

# **Bază de date pentru gestionarea unui magazin online**

Maciuca Teodor Alexandru

Grupa 133

Anul 1

# Cuprins

1. Descrierea modelului real, a utilității și a regulilor de funcționare
2. Prezentarea constrângerilor impuse asupra modelului
3. Descrierea entităților
4. Descrierea relațiilor
5. Descrierea atributelor
6. Diagrama entitate-relație
7. Diagrama conceptuală
8. Enumerarea schemelor relaționale
9. Realizarea normalizărilor până la FN3
10. Crearea secvenței
11. Crearea tabelelor în SQL
12. Cereri SQL
13. Operații de actualizare și suprimare a datelor
14. Cereri SQL: outer-join și division
15. Optimizarea cererii

## **1.Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Aceasta este o baza de date pentru un magazin online. Modelul de date va gestiona informații legate de organizarea și funcționarea corectă a unui magazin online. Aceasta baza de date este utilă deoarece permite stocarea informațiilor despre angajați, produse, review-urile și furnizorii acestora, stocurile produselor în magazine, categorii de produse, magazine fizice, comenzi, plăți, clienți și informațiile acestora.

Magazinul permite tuturor clienților să comande unul sau mai multe produse și fiecare comandă are o plată unică. Fiecare comandă poate avea unul sau mai mulți curieri și o companie de curierat poate să aibă 1 sau mai multe comenzi. Fiecare produs are un unic furnizor și face parte dintr-o singură categorie. Produsele sunt unice și pot fi găsite și în magazinele fizice iar review-urile acestora sunt făcute de clienți. Fiecare client are un cont cu username unic și parola unică acesta având informații unice. La fiecare magazin lucrează cel puțin un angajat iar fiecare angajat are cel puțin un job.

## **2. Prezentarea constrângerilor impuse asupra modelului.**

- În fiecare magazin fizic există unul sau mai mulți angajați, prin urmare, trebuie ca la un magazin să lucreze măcar un angajat, iar un angajat trebuie să lucreze la un singur magazin.
- Fiecare angajat are un job și un job poate avea mai mulți angajați, prin urmare, toți angajații trebuie să aibă un job unic dar pot exista joburi pentru care nu s-a găsit angajat.
- Fiecare cont de client are informații, așadar fiecare cont al clienților are informații unice și informațiile unui client sunt atribuite unui cont unic.
- Un produs poate aparține unei singure categorii și o categorie poate avea mai multe produse. Un produs trebuie să aparțină unei categorii unice iar o categorie poate să nu aibă niciun produs.
- Un furnizor poate avea mai multe produse, așadar fiecare furnizor trebuie să aibă cel puțin un produs iar fiecare produs are un unic furnizor.
- Un cont poate avea mai multe comenzi, prin urmare trebuie ca o comandă să aparțină unui singur cont dar există și conturi care nu au comenzi.
- O comandă are o plată și fiecare plată are o comandă, prin urmare, fiecare comandă trebuie să aibă o plată unică și fiecare plată este efectuată la o singură comandă.

- O comanda poate avea mai multe produse, așadar un produs poate aparține cel puțin unei comenzi
- Comenzile sunt livrate de companii de curierat, așadar o comanda poate fi livrată de una sau mai multe companii de curierat iar companiile de curierat pot livra una sau mai multe comenzi.
- Produsele pot avea mai multe review-uri, prin urmare, un review trebuie sa fie la un unic produs dar un produs poate sa nu aibă nici un review.
- Conturile pot scrie mai multe review-uri, prin urmare, un review trebuie sa apartina unui cont unic dar un cont poate sa nu aibă nici un review.
- Magazinele pot avea mai multe produse, prin urmare, un produs poate aparține mai multor magazine sau la nici unul, iar un magazin trebuie sa contina cel puțin un produs.

### 3. Descrierea entitatilor

#### **Magazin**

Aici se memorează toate informațiile și datele despre magazinele fizice, acestea fiind identificate după cheia primară magazin\_id.

#### **Cont**

Aici se memorează datele de conectare pentru fiecare client, acestea fiind identificate după cheia primară cont\_id.

#### **Info\_cont**

Aici se memorează toți clienții și datele lor, ei fiind identificați după cheia primară client\_id.

#### **Angajat**

Aici se memorează toate informațiile despre angajați, ei fiind identificați după cheia primară angajat\_id.

#### **Produse**

Aici sunt memorate toate informațiile despre produsele din magazine, iar ele sunt identificate după cheia primară produs\_id.

### **Plata**

Aici se memorează detaliile plății făcută pentru fiecare comandă, ea fiind identificare prin cheia primara plata\_id.

### **Comanda**

Aici se memorează datele fiecărei comenzi, acestea fiind identificate prin cheia primara comanda\_id.

### **Curierat**

Aici sunt memorate informațiile despre companiile de curierat care livrează comenzi, acestea fiind identificate după cheia primară curierat\_id.

### **Furnizor**

Aici sunt memorate informațiile tuturor furnizorilor de produse, aceștia fiind identificați după cheia primara furnizor\_id.

### **Categorii**

Aici sunt memorate tipurile de categorii ale produselor, acestea fiind identificate după cheia primară categorii\_id.

### **Review\_produc**

Aici sunt memorate detaliile fiecărui review al unui produs făcut de un client, acestea fiind identificate după cheia primară compusă formată din produs\_id și cont\_id deoarece review-urile pentru un anumit produs sunt unice pentru fiecare client.

## **4. Descrierea relațiilor**

### **Comanda livrare Curierat**

Relația are cardinalitatea maximă many-many (n:m).

Relația are cardinalitatea minimă one-one (1:1).

### **Produse are furnizorul Furnizor**

Relația are cardinalitatea maximă many-many (n:1).

Relația are cardinalitatea minimă one-one (1:1).

### **Produse apartine categoriei Categorii**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (0:1).

### **Magazin are angajați Angajat**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (1:1).

### **Comanda este plătită prin Plata**

Relația are cardinalitatea maximă one-one (1:1).

Relația are cardinalitatea minimă one-one (1:1).

### **Angajat are jobul Joburi**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-one (0:1).

### **Produse are review-uri Review\_produs**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-zero (1:0).

### **Cont are comenzi Comanda**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-zero (1:0).

### **Cont are informațiile Info\_cont**

Relația are cardinalitatea maximă one-one (1:1).

Relația are cardinalitatea minimă one-one (1:1).

### **Cont a scris review Review\_produs**

Relația are cardinalitatea maximă many-one (n:1).

Relația are cardinalitatea minimă one-zero (0:1).

### **Magazin are stocuri Produse**

Relația are cardinalitatea maximă many-one (n:m).

Relația are cardinalitatea minimă one-one (1:0).

## Comanda obiecte livrare Produse

Relația are cardinalitatea maximă many-one (n:m).

Relația are cardinalitatea minimă one-one (1:0).

