se află magazinul
- programStart - tip de dată varchar(10), ora la care se deschide magazinul
- programFinal - tip de dată varchar(10), ora la care se închide magazinul
- dataDeschiderii - tip de dată date, data exactă în care s-a deschis pentru prima dată magazinul (data în care este adăugat în baza de date)
- idFranciza - tip de dată int, **cheie străină** către coloana id din tabelul Franciza, franciza din care face parte magazinul

### Raion

**Descriere** - raioanele care se află în magazine

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea raionului
- nume - tip de dată varchar(20), numele raionului
- tipRaion - tip de dată varchar(20), tipul raionului (congelate, perisabile, etc.)
- idMagazin - tip de dată int, **cheie străină** către coloana id din tabelul Magazin, magazinul din care face parte raionul

## Produs

**Descriere** - produsele vândute în magazine

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea produsului
- nume - tip de dată varchar(20), numele produsului
- pret - tip de dată float, pretul produsului

## Produs\_Raion

**Descriere** - tabel asociativ care face legătura între produse și raioane

**Atribute și chei:**

- idProdus - tip de dată int, **cheie străină** către coloana id din tabelul Produs
- idRaion - tip de dată int, **cheie străină** către coloana id din tabelul Raion
- ambele reprezintă o **cheie primară** compusă

## Oferta

**Descriere** - ofertele aplicate produselor din magazine

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea ofertei
- nume - tip de dată varchar(20), numele produsului
- dataInceput - tip de dată date, data exactă la care începe oferta
- dataFinal - tip de dată date, data exactă la care se termină oferta
- procentajScazut - tip de dată float, procentajul care se scade din prețul produsului

## Istoric\_Oferte

**Descriere** - tabel asociativ care face legătura între produse și oferte, și menține un istoric

**Atribute și chei:**

- idProdus - tip de dată int, **cheie străină** către coloana id din tabelul Produs
- idOferta - tip de dată int, **cheie străină** către coloana id din tabelul Oferta
- ambele reprezintă o **cheie primară** compusă

## Angajat

**Descriere** - angajații care lucrează la magazine

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea angajatului
- nume - tip de dată varchar(20), numele angajatului
- prenume - tip de dată varchar(20), prenumele produsului
- dataAngajarii - tip de dată date, prima dată la care a început să lucreze angajatul (data în care este adăugat în baza de date)

## Post

**Descriere** - posturile pe care le pot avea angajații

**Atribute și chei:**

- id - tip de dată int, auto-increment, **cheie primară** care definește unicitatea postului
- titlu - tip de dată varchar(20), titlul propriu al postului

- salariuBrut - tip de dată float, salariul pe care îl are postul
- programStart - tip de dată varchar(10), ora la care acest post începe munca
- programFinal - tip de dată varchar(10), ora la care acest post termină munca

### **Contract**

**Descriere** - tabel asociativ care face legătura dintre angajat, post și magazin, și menține un istoric al legăturilor

#### **Atribute și chei:**

- idAngajat - tip de dată int, **cheie străină** către coloana id din tabelul Angajat
- dataInceput - tip de dată date, data la care angajatul începe sa lucreze la acest magazin pe acest post
- dataFinal - tip de dată date, data la care angajatul termină sa lucreze la acest magazin pe acest post
- idMagazin - tip de dată int, **cheie străină** către coloana id din tabelul Magazin
- idPost - tip de dată int, **cheie străină** către coloana id din tabelul Post
- (idAngajat, dataInceput) - **cheie primară** compusă

### **Descrierea relațiilor și a cardinalităților**

#### **Franciza conține magazine - relație one-to-many**

- O franciza conține 0 sau mai multe magazine.
- Un magazin aparține de maxim o franciza și minim 0.
- **Cardinalitate** minimă 0:0, maximă M:1

#### **Magazinul conține raioane - relație one-to-many**

- Un magazin conține 0 sau mai multe raioane.
- Un raion aparține de maxim un magazin și minim unul.
- **Cardinalitate** minimă 0:1, maximă M:1

