UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT**

**BAZE DE DATE**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU-LAURENȚIU

STUDENT:

BĂNESARU DENISA-GEORGIANA

BUCUREȘTI

2022

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**GESTIONAREA UNUI**

**DEPOZIT DE PRODUSE COSMETICE**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU-LAURENȚIU

STUDENT:

BĂNESARU DENISA-GEORGIANA

BUCUREȘTI

2022

**CUPRINS**

[Prezentarea modelului din viața reală 6](#_Toc8618)

[1. Regulile modelului 6](#_Toc2139)

[2. Diagrama Entitate-Relație 8](#_Toc16927)

[2.1. Reprezentarea diagramei 8](#_Toc25787)

[2.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților 8](#_Toc222)

[3.2.1 Descrierea entităților, atributelor, cheilor 8](#_Toc1415)

[3.2.1.1. Tabelul PRODUCATOR 8](#_Toc32132)

[3.2.1.2 Tabelul APROVIZIONARE 9](#_Toc4115)

[3.2.1.4 Tabelul COSMETICE 10](#_Toc897)

[3.2.1.6 Tabelul ANGAJAT 11](#_Toc6102)

[3.2.1.7 Tabelul TRANSPORT 12](#_Toc32555)

[3.2.1.8 Tabelul MAGAZIN 12](#_Toc19358)

[3.2.2 Descrierea relațiilor și a cardinalităților 13](#_Toc20644)

[3.2.2.1 PRODUCATOR - APROVIZIONARE 13](#_Toc28008)

[3.2.2.2 APROVIZIONARE - COSMETICE 13](#_Toc4810)

[3.2.2.3 COSMETICE - COMANDA 13](#_Toc29194)

[3.2.2.4 COMANDA - ANGAJAT 14](#_Toc24261)

[3.2.2.5 COMANDA - TRANSPORT 14](#_Toc20238)

[3.2.2.6 COMANDA - MAGAZIN 15](#_Toc6389)

[4. Diagrama Conceptuală 15](#_Toc28776)

[4.1 Reprezentarea diagramei 15](#_Toc31302)

[4.2 Descrierea constrângerilor de integritate 17](#_Toc23799)

[4.2.1 Tabelul PRODUCATOR 17](#_Toc24888)

[4.2.2 Tabelul APROVIZIONARE 17](#_Toc2027)

[4.2.3 Tabelul COSMETICE 17](#_Toc4126)

[4.2.4 Tabelul FACTURA\_A 18](#_Toc32396)

[4.2.5 Tabelul COMANDA 18](#_Toc27235)

[4.2.6 Tabelul FACTURA\_C 19](#_Toc23789)

[4.2.7 Tabelul ANGAJAT 20](#_Toc4303)

[4.2.8 Tabelul TRANSPORT 20](#_Toc2456)

[4.2.9 Tabelul MAGAZIN 20](#_Toc32037)

[4.3 Descierea entităților, atributelor și cheilor pentru tabelele asociative 21](#_Toc30515)

[4.3.1 Tabelul FACTURA\_A 21](#_Toc32664)

[4.3.1 Tabelul FACTURA\_C 21](#_Toc25772)

[4.4 Scheme relaționale 22](#_Toc21724)

[4.4.1 Reprezentarea schemelor relaționale 22](#_Toc6002)

[4.4.2 Descrierea constrângerilor CASCADE, SET NULL 23](#_Toc491)

[Constrângeri de tipul ON DELETE / ON UPDATE : 23](#_Toc10434)

[5. Scriptul SQL 24](#_Toc19116)

[5.1 Etapa de DROP 24](#_Toc1554)

[5.2 Crearea bazei de date 24](#_Toc801)

[5.3 Crearea tabelului PRODUCATOR 24](#_Toc1568)

[5.4 Crearea tabelului APROVIZIONARI 25](#_Toc14030)

[5.5 Crearea tabelului COSMETICE 25](#_Toc16177)

[5.6 Crearea tabelului FACTURA\_A 26](#_Toc16493)

[5.7 Crearea tabelului COMANDA 27](#_Toc28799)

[5.8 Crearea tabelului FACTURA\_C 29](#_Toc31787)

[5.9 Crearea tabelului ANGAJAT 30](#_Toc8171)

[5.10 Crearea tabelului TRANSPORT 30](#_Toc20750)

[5.11 Crearea tabelului MAGAZIN 30](#_Toc22125)

[5.12 Inserarea datelor în tabelul PRODUCATOR 31](#_Toc29226)

[5.13 Inserarea datelor în tabelul APROVIZIONARE 31](#_Toc30937)

[5.14 Inserarea datelor în tabelul COSMETICE 31](#_Toc32457)

[5.15 Inserarea datelor în tabelul FACTURA\_A 32](#_Toc26608)

[5.16 Inserarea datelor în tabelul ANGAJAT 32](#_Toc8390)

[5.17 Inserarea datelor în tabelul MAGAZIN 32](#_Toc10711)

[5.18 Inserarea datelor în tabelul TRANSPORT 32](#_Toc663)

[5.19 Inserarea datelor în tabelul COMANDA 33](#_Toc12396)

[5.20 Inserarea datelor în tabelul FACTURA\_C 33](#_Toc739)

# Prezentarea modelului din viața reală

Tema aleasă pentru proiectul la disciplina Baze de Date este Gestionarea unui depozit de produse cosmetice. Menționez că baza de date este pentru un depozit din România iar comenzile administrate se livrează tot pe teritoriul României. Prin intermediul bazei de date urmăresc parcursul produselor cosmetice de la producător la depozit și mai apoi la magazine.