## 5. Descrierea atributelor

### 5.1 Descrierea atributelor entitaților

#### Magazin

- id\_magazin
  - un număr unic în funcție de care este identificat fiecare magazin
  - tip de date: number
  - nu poate fi null
- oras
  - orașul unde se afla magazinul
  - tip de date: varchar
- adresa
  - adresa unde se afla magazinul
  - tip de date varchar

#### Cont

- cont\_id
  - un număr unic în funcție de care se identifica fiecare cont
  - tip de date: number
  - nu poate fi null
- nume\_utilizator
  - numele de utilizator al clientului folosit la logare
  - tip de date: varchar
- parola
  - parola clientului folosită la logare
  - tip de date: varchar
- client\_id
  - id-ul informațiilor fiecărui client
  - tip de date: number
  - este cheie externă(referențiază tabela Info\_cont)

## Info\_cont

- client\_id
  - un număr unic în funcție de care se identifică informațiile fiecărui client
  - tip de date: number
  - nu poate fi null
- nume
  - numele clientului
  - tip de date: varchar
- prenume
  - prenumele clientului
  - tip de date: varchar
- data\_nasterii
  - data nașterii a clientului
  - tip de date: date
- adresa\_client
  - adresa clientului
  - tip de date: varchar
- email\_client
  - emailul clientului
  - tip de date: varchar
- număr\_telefon
  - numărul de telefon al clientului
  - tip de date: varchar

## Angajat

- angajat\_id
  - un număr unic în funcție de care se identifică informațiile fiecărui angajat
  - tip de date: number
  - nu poate fi null
- nume
  - numele angajatului
  - tip de date: varchar
- prenume
  - prenumele angajatului
  - tip de date: varchar



- data\_nasterii
  - data nașterii a angajatului
  - tip de date: date
- adresa
  - adresa angajatului
  - tip de date: varchar
- email
  - emailul angajatului
  - tip de date: varchar
- număr\_telefon
  - numărul de telefon al angajatului
  - tip de date: varchar
- job\_id
  - id-ul jobului fiecărui angajat
  - tip de date: number
  - este cheie externă(referențiază tabela Joburi)
- magazin\_id
  - id-ul magazinului unde lucrează angajatul
  - tip de date: number
  - este cheie externă(referențiază tabela Magazin)

## Produse

- produs\_id
  - un număr unic în funcție de care se identifică informațiile fiecărui produs
  - tip de date: number
  - nu poate fi null
- numele\_produsului
  - numele fiecărui produs
  - tip de date: varchar
- furnizor
  - id-ul furnizorului fiecărui produs
  - tip de date: number
  - este cheie externă(referențiază tabela Furnizor)
- categorii\_id
  - id-ul categoriilor în care aparține produsul
  - tip de date: number
  - este cheie externă(referențiază tabela Categori)

## Plata

- plata\_id
  - un număr unic în funcție de care se identifica informațiile plății
  - tip de date: number
  - nu poate fi null
- comanda\_id
  - id-ul comenzii asociată plății
  - tip de date: number
  - este cheie externă(referențiază tabela Comanda)
- suma
  - suma de plata pentru o comandă
  - tip de date: number
- data\_plătii
  - data în care a fost efectuată plata
  - tip de date: date

## Comanda

- comanda\_id
  - un număr unic în funcție de care se identifică informațiile comenzii
  - tip de date: number
  - nu poate fi null
- status
  - statusul comenzii
  - tip de date: varchar
- conti\_id
  - id-ul contului în care a efectuat comanda
  - tip de date: number
  - este cheie externă(referențiază tabela Cont)

## Furnizor

- furnizor\_id
  - un număr unic în funcție de care se identifică informațiile furnizorilor
  - tip de date: number
  - nu poate fi null
- nume\_furnizor

- numele furnizorului
  - tip de date: varchar
- adresa\_furnizor
  - adresa furnizorului
  - tip de date: varchar
- email\_furnizor
  - emailul furnizorului
  - tip de date: varchar
- telefon\_furnizor
  - numărul de telefon al furnizorului
  - tip de date: varchar

## Categorii

- categorii\_id
  - un număr unic în funcție de care se identifică categoriile
  - tip de date: number
  - nu poate fi null
- nume\_categorie
  - numele categoriei
  - tip de date: varchar

## Joburi

- job\_id
  - un număr unic în funcție de care se identifica tipul de job
  - tip de date: number
  - nu poate fi null
- nume\_job
  - numele jobului
  - tip de date: varchar
- min\_salariu
  - salariul minim al acestui job
  - tip de date: number
- max\_salariu
  - salariul maxim al acestui job
  - tip de date: number

## Review\_produc

- review\_produs\_id
  - un număr unic în funcție de care se identifica review-ul unui produs
  - tip de date: number
  - nu poate fi null
- titlu
  - titlul review-ului
  - tip de date: varchar
- rating
  - rating-ul produsului
  - tip de date: number
- descriere
  - descrierea produsului
  - tip de date: varchar
- data\_publicatie
  - data in care a fost publicat review-ul
  - tip de date: varchar
- produs\_id
  - id-ul produsului despre care este review ul
  - tip de date: number
  - este cheie externă(referențiază tabela Produse)
- cont\_id
  - id-ul contului care a publicat review-ul
  - tip de date: number
  - este cheie externa(referențiază tabela Cont)

## 5.2 Descrierea atributelor relațiilor

### Stocuri

- magazin\_id
  - id-ul magazinelor fiecărui stoc
  - tip de date: number
  - este cheie externă(referențiază tabela Magazin)
- produs\_id
  - id-ul produselor fiecărui stoc
  - tip de date: number
  - este cheie externă(referențiază tabela Produse)
- cantitate
  - cantitatea unui anumit produs într-un anumit magazin