#### **Raioanele conțin produse - relație many-to-many**

- Un raion conține 0 sau mai multe produse.
- Un produs se vinde în 0 sau mai multe raioane.
- **Cardinalitate** minimă 0:0, maximă M:N

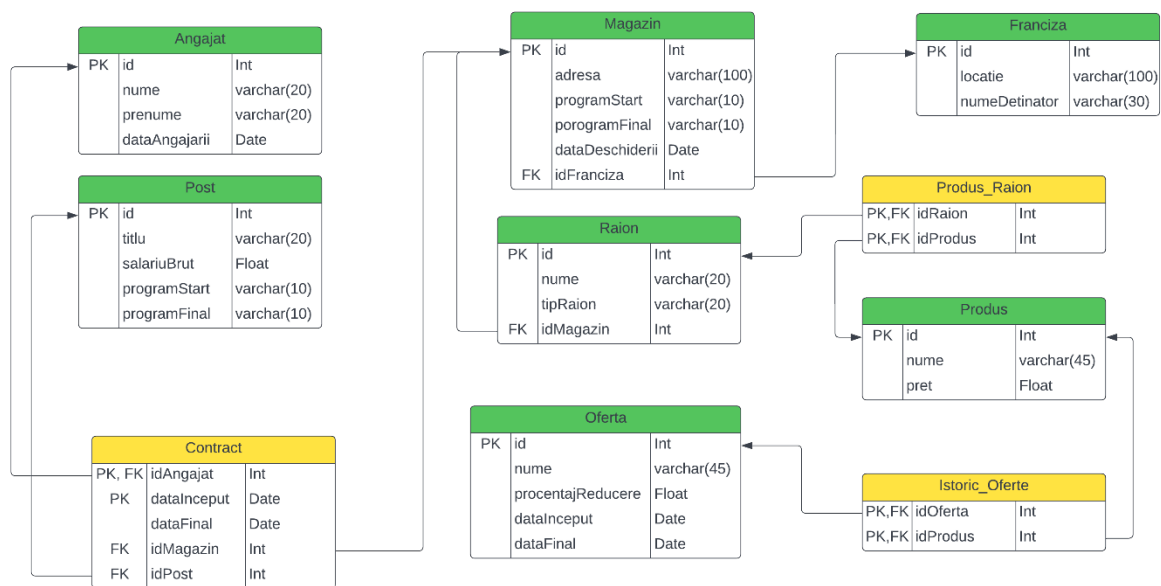
#### **Produsele au oferte - relație many-to-many**

- Un produs are 0 sau mai multe oferte.
- O oferta se aplică la 0 sau mai multe produse.
- **Cardinalitate** minimă 0:0, maximă M:N

#### **Angajatul lucrează ca un post la un magazin - relație de tip 3**

- Această relație este făcută prin intermediul unui tabel asociativ **Contracte** în care sunt păstrate toate contractele angajaților curenți
- Un angajat a lucrat la 0 sau mai multe magazine.
- Un post este aplicat la 0 sau mai mulți angajați.
- Un magazin angajează 0 sau mai mulți angajați.
- **Cardinalitate** minimă 0:0, 0:0, 0:0, maximă M:N, M:N, M:N

## Diagrama conceptuala



## Constrângeri de integritate

### Franciza

id - constrângere PRIMARY KEY, auto\_increment

locatie - constrângere not null

numeDetinator - constrângere not null

### Magazin

id - constrângere PRIMARY KEY, auto\_increment

adresa - constrângere not null

programStart - constrângere not null

programFinal - constrângere not null

dataDeschiderii - constrângere not null, default current\_timestamp

idFranciza - constrângere FOREIGN KEY către id din tabelul Franciza cu specificațiile ON DELETE SET NULL și ON UPDATE SET NULL