De transportul comenzilor de aprovizionare se ocupă producătorul iar de transportul comenzilor la magazine se ocupă depozitul cu ajutorul firmelor de curierat.

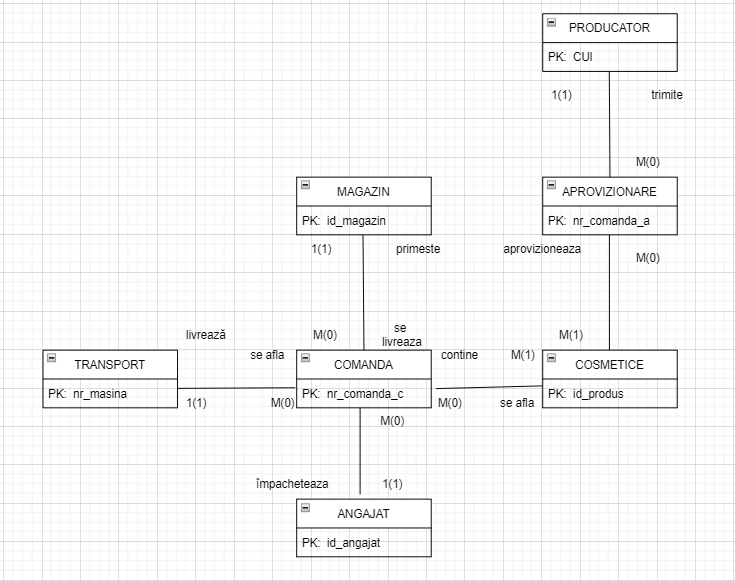
Am realizat această bază de date cu scopul de a administra comenzile de aprovizionare a stocului depozitului dar și pentru gestionarea comenzilor plasate de către magazinele cu care depozitul a încheiat un contract.

1. Regulile modelului

* Un producător poate trimite mai multe comenzi de aprovizionare dar există și producători care nu trimit nici o comandă.
* O comandă de aprovizionare trebuie să fie trimisă de un singur producător.
* O comandă de aprovizionare poate avea mai multe cosmetice.
* Un produs cosmetic se poate află în mai multe comenzi de aprovizionare dar sunt produse care nu se află în comenzile de aprovizionare.
* O comandă către un magazin poate avea mai multe cosmetice.
* Un produs cosmetic se poate afla în mai multe comenzi sau în nici o comanda. (nu a fost comandat încă);
* Un angajat poate împacheta una sau mai multe comenzi însă pot fi angajați care nu împachetează nici o comandă. (de exemplu un angajat nou);
* Comanda trebuie să fie împachetată de un singur angajat.
* O comandă trebuie să se afle într-un singur transport.
* Într-o livrare se pot afla una sau mai multe comenzi dar sunt și mașini care nu livrează. (de exemplu nu sunt destule comenzi sau toți șoferii sunt deja în cursă);
* Magazinul poate primi una sau mai multe comenzi. (dar există magazine care încă nu au dat comandă).
* O comandă trebuie să fie trimisă la un singur magazin.

1. Diagrama Entitate-Relație

## Reprezentarea diagramei



## Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

### 3.2.1 Descrierea entităților, atributelor, cheilor

#### 3.2.1.1. Tabelul PRODUCATOR

Tabelul PRODUCATOR stochează detalii despre producătorii care alimentează depozitul.

Structura tabelului PRODUCATOR este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | CUI | DECIMAL(8,0) | Codul Unic de Inregistrare |
|  | denumire | VARCHAR(35) | numele firmei |
|  | brand | VARCHAR(25) | brand-ul produselor |
|  | localitate | VARCHAR(25) | adresă |
|  | strada | VARCHAR(25) | adresă |

#### 

#### 3.2.1.2 Tabelul APROVIZIONARE

Tabelul APROVIZIONARE conține date despre comenzile care aprovizionează cu produse care nu mai sunt pe stoc .

Structura tabelului APROVIZIONARE este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | nr\_comanda\_a | DECIMAL(6,0) | identificatorul comenzii de aprovizionare |
| FK | CUI | DECIMAL(8,0) | Codul Unic de Inregistrare |
|  | data\_primire | DATE | data |

#### 3.2.1.4 Tabelul COSMETICE

Tabelul COSMETICE stochează informații cu privire la produsele aflate în depozit.