- tip de date: number

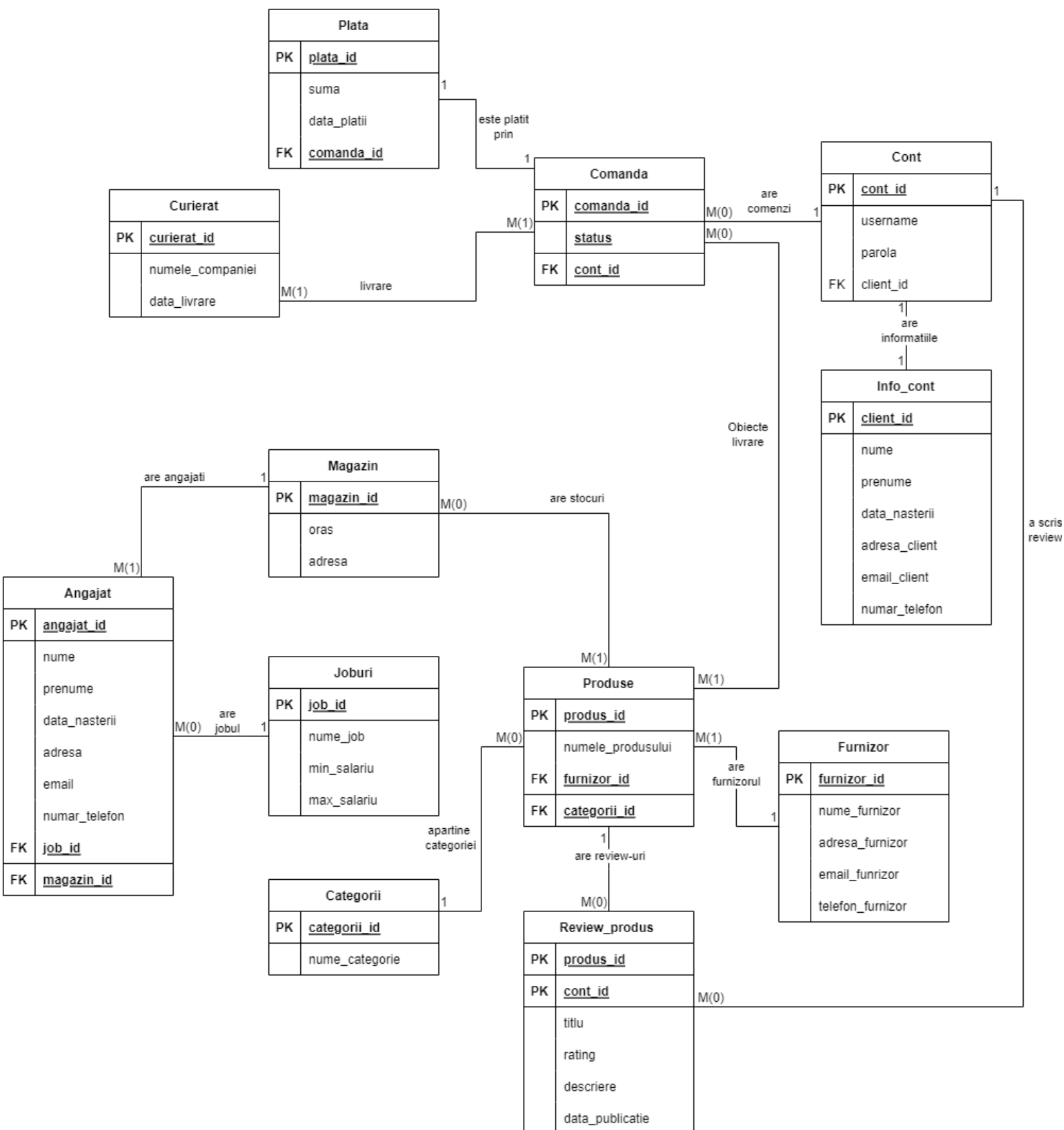
## **Livrare**

- comanda\_id
  - id-ul comenzii care trebuie livrate
  - tip de date: number
  - este cheie externă(referențiază tabela Comanda)
- curierat\_id
  - id-ul companiei de curierat care preia comanda
  - tip de date: number
  - este cheie externă(referențiază tabela Curierat)

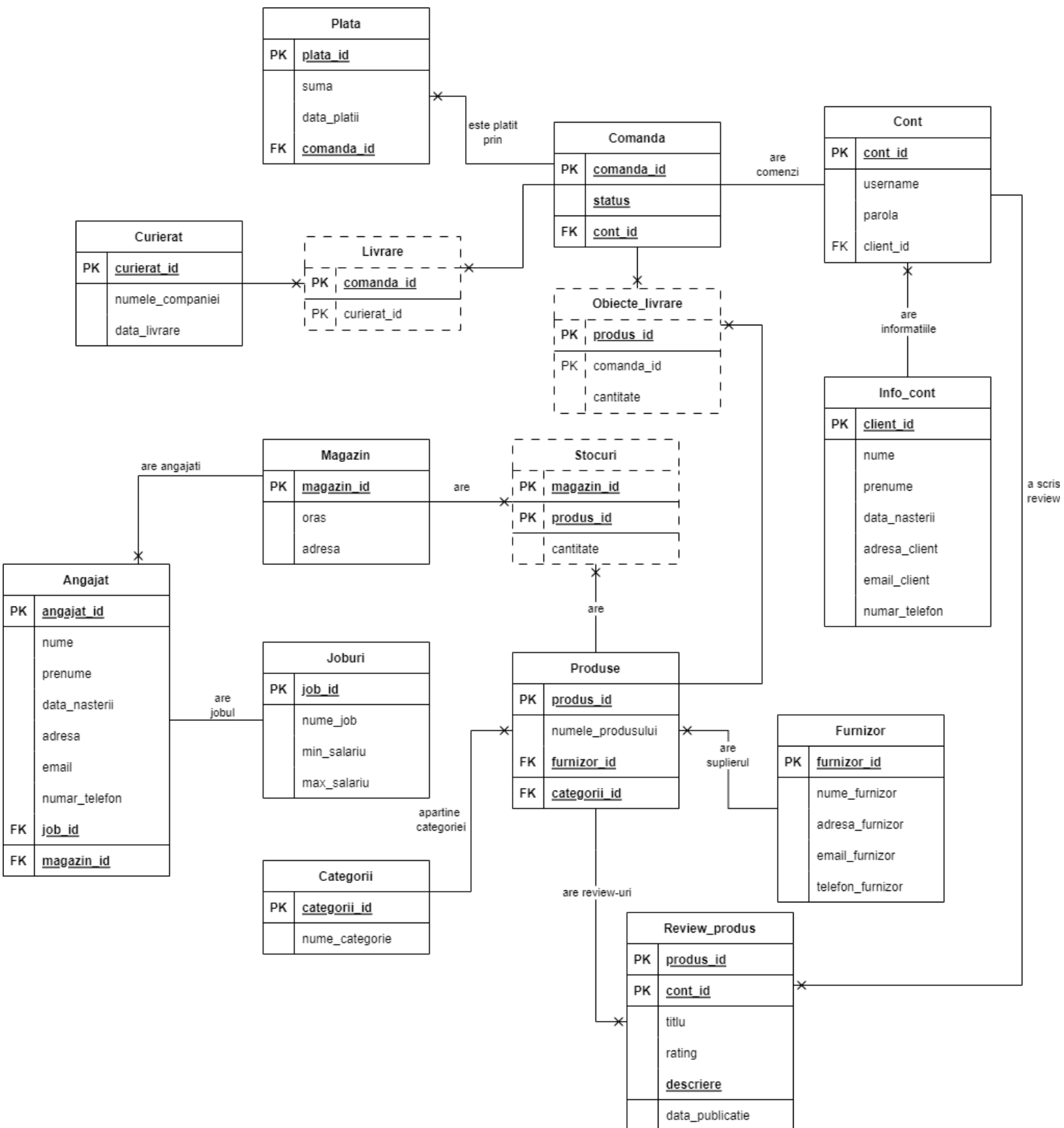
## **Obiecte\_livrare**

- produs\_id
  - id-ul produsului
  - tipul de date: number
  - este cheie externă(referențiază tabela Produse)
- comanda\_id
  - id-ul comenzii în care trebuie livrat produsul
  - tipul de date: number
  - este cheie externă(referențiază tabela Comanda)