### Raion

id - constrângere PRIMARY KEY, auto\_increment

nume - constrângere not null

tipRaion - constrângere not null

idMagazin - constrângere not null

### Produs

id - constrângere PRIMARY KEY, auto\_increment

nume - constrângere not null

pret - constrângere not null, check pret mai mare ca 0

### Produs\_Raion

idProdus - constrângere not null, PRIMARY KEY, FOREIGN KEY către id din tabelul Produs cu specificațiile ON DELETE CASCADE și ON UPDATE CASCADE



idRaion - constrângere not null, PRIMARY KEY, FOREIGN KEY către id din tabelul Raion cu specificațiile ON DELETE CASCADE și ON UPDATE CASCADE

### **Oferta**

id - constrângere PRIMARY KEY, auto\_increment

nume - constrângere not null

dataInceput - constrângere not null

dataFinal - constrângere not null

procentajReducere - constrângere not null, check procentajReducere intre 0 si 1

### **Istoric\_Oferte**

idProdus - constrângere not null, PRIMARY KEY, FOREIGN KEY către id din tabelul Produs cu specificațiile ON DELETE CASCADE și ON UPDATE CASCADE

idOferta - constrângere not null, PRIMARY KEY, FOREIGN KEY către id din tabelul Oferta cu specificațiile ON DELETE CASCADE și ON UPDATE CASCADE

### **Angajat**

id - constrângere PRIMARY KEY, auto\_increment

nume - constrângere not null

prenume - constrângere not null

dataAngajari - constrângere not null, default current\_timestamp

### **Post**

id - constrângere PRIMARY KEY, auto\_increment

titlu - constrângere not null

salariu - constrângere not null, check salariu mai mare ca 0

programStart - constrângere not null

programFinal - constrângere not null

### **Contract**

idAngajat - constrângere not null, PRIMARY KEY, FOREIGN KEY către id din tabelul Angajat cu specificațiile ON DELETE CASCADE și ON UPDATE CASCADE

dataInceput - constrângere not null, PRIMARY KEY

idPost - constrângere FOREIGN KEY către id din tabelul Post cu specificațiile ON DELETE SET NULL și ON UPDATE SET NULL

idMagazin - constrângere FOREIGN KEY către id din tabelul Magazin cu specificațiile ON DELETE SET NULL și ON UPDATE SET NULL

dataFinal - constrângere not null

## **Scheme relaționale:**

**Franciza**(id#, locatie, numeDetinator);

**Magazin**(id#, adresa, programStart, programFinal, dataDeschiderii, idFranciza#);

**Raion**(id#, nume, tipRaion, idMagazin#);

**Produs**(id#, nume, pret);

**Oferta**(id#, nume, procentajReducere, dataInceput, dataFinal);

**Produs\_Raion**(idRaion#, idProdus#);

**Istoric\_Oferte**(idOferta#, idProdus#);

**Angajat**(id#, nume, prenume, dataAngajarii);  
**Post**(id#, titlu, salariuBrut, programStart, programFinal);  
**Contract**(idAngajat#, dataInceput#, dataFinal, idMagazin#, idPost#);

## Crearea tabelelor

Baza de date a fost creată în MySQL, cu ajutorul programului XAMPP, PhpMyAdmin. Tabelele sunt create in ordinea de mai jos.

### Tabelul franciza

```
create table franciza
(
    id int auto_increment primary key,
    locatie varchar(100) not null,
    numeDetinator varchar(30) not null
)
engine = InnoDB;
```

### Tabelul magazin

```
create table magazin
(
    id int auto_increment primary key,
    adresa varchar(100) not null,
    programStart varchar(10) not null,
    programFinal varchar(10) not null,
    dataDeschiderii date not null default current_timestamp,
    idFranciza int,
    foreign key (idFranciza) references franciza(id)
    on delete set null
    on update set null
)
engine = InnoDB;
```

### Tabelul angajat

```
create table angajat
(
    id int auto_increment primary key,
    nume varchar(20) not null,
    prenume varchar(20) not null,
    dataAngajarii date not null default current_timestamp
)
engine = InnoDB;
```

### Tabelul post

```
create table post
(
```

```

    id int auto_increment primary key,
    titlu varchar(20) not null,
    salariu float not null check (salariu > 0),
    programStart varchar(10) not null,
    programFinal varchar(10) not null
)
engine = InnoDB;