Structura tabelului COSMETICE este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | id\_produs | DECIMAL(4,0) | identificatorul unic al produsului |
|  | nume | VARCHAR(25) | numele produsului |
|  | categorie | VARCHAR(25) | categoria produsului |
|  | stoc | DECIMAL(4,0) | numărul de produse |
| PK | pret\_unit | DECIMAL(6,2) | prețul pentru un produs |
|  | data\_expirare | DATE | data de expirare |

3.2.1.5 Tabelul COMANDA

Tabelul COMANDA stochează datele despre comenzile plasate de către magazine.

Structura tabelului COMANDA este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | nr\_comanda\_c | DECIMAL(6,0) | identificatorul unic al comenzii |
|  | data\_plasare | DATE | dată |
|  | data\_expediere | DATE | dată |
| FK | id\_magazin | DECIMAL(4,0) | identificatorul unic al magazinului |
| FK | nr\_masina | VARCHAR(7) | numărul mașinii firmei |
| FK | id\_angajat | DECIMAL(4,0) | identificatorul unic al angajatului |

#### 3.2.1.6 Tabelul ANGAJAT

Tabelul ANGAJAT memorează datele despre angajații care se ocupă de împachetarea comenzilor.

Structura tabelului ANGAJAT este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | id\_angajat | DECIMAL(4,0) | identificatorul unic al angajatului |
|  | nume | VARCHAR(25) | date angajat |
|  | prenume | VARCHAR(25) | date angajat |

#### 3.2.1.7 Tabelul TRANSPORT

Tabelul TRANSPORT memorează informații despre firmele de curierat care vor livra coletele.

Structura tabelului TRANSPORT este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | nr\_masina | VARCHAR(7) | numele firmei |
|  | firma\_curierat | VARCHAR(25) | date angajat |
|  | id\_sofer | DECIMAL(4,0) | identificatorul unic al șoferului |

#### 3.2.1.8 Tabelul MAGAZIN

Tabelul MAGAZIN conține informații despre locațiile magazinelor care plasează comenzi către depozit.

Structura tabelului MAGAZIN este următoarea

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | id\_magazin | DECIMAL(4,0) | identificatorul unic al magazinului |
|  | nume | VARCHAR(25) | date magazin |
|  | localitate | VARCHAR(25) | adresă |
|  | strada | VARCHAR(25) | adresă |
|  | numar | DECIMAL(4,0) | adresă |

### 3.2.2 Descrierea relațiilor și a cardinalităților

#### 3.2.2.1 PRODUCATOR - APROVIZIONARE

**Relația:** PRODUCĂTORII trimit comenzi de APROVIZIONARE.

**Cardinalități:**

Cardinalități tabel *producator*:

1. Un producător poate trimite mai multe comenzi → M (c. maximală)
2. Un producător poate să nu trimită nici o comandă → 0 (c. minimală)

Cardinalități tabel *aprovizionare*:

1. O aprovizionarea trebuie făcută de un singur producător → 1 (c. maximală și minimală)

#### 3.2.2.2 APROVIZIONARE - COSMETICE

**Relația:** Comenzile de APROVIZIONARE alimentează cu COSMETICE depozitul.

**Cardinalități:**

Cardinalități tabel *aprovizionare*:

1. O comandă poate avea mai multe produse → M (c. maximală)
2. O comandă trebuie să aibă minim un produs → 1 (c. minimală)

Cardinalități tabel *cosmetice*:

1. Un produs cosmetic se poate afla în mai multe comenzi → M (c. maximală)
2. Există produse care nu se află în comenzi de aprovizionare → 0 (c. minimală)

#### 3.2.2.3 COSMETICE - COMANDA

**Relația:** COMANDA conține produse COSMETICE .

**Cardinalități:**

Cardinalități tabel *cosmetice*:

1. Într-o comandă pot fi mai multe produse → M (c. maximală)
2. Există produse care nu se află în comenzi → 0 (c. minimală)

Cardinalități tabel *comandă*:

1. O comandă poate avea mai multe produse → M (c. maximală)
2. O comandă trebuie să aibă minim un produs → 1 (c. minimală)

#### 3.2.2.4 COMANDA - ANGAJAT

**Relația:** COMANDA este împachetată de către un ANGAJAT .

**Cardinalități:**

Cardinalități tabel *comandă*:

1. O comandă trebuie să fie împachetată de un singur angajat → 1 (c. maximală și minimală)

Cardinalități tabel *angajat*:

1. Un angajat poate împacheta mai multe comenzi → M (c. maximală)
2. Există angajați care nu împachetează comenzi? Da → 0 (c. minimală)

#### 3.2.2.5 COMANDA - TRANSPORT

**Relația:** COMANDA se află într-un TRANSPORT .

**Cardinalități:**

Cardinalități tabel *comandă*:

1. În câte transporturi se poate afla o comandă? Unul singur → 1 (c. maximală și minimală)

Cardinalități tabel *transport*:

1. Un transport poate livra mai multe comenzi → M (c. maximală)
2. Există mașini care nu livrează comenzi? Da → 0 (c. minimală)