## **6. Diagrama entitate-relație**



## 7. Diagrama conceptuala



## 8. Enumerarea schemelor relaționale

**Magazin** (magazin\_id#, oras, adresa)

**Angajat** (angajat\_id#, nume, prenume, data\_nasterii, adresa, email, numar\_telefon, job\_id, magazin\_id)

**Cont**(cont\_id#, username, parola, client\_id)

**Info\_cont** (client\_id#, nume, prenume, data\_nasterii, adresa\_client, email\_client, numar\_telefon )

**Stocuri** (magazin\_id#, produs\_id#, cantitate)

**Plata** (plata\_id#, suma, data\_plata, comanda\_id)

**Produse** (produs\_id#, numele\_produsului, categorii\_id, furnizor\_id)

**Comanda** (comanda\_id#, cont\_id, status)

**Categorii** (categorii\_id#, nume\_categorie)

**Review\_produs** (produs\_id#, cont\_id#, titlu, descriere, rating, data\_publicare,)

**Furnizor** (furnizor\_id#, nume\_furnizor, adresa\_furnizor, email\_furnizor, telefon\_furnizor)

**Jobs** (job\_id#, nume\_job, min\_salariu, max\_salariu)

**Curierat** (curierat\_id#, numele\_companiei, data\_livrare, comanda\_id)

**Obiecte\_livrare**(cantitate, comanda\_id#, produs\_id#)

**Livrare**(comanda\_id#, curierat\_id#)

## 9. Formele normale FN1, FN2, FN3

### FN1

Modelul proiectat se află în forma normală 1.

Exemplu de relație care nu se află în forma normală 1:

furnizor_id#	produs_id#
1	1, 8
2	2, 6, 10
3	3, 5, 9



4	4, 7
---	------

Aducere în FN1:

furnizor_id#	produs_id#
1	1
1	8
2	2
2	6
2	10
3	3
3	5
3	9
4	4
4	7

## FN2

Modelul proiectat se află în forma normală 2.

Exemplu de relație care nu se află în forma normală 2:

**Produse\_Categorii** (produs\_id#,categorii\_id#, numele\_produsului, furnizor\_id, numele\_categoriei)

produs_id#	categorii_id#	numele_produsului	furnizor_id	numele_categoriei
1	2	iPhone 12	2	Electronics
2	2	Samsung Galaxy S21	1	Electronics
3	1	Nike Air Max 90	4	Shoes
4	3	Harry Potter	2	Books

id\_produs -> numele\_produsului, furnizor\_id, categorii\_id

Acstea trebuie despărțite în:

**Produse** (produs\_id#, numele\_produsului, categorii\_id, furnizor\_id)

**Categorii** (categorii\_id#, nume\_categorie)

### Produse

produs_id#	numele_produsului	furnizor_id	categorii_id
1	iPhone 12	2	2
2	Samsung Galaxy S21	1	2
3	Nike Air Max 90	4	1
4	Harry Potter	2	3

### Categorii

categorii_id#	nume_categorie
1	Shoes
2	Electronics
3	Books

### FN3

Modelul proiectat se află în forma normală 3.

Exemplu de relație care nu se află în forma normală 3:

**Magazin** (magazin\_id#, tara, oras, adresa)

(oras depinde de tara care depinde de cheia primara magazin\_id)

magazin_id#	tara	oras	adresa
1	100	New York	123 Main St
2	150	London	456 High St
3	200	Paris	89 Rue de la Paix

**Magazin** (magazin\_id#, oras, adresa)

magazin_id#	oras	adresa
1	New York	123 Main St
2	London	456 High St
3	Paris	89 Rue de la Paix

## 10. Crearea unei secvențe pentru inserarea în tabele

create sequence seq\_client -- setarea denumirii secvenței  
start with 1 -- valoarea de la care pornește secvența  
increment by 1 -- valoarea care este incrementată la fiecare pas  
minvalue 0 -- valoarea minimă la care poate ajunge secvența  
maxvalue 100 -- valoarea maximă la care poate ajunge secvența  
nocycle; – când ajunge la maxvalue nu o mai ia de la capăt, ci va arunca o excepție

## 11. Inserarea în tabele

### Magazin

```
CREATE TABLE Magazin (
  magazin_id int NOT NULL,
  oras varchar(100),
  adresa varchar(100),
  PRIMARY KEY (magazin_id)
);
```

```
create sequence seq_magazin
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'New York', '123 Main St');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'London', '456 High St');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Paris', '789 Rue de la Paix');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Berlin', '10 Unter den Linden');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Sydney', '555 George St');
```

```

INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Toronto', '789 Yonge St');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Tokyo', '1-2 Shibuya St');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Madrid', '123 Gran Via');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Los Angeles', '789 Sunset Blvd');
INSERT INTO Magazin(magazin_id, oras, adresa) VALUES (seq_magazin.nextval, 'Toronto', '123 Main St');

```

MAGAZIN_ID	ORAS	ADRESA
1	New York	123 Main St
2	London	456 High St
3	Paris	789 Rue de la Paix
4	Berlin	10 Unter den Linden
5	Sydney	555 George St
6	Toronto	789 Yonge St
7	Tokyo	1-2 Shibuya St
8	Madrid	123 Gran Via
9	Los Angeles	789 Sunset Blvd
10	Toronto	123 Main St

## Angajat

```

CREATE TABLE Angajat (
  angajat_id int NOT NULL,
  nume varchar2(50) NOT NULL,
  prenume varchar2(50) NOT NULL,
  data_nasterii date,
  adresa varchar(100),
  email varchar2(100),
  numar_telefon varchar2(10),
  magazin_id int,
  job_id int,
  PRIMARY KEY (angajat_id),
  FOREIGN KEY (magazin_id) REFERENCES Magazin(magazin_id),
  FOREIGN KEY (job_id) REFERENCES Jobs(job_id)
);