```

### **Tabelul contract**

```

create table contract
(
    dataInceput date not null,
    dataFinal date not null,
    idAngajat int not null,
    idPost int,
    idMagazin int,
    foreign key (idAngajat) references angajat(id)
    on delete cascade
    on update cascade,
    foreign key (idPost) references post(id)
    on delete set null
    on update set null,
    foreign key (idMagazin) references magazin(id)
    on delete set null
    on update set null,
    index id (idAngajat, dataInceput),
    primary key (idAngajat, dataInceput), -- cheie compusa
    constraint contract_data_chk check (dataInceput < dataFinal)
)
engine = InnoDB;

```

### **Tabelul raion**

```

create table raion
(
    id int auto_increment primary key,
    nume varchar(20) not null,
    tipRaion varchar(20) not null,
    idMagazin int not null,
    foreign key (idMagazin) references magazin(id)
    on delete cascade
    on update cascade
)
engine = InnoDB;

```

### **Tabelul oferta**

```

create table oferta

```

```
(
  id int auto_increment primary key,
  nume varchar(45) not null,
  dataInceput date not null,
  dataFinal date not null,
  procentajReducere float not null check (procentajReducere > 0 and procentajReducere <
1),
  constraint oferta_data_chk check (dataInceput < dataFinal)
)
engine = InnoDB;
```

### **Tabelul produs**

```
create table produs
(
  id int auto_increment primary key,
  nume varchar(45) not null,
  pret float not null check (pret > 0)
)
engine = InnoDB;
```

### **Tabelul produs\_raion**

```
create table produs_raion
(
  idProdus int not null,
  idRaion int not null,
  primary key (idProdus, idRaion), -- cheie compusa
  foreign key (idProdus) references produs(id)
  on delete cascade
  on update cascade,
  foreign key (idRaion) references raion(id)
  on delete cascade
  on update cascade
)
engine = InnoDB;
```

### **Tabelul istoric\_oferte**

```
create table istoric_oferte
(
  idProdus int not null,
  idOferta int not null,
  primary key (idProdus, idOferta), -- cheie compusa
  foreign key (idProdus) references produs(id)
  on delete cascade
  on update cascade,
  foreign key (idOferta) references oferta(id)
)
```

```
on delete cascade
on update cascade
)
engine = InnoDB;
```