#### 3.2.2.6 COMANDA - MAGAZIN

**Relația:** COMANDA se livrează către MAGAZIN .

**Cardinalități:**

Cardinalități tabel *comandă*:

1. La câte magazine se poate livra o comandă? Unul singur → 1 (c. maximală și minimală)

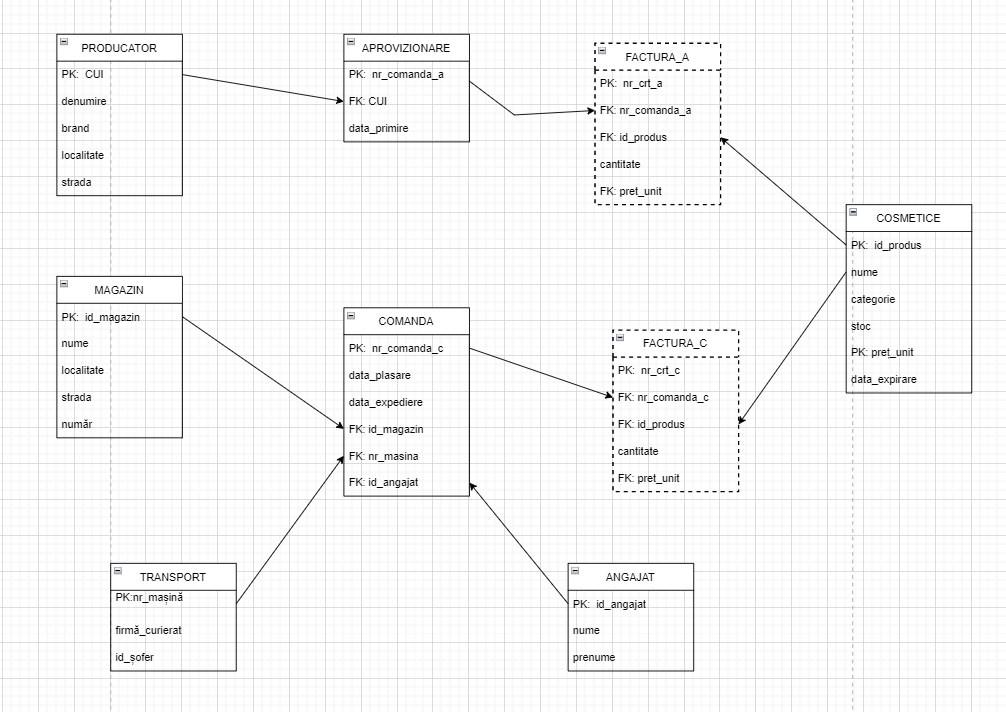
Cardinalități tabel *magazin*:

1. Un magazin poate primi mai multe comenzi → M (c. maximală)

2. Există magazine care nu au primit comenzi? Da → 0 (c. minimală)

# 4. Diagrama Conceptuală

## 4.1 Reprezentarea diagramei



## 4.2 Descrierea constrângerilor de integritate

### 4.2.1 Tabelul PRODUCATOR

* **Primary Key**
* Nume constrângere: prod\_CUI\_PK → atribut *CUI*
* **Unique**
* Pentru atributul*denumire* deoarece nu există firme cu același nume
* **Not Null**
* Pentru atributele *denumire, brand* și *localitate,* producătorul trebuie să aibă aceste atribute pentru a crea firma

### 4.2.2 Tabelul APROVIZIONARE

* **Primary Key**
* Nume constrângere: apr\_NR\_PK → atribut *nr\_comanda\_a*
* **Foreign Key**
* Nume constrângere: FK\_Aprovizionare → atribut CUI, referă tabelul producător
* **Not Null**
* Pentru atributele *CUI* și *data\_primire,* un producător trebuie să trimit comanda

### 4.2.3 Tabelul COSMETICE

* **Primary Key**
* Nume constrângere: id\_PK → atribut *id\_produs*
* **Not Null**
* Pentru atributele *nume, pret\_unit* și *data\_expirare,* un produs trebuie să aibă obligatoriu aceste atribute pentru a putea fi identificat
* **Check**
* Nume constrângere: pret\_const\_cosm → atribut *pret\_unit,* se verifică dacă prețul este pozitiv

### 4.2.4 Tabelul FACTURA\_A

* **Primary Key**
* Nume constrângere: fact\_a\_PK → atribut *nr\_crt\_a*
* **Foreign Key**
* Nume constrângere: FK\_Fact\_A\_nr → atribut *nr\_comanda\_a*, face legătura cu tabelul APROVIZIONARE și se atribuie o factură
* Nume constrângere: FK\_Fact\_A\_id\_pret → atribut *id\_produs, pret\_unit*, face legătura cu tabelul COSMETICE, determină id-ul și prețul produselor de pe factură
* **Not Null**
* Pentru atributele *nr\_comanda\_a* și *cantitate,* orice comandă trebuie să aibă minim un produs
* **Check**
* Nume constrângere: check\_fact\_A\_cantitate → atribut *cantitate,* obligă să fie pozitivă valoarea cantității