```

```
create sequence seq_angajat
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'John', 'Doe', DATE '1990-05-15', '123 Main St', 'john.doe@example.com',
'1234567890', 1, 1);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Jane', 'Smith', DATE '1988-12-10', '456 High St', 'jane.smith@example.com',
'9876543210', 2, 2);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Michael', 'Johnson', DATE '1995-07-20', '789 Rue de la Paix',
'michael.johnson@example.com', '4561237890', 3, 2);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Emily', 'Johnson', DATE '1992-09-28', '10 Unter den Linden',
'emily.johnson@example.com', '1239874560', 4, 1);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'David', 'Brown', DATE '1991-03-17', '555 George St',
'david.brown@example.com', '9876543219', 5, 2);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Sarah', 'Wilson', DATE '1994-11-05', '789 Yonge St', 'sarah.wilson@example.com',
'4561237895', 6, 1);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Daniel', 'Miller', DATE '1989-07-12', '1-2 Shibuya St',
'daniel.miller@example.com', '7896541230', 7, 2);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Olivia', 'Anderson', DATE '1993-02-09', '123 Gran Via',
'olivia.anderson@example.com', '4567891230', 8, 1);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Mihai', 'Mirel', DATE '1989-07-12', '1-2 Shibuya St', 'mihai.mirel@example.com',
'7896541230', 9, 2);
INSERT INTO Angajat (angajat_id, nume, prenume, data_nasterii, adresa, email, numar_telefon, magazin_id,
job_id)
VALUES (seq_angajat.nextval, 'Bodea', 'Ionut', DATE '1993-02-09', '123 Gran Via', 'bodea.ionut@example.com',
'4567891230', 10, 1);
```

ANGAJAT_ID	NUME	PRENUME	DATA_NASTERII	ADRESA	EMAIL	NUMAR_TELEFON	MAGAZIN_ID	JOB_ID
1	John	Doe	15-MAY-90	123 Main St	john.doe@example.com	1234567890	1	1
2	Jane	Smith	10-DEC-88	456 High St	jane.smith@example.com	9876543210	2	2
3	Michael	Johnson	20-JUL-95	789 Rue de la Paix	michael.johnson@example.com	4561237890	3	2
4	Emily	Johnson	28-SEP-92	10 Unter den Linden	emily.johnson@example.com	1239874560	4	1
5	David	Brown	17-MAR-91	555 George St	david.brown@example.com	9876543219	5	2
6	Sarah	Wilson	05-NOV-94	789 Yonge St	sarah.wilson@example.com	4561237895	6	1
7	Daniel	Miller	12-JUL-89	1-2 Shibuya St	daniel.miller@example.com	7896541230	7	2
8	Olivia	Anderson	09-FEB-93	123 Gran Via	olivia.anderson@example.com	4567891230	8	1
9	Mihai	Mirel	12-JUL-89	1-2 Shibuya St	mihai.mirel@example.com	7896541230	9	2
10	Bodea	Ionut	09-FEB-93	123 Gran Via	bodea.ionut@example.com	4567891230	10	1

## Cont

```
CREATE TABLE Cont(
  cont_id int NOT NULL,
  username varchar(20) NOT NULL,
  parola varchar(30) NOT NULL,
  client_id int,
  PRIMARY KEY (cont_id),
  FOREIGN KEY (client_id) REFERENCES Info_cont(client_id)
);
```

```
create sequence seq_cont
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'johnsmith', 'p@ssw0rd1', 1);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'janedoe', 'securepass', 2);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'mikejones', 'abc123!', 3);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'emilywilson', 'pass1234', 4);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'davidbrown', 'secretword', 5);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'sarahmiller', 'mysecretpass', 6);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'robertthomas', 'passw0rd!', 7);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'laurawalker', '12345678', 8);
```

```

INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'alexandercruz', 'secure1!', 9);
INSERT INTO Cont (cont_id, username, parola, client_id)
VALUES (seq_cont.nextval, 'mollyallen', 'mypass123', 10);

```

CONT_ID	USERNAME	PAROLA	CLIENT_ID
1	johnsmith	p@ssw0rd1	1
2	janedoe	securepass	2
3	mikejones	abc123!	3
4	emilywilson	pass1234	4
5	davidbrown	secretword	5
6	sarahmiller	mysecretpass	6
7	robertthomas	passw0rd!	7
8	laurawalker	12345678	8
9	alexandercruz	secure1!	9
10	mollyallen	mypass123	10

## Info\_cont

```

CREATE TABLE Info_cont (
  client_id int NOT NULL,
  nume varchar2(50),
  prenume varchar2(50),
  data_nasterii date,
  adresa_client varchar(100),
  email_client varchar2(100),
  numar_telefon varchar2(10),
  PRIMARY KEY (client_id)
);

```

```

create sequence seq_client
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;

```

```

INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'John', 'Doe', DATE '1990-05-15', '123 Main St', 'john.doe@example.com',
'1234567890');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Jane', 'Smith', DATE '1988-12-10', '456 High St', 'jane.smith@example.com',
'9876543210');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Michael', 'Johnson', DATE '1995-07-20', '789 Rue de la Paix',
'michael.johnson@example.com', '4561237890');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Emily', 'Johnson', DATE '1992-09-28', '10 Unter den Linden',
'emily.johnson@example.com', '1239874560');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'David', 'Brown', DATE '1991-03-17', '555 George St', 'david.brown@example.com',
'9876543219');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Sarah', 'Wilson', DATE '1994-11-05', '789 Yonge St', 'sarah.wilson@example.com',
'4561237895');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Daniel', 'Miller', DATE '1989-07-12', '1-2 Shibuya St', 'daniel.miller@example.com',
'7896541230');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Olivia', 'Anderson', DATE '1993-02-09', '123 Gran Via',
'olivia.anderson@example.com', '4567891230');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Sophia', 'Garcia', DATE '1994-09-17', '789 Oak St', 'sophia.garcia@example.com',
'1234567890');
INSERT INTO Info_cont (client_id, nume, prenume, data_nasterii, adresa_client, email_client, numar_telefon)
VALUES (seq_client.nextval, 'Liam', 'Martinez', DATE '1993-07-23', '456 Maple Ave',
'liam.martinez@example.com', '9876543210');

```

CLIENT_ID	NUME	PRENUME	DATA_NASTERII	ADRESA_CLIENT	EMAIL_CLIENT	NUMAR_TELEFON
1	John	Doe	15-MAY-90	123 Main St	john.doe@example.com	1234567890
2	Jane	Smith	10-DEC-88	456 High St	jane.smith@example.com	9876543210
3	Michael	Johnson	20-JUL-95	789 Rue de la Paix	michael.johnson@example.com	4561237890
4	Emily	Johnson	28-SEP-92	10 Unter den Linden	emily.johnson@example.com	1239874560
5	David	Brown	17-MAR-91	555 George St	david.brown@example.com	9876543219
6	Sarah	Wilson	05-NOV-94	789 Yonge St	sarah.wilson@example.com	4561237895
7	Daniel	Miller	12-JUL-89	1-2 Shibuya St	daniel.miller@example.com	7896541230
8	Olivia	Anderson	09-FEB-93	123 Gran Via	olivia.anderson@example.com	4567891230
9	Sophia	Garcia	17-SEP-94	789 Oak St	sophia.garcia@example.com	1234567890
10	Liam	Martinez	23-JUL-93	456 Maple Ave	liam.martinez@example.com	9876543210



```

CREATE TABLE Stocuri (
    magazin_id int NOT NULL,
    produs_id int NOT NULL,
    cantitate int,
    PRIMARY KEY (magazin_id, produs_id),
    FOREIGN KEY (produs_id) REFERENCES Produse(produs_id),
    FOREIGN KEY (magazin_id) REFERENCES Magazin(magazin_id)
);