## Inserarea datelor în baza de date

Am introdus cel puțin câte un rând în fiecare tabel, pentru a arăta procedura inserției de date în această bază de date.

```
INSERT INTO `franciza` (`locatie`, `numeDetinator`)
VALUES ('Romania, Bucuresti', 'Schmidt Robert');
```

```
INSERT INTO `franciza` (`locatie`, `numeDetinator`)
VALUES
('Romania, Cluj-Napoca', 'Doe John'),
('Romania, Timisoara', 'Smith Jane'),
('Romania, Iasi', 'Brown Michael');
```

```
INSERT INTO `magazin` (`adresa`, `programStart`, `programFinal`, `dataDeschiderii`,
`idFranciza`)
VALUES ('Str. Drumul Taberei, Nr. 53', '10:00', '22:00', current_timestamp(), '1');
```

```
INSERT INTO `magazin` (`adresa`, `programStart`, `programFinal`, `dataDeschiderii`,
`idFranciza`)
VALUES ('Str. Mihai Viteazul, Nr. 21', '09:00', '20:00', current_timestamp(), '1'),
('Str. 1 Mai, Nr. 2', '08:00', '18:00', current_timestamp(), '2'),
('Str. Mihai Eminescu, Nr. 3', '08:00', '18:00', current_timestamp(), '1'),
('Str. Ion Creanga, Nr. 4', '08:00', '18:00', current_timestamp(), '1'),
('Str. Vasile Alecsandri, Nr. 5', '08:00', '18:00', current_timestamp(), '2'),
('Str. Cara Anghel, Nr. 4', '08:00', '18:00', current_timestamp(), '2'),
('Soseaua Panduri, Nr. 4', '08:00', '18:00', current_timestamp(), '2'),
('Str. Mihail Sebastian, Nr. 5', '08:00', '18:00', current_timestamp(), '2'),
('Str. Iovita, Nr. 4', '08:00', '18:00', current_timestamp(), '3');
```

```
INSERT INTO `angajat` (`nume`, `prenume`, `dataAngajarii`)
VALUES ('Schmidt', 'Helmuth', current_timestamp());
```

```
INSERT INTO `angajat` (`nume`, `prenume`, `dataAngajarii`)
VALUES ('Popescu', 'Alex', current_timestamp()),
('Gheorghe', 'Bogdan', '2022-12-27'),
('Mihai', 'Iulian', current_timestamp());
```

```
INSERT INTO `post` (`titlu`, `salariu`, `programStart`, `programFinal`)
VALUES ('Manager', '10000', '9:00', '16:00');
```

```
INSERT INTO `post` (`titlu`, `salariu`, `programStart`, `programFinal`)
VALUES ('Casier', '5000', '10:00', '18:00'),
      ('Lucrator comercial', '6500', '08:00', '16:00'),
      ('Lucrator comercial', '6500', '14:00', '20:00');
```

```
INSERT INTO `contract` (`dataInceput`, `dataFinal`, `idAngajat`, `idPost`, `idMagazin`)
VALUES ('2022-12-27', '2024-12-01', '1', '1', '1');
```

```
INSERT INTO `contract` (`dataInceput`, `dataFinal`, `idAngajat`, `idPost`, `idMagazin`)
VALUES ('2022-12-27', '2024-12-01', '2', '2', '1'),
      ('2022-12-27', '2024-12-01', '3', '3', '1'),
      ('2022-12-27', '2024-12-01', '4', '4', '1');
```

```
INSERT INTO `raion` (`nume`, `tipRaion`, `idMagazin`)
VALUES ('Fructe', 'perisabile', '1');
```

```
INSERT INTO `raion` (`nume`, `tipRaion`, `idMagazin`)
VALUES ('Carne', 'congelate', '1'),
      ('Lactate', 'perisabile', '1'),
      ('Bauturi', 'non-perisabile', '1');
```

```
INSERT INTO `produs` (`nume`, `pret`)
VALUES ('Mere', '2.4');
```

```
INSERT INTO `produs` (`nume`, `pret`)
VALUES ('Pere', '2.8');
```

```
INSERT INTO `produs` (`nume`, `pret`)
VALUES ('Ceafa de porc', '35.99'),
      ('Suc', '2.99'),
      ('Lapte', '3.99'),
      ('Vin', '15.99');
```

```
INSERT INTO `produs_raion` (`idProdus`, `idRaion`)
VALUES ('1', '1');
```

```
INSERT INTO `produs_raion` (`idProdus`, `idRaion`)
VALUES ('2', '1');
```

```
INSERT INTO `produs_raion` (`idProdus`, `idRaion`)
VALUES ('3', '2');
```

```
INSERT INTO `produs_raion` (`idProdus`, `idRaion`)
VALUES ('4', '4'),
      ('5', '3');
```

('6', '4');

```
INSERT INTO `oferta` (`nume`, `dataInceput`, `dataFinal`, `procentajReducere`)  
VALUES ('Carnuri inainte de revelion', '2022-12-30', '2022-12-31', '0.25');
```

```
INSERT INTO `oferta` (`nume`, `dataInceput`, `dataFinal`, `procentajReducere`)  
VALUES ('Produse non-perisabile inainte de revelion', '2022-12-30', '2022-12-31', '0.15'),  
('Lapte flash sale', '2023-01-01', '2023-01-02', '0.30');
```

```
INSERT INTO `istoric_oferte` (`idProdus`, `idOferta`)  
VALUES ('3', '1');
```

```
INSERT INTO `istoric_oferte` (`idProdus`, `idOferta`)  
VALUES ('5', '3'),  
('6', '2'),  
('5', '2'),  
('4', '2');
```