### 4.2.5 Tabelul COMANDA

* **Primary Key**
* Nume constrângere: nr\_com\_PK → atribut *nr\_comanda\_c*
* **Foreign Key**
* Nume constrângere: FK\_Angajat → atribut *id\_angajat*, face legătura cu tabelul ANGAJAT determină id-ul angajatului care a împachetat comanda
* Nume constrângere: FK\_Transport → atribut *nr\_masina*, referă tabelul TRANSPORT, determină numărul mașinii care va livra
* Nume constrângere: FK\_Magazin → atribut *id\_magazin*, referă tabelul MAGAZIN, determină id-ul magazinului care a plasat comanda
* **Not Null**
* Pentru atributele *data\_plasare* și *id\_magazin,* o comandă trebuie să ajungă la un magazin
* **Check**
* Nume constrângere: check\_data → atribute *data\_plasare, data\_expediere,* data plasării comenzii este obligatoriu să fie mai mică (sau egală) decât data expedierii

### 4.2.6 Tabelul FACTURA\_C

* **Primary Key**
* Nume constrângere: fact\_c\_PK → atribut *nr\_crt\_c*
* **Foreign Key**
* Nume constrângere: FK\_Fact\_C\_nr → atribut *nr\_comanda\_c*, face legătura cu tabelul COMANDA și se atribuie o factură
* Nume constrângere: FK\_Fact\_C\_id\_pret → atribut *id\_produs, pret\_unit*, face legătura cu tabelul COSMETICE, determină id-ul și prețul produselor de pe factură
* **Not Null**
* Pentru atributele *nr\_comanda\_c* și *cantitate,* orice comandă trebuie să aibă un număr de comandă și numărul de produse comandate
* **Check**
* Nume constrângere: check\_fact\_C\_cantitate → atribut *cantitate,* obligă să fie pozitivă valoarea cantității

### 4.2.7 Tabelul ANGAJAT

* **Primary Key**
* Nume constrângere: ang\_PK → atribut *id\_angajat*
* **Not Null**
* Pentru atributul *nume,* un angajat trebuie să aibă obligatoriu un nume de familie

### 4.2.8 Tabelul TRANSPORT

* **Primary Key**
* Nume constrângere: trans\_PK → atribut *nr\_masina*
* **Not Null**
* Pentru atributul *firma\_curierat,* este obligatoriu ca o mașină să aparțină unei firme
* **Check**
* Nume constrângere: trans\_nr\_check → atribut *nr\_masina,* prin funcția LENGTH(), se verifică dacă numărul mașinii are fix 7 caractere
* **Unique**
* Pentru atributul*id\_sofer* deoarece soferul se identifică printr-un număr unic

### 4.2.9 Tabelul MAGAZIN

* **Primary Key**
* Nume constrângere: magaz\_PK → atribut *id\_magazin*
* **Not Null**
* Pentru atributul *nume, localitate și strada,* este obligatoriu ca un magazin să aibă o adresă

## 4.3 Descierea entităților, atributelor și cheilor pentru tabelele asociative

### 4.3.1 Tabelul FACTURA\_A

Tabelul FACTURA\_A este un tabel asociativ în care se stochează date despre facturile de aprovizionare .

Structura tabelului FACTURA\_A este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | nr\_crt\_a | DECIMAL(3,0) | numărul curent |
| FK | nr\_comanda\_a | DECIMAL(6,0) | identificatorul unic al comenzii de aprovizionare |
| FK | id\_produs | DECIMAL(4,0) | Identificatorul unic al produsului |
|  | cantitate | DECIMAL(4,0) | numărul de produse |
| FK | pret\_unit | DECIMAL(6,2) | prețul |

### 4.3.1 Tabelul FACTURA\_C

Tabelul FACTURA\_C este un tabel asociativ în care se stochează date despre facturile comenzilor plasate de către magazine .

Structura tabelului FACTURA\_C este următoarea:

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Atribut | Tip | Descriere |
| PK | nr\_crt\_c | DECIMAL(3,0) | numărul curent |
| FK | nr\_comanda\_c | DECIMAL(6,0) | identificatorul unic al comenzii |
| FK | id\_produs | DECIMAL(4,0) | identificatorul unic al produsului |
|  | cantitate | DECIMAL(4,0) | numărul de produse |
| FK | pret\_unit | DECIMAL(6,2) | prețul |

## 4.4 Scheme relaționale

### 4.4.1 Reprezentarea schemelor relaționale

* PRODUCATOR(CUI(PK), denumire, brand, localitate, strada);
* APROVIZIONARE(nr\_comanda\_a (PK), CUI (FK), data\_primire);
* COSMETICE(id\_produs (PK) , nume , categorie, stoc, pret\_unit(PK), data\_expirare);
* FACTURA\_A( nr\_crt\_a (PK) , nr\_comanda\_a(FK) , id\_produs(FK) , cantitate, pret\_unit(FK));
* COMANDA(nr\_comanda\_c (PK), data\_plasare, data\_expediere, id\_magazin (FK) , nr\_masina (FK), id\_angajat (FK));
* FACTURA\_C( nr\_crt\_c (PK) , nr\_comanda\_c(FK) , id\_produs(FK) , cantitate, pret\_unit(FK));
* ANGAJAT(id\_angajat (PK) , nume , prenume);
* TRANSPORT(nr\_masina (PK), firma\_curierat, id\_sofer);
* MAGAZIN(id\_magazin (PK) , nume , localitate, strada , numar)