```

```

INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (1, 1, 50);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (6, 2, 50);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (8, 1,10);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (2, 2, 120);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (3, 1, 320);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (3, 2, 80);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (7, 1, 90);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (4, 2, 10);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (5, 1, 70);
INSERT INTO Stocuri (magazin_id, produs_id, cantitate) VALUES (9, 2, 130);

```

MAGAZIN_ID	PRODUS_ID	CANTITATE
1	1	50
6	2	50
8	1	75
2	2	120
3	1	60
3	2	80
7	1	90
4	2	110
5	1	70
9	2	130

## Plata

```

CREATE TABLE Plata (
    plata_id int NOT NULL,
    comanda_id int NOT NULL,
    suma int,
    data_plata date,
    PRIMARY KEY (plata_id),
    FOREIGN KEY (comanda_id) REFERENCES Comanda( comanda_id)

```

```
);
```

```
create sequence seq_plata  
start with 1  
increment by 1  
minvalue 0  
maxvalue 100  
nocycle;
```

```
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 1, 100, DATE  
'2021-06-01');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 2, 150, DATE  
'2020-06-02');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 3, 200, DATE  
'2022-06-03');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 4, 75, DATE  
'2023-06-04');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 5, 120, DATE  
'2021-06-05');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 6, 90, DATE  
'2022-06-06');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 7, 180, DATE  
'2023-06-07');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 8, 110, DATE  
'2023-06-08');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 9, 140, DATE  
'2021-06-09');  
INSERT INTO Plata (plata_id, comanda_id, suma, data_plata) VALUES (seq_plata.nextval, 10, 200, DATE  
'2023-06-10');
```

PLATA_ID	COMANDA_ID	SUMA	DATA_PLATA
1	1	100	01-JUN-21
2	2	150	02-JUN-20
3	3	200	03-JUN-22
4	4	75	04-JUN-23
5	5	120	05-JUN-21
6	6	90	06-JUN-22
7	7	180	07-JUN-23
8	8	110	08-JUN-23
9	9	140	09-JUN-21
10	10	200	10-JUN-23

## Produse

```
CREATE TABLE Produse (
  produs_id int NOT NULL,
  numele_produsului varchar(50),
  categorii_id int,
  furnizor_id int,
  PRIMARY KEY (produs_id),
  FOREIGN KEY (furnizor_id) REFERENCES Furnizor(furnizor_id),
  FOREIGN KEY (categorii_id) REFERENCES Categorii(categorii_id)
);
```

```
create sequence seq_produs
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'iPhone 12', 1, 1);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Samsung Galaxy S21', 1, 2);
```

```

INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Sony PlayStation 5', 1, 3);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Nike Air Max 90', 4, 4);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Canon EOS R6', 5, 5);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Apple MacBook Pro', 6, 6);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Dell XPS 13', 4, 7);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Adidas Superstar', 2,8);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'Sony WH-1000XM4', 9, 9);
INSERT INTO Produse (produs_id, numele_produsului, categorii_id, furnizor_id) VALUES (seq_produs.nextval,
'LG OLED CX Series', 3, 10);

```

1	iPhone 12	1	1
2	Samsung Galaxy S21	1	2
3	Sony PlayStation 5	1	3
4	Nike Air Max 90	4	4
5	Canon EOS R6	5	5
6	Apple MacBook Pro	6	6
7	Dell XPS 13	4	7
8	Adidas Superstar	2	8
9	Sony WH-1000XM4	9	9
10	LG OLED CX Series	3	10

## Comanda

```

CREATE TABLE Comanda (
  comanda_id int NOT NULL,
  cont_id int,
  status varchar(15)

```

```

PRIMARY KEY (comanda_id),
FOREIGN KEY (cont_id) REFERENCES Cont(cont_id)
);

```

```

create sequence seq_comanda
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;

```

```

INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 1, 'In procesare');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 2, 'In procesare');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 3, 'In procesare');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 4, 'Finalizata');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 5, 'Finalizata');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 5, 'Finalizata');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 2, 'In procesare');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 3, 'Finalizata');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 2, 'Finalizata');
INSERT INTO Comanda (comanda_id, cont_id,status) VALUES (seq_comanda.nextval, 5, 'In procesare');

```

COMANDA_ID	CONT_ID	STATUS
1	1	In procesare
2	2	In procesare
3	3	In procesare
4	4	Finalizata
5	5	Finalizata
6	5	Finalizata
7	2	In procesare
8	3	Finalizata
9	2	Finalizata
10	5	In procesare

## Obiecte\_livrare

```
CREATE TABLE Obiecte_livrare(  
    cantitate int NOT NULL,  
    comanda_id int,  
    produs_id int,  
    PRIMARY KEY (produs_id,comanda_id),  
    FOREIGN KEY (comanda_id) REFERENCES Comanda(comanda_id),  
    FOREIGN KEY (produs_id) REFERENCES Produse(produs_id)  
);
```

```
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (5, 1, 1);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (3, 1, 2);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (2, 2, 3);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (1, 3, 4);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (4, 4, 5);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (2, 4, 6);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (3, 5, 7);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (1, 6, 8);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (2, 7, 9);  
INSERT INTO Obiecte_livrare (cantitate, comanda_id, produs_id)  
VALUES (3, 7, 10);
```

CANTITATE	COMANDA_ID	PRODUS_ID
5	1	1
3	1	2
2	2	3
1	3	4
4	4	5
2	4	6
3	5	7
1	6	8
2	7	9
3	7	10

## Categorii

```
CREATE TABLE Categorii (
  categorii_id int NOT NULL,
  nume_categorie varchar(50),
  PRIMARY KEY (categorii_id)
);
```

```
create sequence seq_categorie
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Electronics');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Clothing');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Home Decor');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Shoes');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Appliances');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Furniture');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Books');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Beauty');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Sports');
INSERT INTO Categorii (categorii_id, nume_categorie) VALUES (seq_categorie.nextval, 'Jewelry');
```

CATEGORII_ID	NUME_CATEGORIE
1	Electronics
2	Clothing
3	Home Decor
4	Shoes
5	Appliances
6	Furniture
7	Books
8	Beauty
9	Sports
10	Jewelry