### 4.4.2 Descrierea constrângerilor CASCADE, SET NULL

## Constrângeri de tipul ON DELETE / ON UPDATE :

Pentru tabelul APROVIZIONARE:

* Nume : FK\_Aprovizionare (CUI) ON DELETE CASCADE ON UPDATE CASCADE - când un producător este șters din baza de date se vor șterge toate comenzile de aprovizionare care conțin CUI-ul respectiv, la fel se întâmplă dacă se actualizează;

Pentru tabelul FACTURA\_A:

* Nume : FK\_Fact\_A\_id\_pret (id\_produs, pret\_unit) ON DELETE SET NULL - dacă un produs este șters din tabelul cosmetice atunci se va seta null pe facturile de aprovizionare (nu se poate șterge produsul de pe facturi deja încasate);
* Nume : FK\_Fact\_A\_nr (nr\_comanda\_a) ON DELETE CASCADE ON UPDATE CASCADE - dacă o comandă este anulată se va șterge și factura corespunzătoare

Pentru tabelul FACTURA\_C:

* Nume : FK\_Fact\_C\_id\_pret (id\_produs, pret\_unit) ON DELETE SET NULL - dacă un produs nu mai există în depozit se setează null pe facturi (produsul poate reveni);
* Nume : FK\_Fact\_C\_nr (nr\_comanda\_c) ON DELETE CASCADE ON UPDATE CASCADE - dacă o comandă este anulată se va șterge și factura corespunzătoare;

Pentru tabelul COMANDA:

* Nume : FK\_Angajat (id\_angajat ) ON DELETE CASCADE ON UPDATE CASCADE - la ștergerea unui angajat se vor elimina comenzile împachetate de el;
* Nume : FK\_Transport (nr\_masina) ON DELETE CASCADE ON UPDATE CASCADE - când se șterge o mașină din baza de date se șterge și din comandă;
* Nume : FK\_Magazin (id\_magazin) ON DELETE CASCADE ON UPDATE CASCADE - la eliminarea unui magazin se vor anula comenzile aferente;

# 5. Scriptul SQL

Pentru script am folosit MySQL Workbench 8.0 CE, iar pentru bază Local instance MySQL80.

## 5.1 Etapa de DROP

Prima etapă constă în ștergerea tuturor tabelelor și a bazei de date pentru a putea fi rulat scriptul de câte ori se dorește.

**drop table** factura\_a;

**drop table** aprovizionare;

**drop table** producator;

**drop table** factura\_c;

**drop table** cosmetice;

**drop table** comanda;

**drop table** angajat;

**drop table** transport;

**drop database** DEPOZIT;

## 5.2 Crearea bazei de date

Pentru a crea baza de date am folosit două comenzi:

**CREATE DATABASE** DEPOZIT;

**USE** DEPOZIT;

## 5.3 Crearea tabelului PRODUCATOR

**CREATE TABLE** producator

( CUI **DECIMAL(8,0)**

, denumire **VARCHAR(35) NOT NULL UNIQUE**

, brand **VARCHAR(25) NOT NULL**

, localitate **VARCHAR(25) NOT NULL**

, strada **VARCHAR(25)**

, **CONSTRAINT** prod\_CUI\_PK **PRIMARY KEY** (CUI)

);

## 5.4 Crearea tabelului APROVIZIONARI

**CREATE TABLE** aprovizionare

( nr\_comanda\_a **DECIMAL(6,0)**

, CUI **DECIMAL(8,0) NOT NULL**

, data\_primire **DATE NOT NULL**

);

**ALTER TABLE** aprovizionare

**ADD** (

**CONSTRAINT** apr\_NR\_PK  **PRIMARY KEY** (nr\_comanda\_a)

, **CONSTRAINT** FK\_Aprovizionare

**FOREIGN KEY** (CUI) REFERENCES producator(CUI) **ON DELETE CASCADE ON UPDATE CASCADE**

);

## 5.5 Crearea tabelului COSMETICE

**CREATE TABLE** cosmetice

( id\_produs **DECIMAL(4,0)**

, nume **VARCHAR(25) NOT NULL**

, categorie **VARCHAR(25)**

, stoc **DECIMAL(4,0)**

, pret\_unit **DECIMAL(6,2) NOT NULL**

, data\_expirare **DATE NOT NULL**

);

**ALTER TABLE** cosmetice

**ADD** (

**CONSTRAINT** id\_PK **PRIMARY KEY** (id\_produs, pret\_unit)

, **CONSTRAINT** pret\_const\_cosm

**CHECK** (pret\_unit > 0)

);

## 5.6 Crearea tabelului FACTURA\_A

**CREATE TABLE** factura\_a

( nr\_crt\_a **DECIMAL(3,0)**

, nr\_comanda\_a **DECIMAL(6,0) NOT NULL**

, id\_produs **DECIMAL(4,0)**

, cantitate **DECIMAL(4,0) NOT NULL**

, pret\_unit **DECIMAL(6,2)**

, **CONSTRAINT** fact\_a\_PK **PRIMARY KEY** (nr\_crt\_a)

);

**ALTER TABLE** factura\_a

**ADD** (

**CONSTRAINT** FK\_Fact\_A\_nr

**FOREIGN KEY** (nr\_comanda\_a) **REFERENCES** aprovizionare(nr\_comanda\_a) **ON DELETE CASCADE ON UPDATE CASCADE**,

**CONSTRAINT** FK\_Fact\_A\_id\_pret

**FOREIGN KEY** (id\_produs, pret\_unit) **REFERENCES** cosmetice(id\_produs, pret\_unit) **ON DELETE SET NULL**,

**CONSTRAINT** check\_fact\_A\_cantitate

**CHECK** (cantitate > 0)

);

## 5.7 Crearea tabelului COMANDA

**CREATE TABLE** comanda

( nr\_comanda\_c **DECIMAL(6,0)**

, data\_plasare **DATE NOT NULL**

, data\_expediere **DATE**

, id\_magazin **DECIMAL(4,0) NOT NULL**

, nr\_masina **VARCHAR(7) NOT NULL**

, id\_angajat **DECIMAL(4,0) NOT NULL**

);

**ALTER TABLE** comanda

**ADD** (

**CONSTRAINT** nr\_com\_PK **PRIMARY KEY** (nr\_comanda\_c) ,

**CONSTRAINT** check\_data

**CHECK** (data\_plasare <= data\_expediere)

);

* Menționez că aceste comenzi au fost rulate după crearea tabelelor ANGAJAT, TRANSPORT, MAGAZIN

**ALTER TABLE** comanda

**ADD CONSTRAINT** FK\_Angajat

**FOREIGN KEY** (id\_angajat ) **REFERENCES** angajat(id\_angajat) **ON DELETE CASCADE ON UPDATE CASCADE**;

**ALTER TABLE** comanda

**ADD CONSTRAINT** FK\_Transport

**FOREIGN KEY** (nr\_masina) **REFERENCES** transport(nr\_masina) **ON DELETE CASCADE ON UPDATE CASCADE**;

**ALTER TABLE** comanda

**ADD CONSTRAINT** FK\_Magazin

**FOREIGN KEY** (id\_magazin) REFERENCES magazin(id\_magazin) **ON DELETE CASCADE ON UPDATE CASCADE**;

## 5.8 Crearea tabelului FACTURA\_C

**CREATE TABLE** factura\_c

( nr\_crt\_c **DECIMAL(3,0)**

, nr\_comanda\_c **DECIMAL(6,0) NOT NULL**

, id\_produs **DECIMAL(4,0)**

, cantitate **DECIMAL(4,0) NOT NULL**

, pret\_unit **DECIMAL(6,2)**

, **CONSTRAINT** fact\_c\_PK **PRIMARY KEY** (nr\_crt\_c)

);

**ALTER TABLE** factura\_c

**ADD** (

**CONSTRAINT** FK\_Fact\_C\_id\_pret

**FOREIGN KEY** (id\_produs, pret\_unit) **REFERENCES** cosmetice(id\_produs, pret\_unit) **ON DELETE SET NULL**,

**CONSTRAINT** FK\_Fact\_C\_nr

**FOREIGN KEY** (nr\_comanda\_c) **REFERENCES** comanda(nr\_comanda\_c) **ON DELETE CASCADE ON UPDATE CASCADE**,

**CONSTRAINT** check\_fact\_C\_cantitate

**CHECK** (cantitate > 0)

);

## 5.9 Crearea tabelului ANGAJAT

**CREATE TABLE** angajat

( id\_angajat **DECIMAL(4,0)**

, nume **VARCHAR(25) NOT NULL**

, prenume **VARCHAR(25)**

, **CONSTRAINT** ang\_PK **PRIMARY KEY** (id\_angajat)

);

## 5.10 Crearea tabelului TRANSPORT

**CREATE TABLE** transport

( nr\_masina **VARCHAR(7)**

, firma\_curierat **VARCHAR(25) NOT NULL**

, id\_sofer **DECIMAL(4,0) UNIQUE**

, **CONSTRAINT** trans\_PK **PRIMARY KEY** (nr\_masina)

, **CONSTRAINT** trans\_nr\_check **CHECK** (length(nr\_masina)=7)

);