## Review\_produs

```
CREATE TABLE Review_produs (
  produs_id int NOT NULL,
  cont_id int NOT NULL,
  titlu varchar(50),
  descriere varchar(200),
  rating int,
  data_publicare date,
  PRIMARY KEY (produs_id, cont_id),
  FOREIGN KEY (produs_id) REFERENCES Produse(produs_id),
  FOREIGN KEY (cont_id) REFERENCES Cont(cont_id)
);
```

```
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (1, 1, 'Great Phone', 'I love this phone!', 5, DATE '2023-06-01');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (2, 2, 'Excellent Quality', 'The product exceeded my expectations.', 4, DATE '2022-06-02');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (3, 3, 'Beautiful Home Decor', 'Adds a touch of elegance to my home.', 5, DATE '2021-06-03');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (4, 4, 'Comfortable Shoes', 'Perfect fit and very comfortable.', 4, DATE '2021-06-04');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
```



```
VALUES (5, 7, 'Highly Recommended', 'Great performance and features.', 5, DATE '2022-06-05');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (6, 8, 'Impressive Laptop', 'Fast and reliable for my work.', 4, DATE '2023-06-06');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (7, 2, 'Sleek Design', 'Looks stylish and performs well.', 4, DATE '2023-06-07');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (8, 9, 'Classic Sneakers', 'Iconic design and comfortable to wear.', 5, DATE '2021-06-08');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (9, 4, 'Great Sound Quality', 'Noise-canceling feature works amazingly.', 4, DATE '2022-06-09');
INSERT INTO Review_produs (produs_id, cont_id, titlu, descriere, rating, data_publicare)
VALUES (10, 5, 'Impressive Display', 'Vivid colors and excellent picture quality.', 5, DATE '2021-06-10');
```

PRODUS_ID	CONT_ID	TITLU	DESCRIERE	RATING	DATA_PUBLICARE
1	1	Great Phone	I love this phone!	5	01-JUN-23
2	2	Excellent Quality	The product exceeded my expectations.	4	02-JUN-22
3	3	Beautiful Home Decor	Adds a touch of elegance to my home.	5	03-JUN-21
4	4	Comfortable Shoes	Perfect fit and very comfortable.	4	04-JUN-21
5	7	Highly Recommended	Great performance and features.	5	05-JUN-22
6	8	Impressive Laptop	Fast and reliable for my work.	4	06-JUN-23
7	2	Sleek Design	Looks stylish and performs well.	4	07-JUN-23
8	9	Classic Sneakers	Iconic design and comfortable to wear.	5	08-JUN-21
9	4	Great Sound Quality	Noise-canceling feature works amazingly.	4	09-JUN-22
10	5	Impressive Display	Vivid colors and excellent picture quality.	5	10-JUN-21

## Furnizor

```
CREATE TABLE Furnizor (
  furnizor_id int NOT NULL,
  nume_furnizor varchar(100),
  adresa_furnizor varchar(100),
  email_furnizor varchar(100),
  telefon_furnizor varchar(10),
  PRIMARY KEY (furnizor_id)
);
```

```
create sequence seq_furnizor
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;
```

```
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
```

```
VALUES (seq_furnizor.nextval, 'ABC Electronics', '123 Main Street', 'abc@example.com', '1234567890');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'XYZ Fashion', '456 Elm Avenue', 'xyz@example.com', '9876543210');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'Global Home Decor', '789 Oak Lane', 'global@example.com', '5678901234');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'ShoeMart', '321 Maple Road', 'shoemart@example.com', '8901234567');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'Appliance World', '654 Pine Street', 'appliance@example.com', '3456789012');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'Furniture Direct', '987 Cedar Avenue', 'furniture@example.com', '6789012345');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'BookWorm Publishers', '234 Birch Lane', 'bookworm@example.com',
'9012345678');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'Beauty Essentials', '567 Walnut Street', 'beauty@example.com', '4321098765');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'Sports Unlimited', '890 Oak Avenue', 'sports@example.com', '7654321098');
INSERT INTO Furnizor (furnizor_id, nume_furnizor, adresa_furnizor, email_furnizor, telefon_furnizor)
VALUES (seq_furnizor.nextval, 'JewelCraft', '123 Elm Lane', 'jewelry@example.com', '2109876543');
```

FURNIZOR_ID	NUME_FURNIZOR	ADRESA_FURNIZOR	EMAIL_FURNIZOR	TELEFON_FURNIZOR
1	ABC Electronics	123 Main Street	abc@example.com	1234567890
2	XYZ Fashion	456 Elm Avenue	xyz@example.com	9876543210
3	Global Home Decor	789 Oak Lane	global@example.com	5678901234
4	ShoeMart	321 Maple Road	shoemart@example.com	8901234567
5	Appliance World	654 Pine Street	appliance@example.com	3456789012
6	Furniture Direct	987 Cedar Avenue	furniture@example.com	6789012345
7	BookWorm Publishers	234 Birch Lane	bookworm@example.com	9012345678
8	Beauty Essentials	567 Walnut Street	beauty@example.com	4321098765
9	Sports Unlimited	890 Oak Avenue	sports@example.com	7654321098
10	JewelCraft	123 Elm Lane	jewelry@example.com	2109876543

## Jobs

```
CREATE TABLE Jobs (
  job_id int NOT NULL,
  nume_job varchar(100),
  min_salariu int,
  max_salariu int,
  PRIMARY KEY (job_id)
);
```

```

create sequence seq_jobs
start with 1
increment by 1
minvalue 0
maxvalue 100
nocycle;

```

```

INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Software Engineer', 50000, 100000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Sales Manager', 60000, 120000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Marketing Specialist', 45000, 90000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Financial Analyst', 55000, 110000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'HR Coordinator', 40000, 80000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Operations Supervisor', 48000, 96000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Customer Service Representative', 35000, 70000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Graphic Designer', 42000, 84000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Quality Assurance Engineer', 52000, 104000);
INSERT INTO Jobs (job_id, nume_job, min_salariu, max_salariu)
VALUES (seq_jobs.nextval, 'Administrative Assistant', 38000, 76000);

```

JOB_ID	NUME_JOB	MIN_SALARIU	MAX_SALARIU
1	Software Engineer	50000	100000
2	Sales Manager	60000	120000
3	Marketing Specialist	45000	90000
4	Financial Analyst	55000	110000
5	HR Coordinator	40000	80000
6	Operations Supervisor	48000	96000
7	Customer Service Representative	35000	70000
8	Graphic Designer	42000	84000
9	Quality Assurance Engineer	52000	104000
10	Administrative Assistant	38000	76000

## Curierat

```
CREATE TABLE Curierat (  
  curierat_id int NOT NULL,  
  numele_companiei varchar(100),  
  data_livrare date,  
  PRIMARY KEY (curierat_id)  
);
```

```
create sequence seq_curierat  
start with 1  
increment by 1  
minvalue 0  
maxvalue 100  
nocycle;
```

```
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'ABC Shipping', DATE '2023-06-15');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'XYZ Logistics', DATE '2023-06-16');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Global Couriers', DATE '2023-06-17');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Express Shipping', DATE '2023-06-18', );  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'FastTrack Logistics', DATE '2023-06-19');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Swift Couriers', DATE '2023-06-20');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'QuickShip Express', DATE '2023-06-21');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Rapid Dispatch', DATE '2023-06-22' );  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Speedy Delivery', DATE '2023-06-23');  
INSERT INTO Curierat (curierat_id, numele_companiei, data_livrare)  
VALUES (seq_curierat.nextval, 'Prime Movers', DATE '2023-06-24');
```