## 5.11 Crearea tabelului MAGAZIN

**CREATE TABLE** magazin

( id\_magazin **DECIMAL(4,0)**

, nume **VARCHAR(25) NOT NULL**

, localitate **VARCHAR(25) NOT NULL**

, strada **VARCHAR(25) NOT NULL**

, numar **DECIMAL(4,0)**

, **CONSTRAINT** magaz\_PK **PRIMARY KEY** (id\_magazin));

## 5.12 Inserarea datelor în tabelul PRODUCATOR

**insert into** producator **values**(9638038, 'L`oreal Romania S.R.L', 'L`oreal', 'Bucuresti', 'Floreasca');

**insert** into producator **values**(36571040, 'Brand Essence S.R.L', 'Essence', 'Bucuresti', 'Opalului');

**insert into** producator **values**(211369, 'Cosmetic Plant Prodcom S.R.L', 'Cosmetic Plant', 'Cluj-Napoca', 'Traian Vuia');

**insert into** producator **values**(4240311, 'S.C. Sarantis Romania S.A', 'Elmiplant', 'Ploiesti', 'Rozelor');

**insert into** producator **values**(38659549, 'REVOLUTION MEDIA COMPANY SRL ', 'Revolution', 'Bucuresti', 'Academiei');

## 5.13 Inserarea datelor în tabelul APROVIZIONARE

**insert into** aprovizionare **values**(100000, 36571040, '2022-01-30');

**insert into** aprovizionare **values**(100001, 4240311, '2022-03-10');

**insert into** aprovizionare **values**(100002, 4240311, '2021-03-12');

**insert into** aprovizionare **values**(100003, 38659549, '2022-06-22');

**insert into** aprovizionare **values**(100004, 9638038, '2021-12-10');

## 5.14 Inserarea datelor în tabelul COSMETICE

**insert into** cosmetice **values**(10, 'Ruj', 'buze', 120, 35.5, '2023-01-10');

**insert into** cosmetice **values**(20, 'Fond de ten', 'fata', 60, 145.8, '2022-05-17');

**insert into** cosmetice **values**(30, 'Crema hidratanta', 'fata', 20, 90.0, '2023-10-21');

**insert into** cosmetice **values**(40, 'Mascara gene', 'ochi', 65, 55.5, '2023-12-30');

**insert into** cosmetice **values**(50, 'Lotiune de corp', 'corp', 18, 40, '2022-11-10');

## 5.15 Inserarea datelor în tabelul FACTURA\_A

**insert into** factura\_a **values**(1, 100001, 20, 50, 145.8);

**insert into** factura\_a **values**(2, 100003, 50, 10, 40);

**insert into** factura\_a **values**(3, 100003, 30, 50, 90.0);

## 5.16 Inserarea datelor în tabelul ANGAJAT

**insert into** angajat **values**(100,'Nemes', 'Tudor');

**insert into** angajat **values**(101,'Popa', 'Marius');

**insert into** angajat **values**(102,'Burgui', 'Roxana');

**insert into** angajat **values**(103,'Popescu', 'Elena');

**insert into** angajat **values**(104,'Marcu', 'Ioana');

## 5.17 Inserarea datelor în tabelul MAGAZIN

**insert into** magazin **values**(1000, 'Nicole', 'Giurgiu', 'Norilor', 3);

**insert into** magazin **values**(1001, 'Eifel', 'Pitesti', 'Aviatorilor', 102);

**insert into** magazin **values**(1002, 'Stavros', 'Galati', 'Lalelelor', null);

**insert into** magazin **values**(1003, 'Adonis', 'Bucuresti', 'Principala', 22);

**insert into** magazin **values**(1004, 'Dimar', 'Buzau', 'Polis', null);

## 5.18 Inserarea datelor în tabelul TRANSPORT

**insert into** transport **values**('IF03GLS','GLS', 80);

**insert into** transport **values**('IF10FAN','Fan Curier', 81);

**insert into** transport **values**('IF11FAN','Fan Curier', 82);

**insert into** transport **values**('IF05GLS','GLS', 83);

**insert into** transport **values**('IF01CAR','Cargus', 84);

## 5.19 Inserarea datelor în tabelul COMANDA

**insert into** comanda **values**(290500, '2021-10-21', '2021-10-23', 1001, 'IF10FAN', 100);

**insert into** comanda **values**(290501, '2021-12-30', '2022-01-03', 1003, 'IF05GLS', 103);

**insert into** comanda **values**(290502, '2021-06-20', '2021-07-01', 1003, 'IF03GLS', 101);

**insert into** comanda **values**(290503, '2020-11-12', '2020-11-15', 1000, 'IF01CAR', 102);

**insert into** comanda **values**(290504, '2022-01-05', null, 1004, 'IF01CAR', 100);

## 5.20 Inserarea datelor în tabelul FACTURA\_C

**insert into** factura\_c **values**(1, 290501, 10, 20, 35.5);

**insert into** factura\_c **values**(2, 290503, 40, 12, 55.5);

**insert into** factura\_c **values**(3, 290503, 20, 30, 145.8);

**insert into** factura\_c **values**(4, 290501, 20, 10, 145.8);