CURIERAT_ID	NUMELE_COMPANIEI	DATA_LIVRARE
1	ABC Shipping	15-JUN-23
2	XYZ Logistics	16-JUN-23
3	Global Couriers	17-JUN-23
4	FastTrack Logistics	19-JUN-23
5	Swift Couriers	20-JUN-23
6	QuickShip Express	21-JUN-23
7	Rapid Dispatch	22-JUN-23
8	Speedy Delivery	23-JUN-23
9	Prime Movers	24-JUN-23

## Livrare

```
CREATE TABLE Livrare(
  curierat_id int NOT NULL,
  comanda_id int NOT NULL,
  PRIMARY KEY (curierat_id,comanda_id),
  FOREIGN KEY (curierat_id) REFERENCES Curierat(curierat_id),
  FOREIGN KEY (comanda_id) REFERENCES Comanda(comanda_id)
);
```

```
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (1, 1);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (2, 2);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (3, 3);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (4, 4);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (5, 5);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (6, 6);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (7, 7);
INSERT INTO Livrare (curierat_id, comanda_id)
VALUES (8, 8);
INSERT INTO Livrare (curierat_id, comanda_id)
```

```
VALUES (9, 9);  
INSERT INTO Livrare (curierat_id, comanda_id)  
VALUES (10, 10);
```

CURIERAT_ID	COMANDA_ID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

## 12. Cereri SQL

### Cererea 1

**Subcereri sincronizate în care intervin cel puțin 3 tabele**

Sa se afișeze toate informațiile despre produsele care au fost într o comanda cu plata peste 100.

```
SELECT *  
FROM Produse  
WHERE produs_id IN (  
    SELECT produs_id  
    FROM Obiecte_livrare  
    WHERE comanda_id IN (  
        SELECT comanda_id  
        FROM Comanda  
        WHERE comanda_id in(  
            SELECT comanda_id
```

```

        FROM Plata
        WHERE suma > 100
    )
)
);

```

```

1 v SELECT *
2 FROM Produse
3 WHERE produs_id IN (
4     SELECT produs_id
5     FROM Obiecte_livrare
6     WHERE comanda_id IN (
7         SELECT comanda_id
8         FROM Comanda
9         WHERE comanda_id in(
10             SELECT comanda_id
11             FROM Plata
12             WHERE suma > 100
13         )
14     )
15 )

```

PRODUS_ID	NUMELE_PRODUSULUI	CATEGORII_ID	FURNIZOR_ID
3	Sony PlayStation 5	1	3
4	Nike Air Max 90	4	4
7	Dell XPS 13	4	4
9	Sony WH-1000XM4	9	3
10	LG OLED CX Series	3	2

## Cererea 2

### Ordonări și utilizarea funcțiilor NVL și DECODE

Sa se afișeze numele produsului, produs\_id, magazin\_id și cantitatea pentru toate produsele din toate magazinele și sa se afișeze dacă acestea sunt disponibile sau nu.

```

SELECT p.numele_produsului, p.produs_id, magazin_id, NVL(s.cantitate, 0) AS cantitate_stoc,
       DECODE(s.cantitate, NULL, 'Indisponibil', 'Disponibil') AS status_stoc
FROM Produse p
LEFT JOIN Stocuri s ON p.produs_id = s.produs_id
ORDER BY cantitate_stoc DESC;

```

```

58 v SELECT p.numele_produsului, p.produs_id, magazin_id, NVL(s.cantitate, 0) AS cantitate_stoc,
59      DECODE(s.cantitate, NULL, 'Indisponibil', 'Disponibil') AS status_stoc
70 FROM Produse p
71 LEFT JOIN Stocuri s ON p.produs_id = s.produs_id
72 ORDER BY cantitate_stoc DESC;

```

NUMELE_PRODUSULUI	PRODUS_ID	MAGAZIN_ID	CANTITATE_STOC	STATUS_STOC
iPhone 12	1	3	320	Disponibil
Samsung Galaxy S21	2	9	130	Disponibil
Samsung Galaxy S21	2	2	120	Disponibil
iPhone 12	1	7	90	Disponibil
Samsung Galaxy S21	2	3	80	Disponibil
iPhone 12	1	5	70	Disponibil
iPhone 12	1	1	50	Disponibil
Samsung Galaxy S21	2	6	50	Disponibil
Samsung Galaxy S21	2	4	10	Disponibil
iPhone 12	1	8	10	Disponibil
Dell XPS 13	7	-	0	Indisponibil
Nike Air Max 90	4	-	0	Indisponibil
Canon EOS R6	5	-	0	Indisponibil

### Cererea 3

**Subcereri nesincronizate în clauza FROM și grupări de date cu funcții de grup și filtrare la nivel de grupuri**

Sa se afișeze numele și prenumele, numărul de plăți și suma totală a plăților pentru clienții cu suma totală a plăților peste 300.



```

SELECT concat(concat(ic.ume, ' '), ic.prenume) as nume_client, COUNT(p.suma) AS
numar_plati, SUM(p.suma) suma_totala
FROM Cont c
LEFT JOIN Info_cont ic ON c.client_id = ic.client_id
LEFT JOIN Comanda co ON c.cont_id = co.cont_id
LEFT JOIN Plata p ON co.comanda_id = p.comanda_id
GROUP BY ic.ume, ic.prenume
HAVING COUNT(p.suma) >= 2 AND SUM(p.suma) > 300;

```

```

51 v SELECT concat(concat(ic.ume, ' '), ic.prenume) as nume_client, COUNT(p.suma) AS numar_plati, SUM(p.suma) suma_totala
52 FROM Cont c
53 LEFT JOIN Info_cont ic ON c.client_id = ic.client_id
54 LEFT JOIN Comanda co ON c.cont_id = co.cont_id
55 LEFT JOIN Plata p ON co.comanda_id = p.comanda_id
56 GROUP BY ic.ume, ic.prenume
57 HAVING COUNT(p.suma) >= 2 AND SUM(p.suma) > 300;
58

```

NUME_CLIENT	NUMAR_PLATI	SUMA_TOTALA
Michael Johnson	2	310
David Brown	2	320
Jane Smith	2	330

## Cererea 4

**Utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE**

Pentru fiecare comanda sa se afiseze adresa clientului, inițialele acestuia, data în care a fost efectuată plata, vârsta clientului și statusul acesteia.

```

SELECT
SUBSTR(ic.adresa_client, 1, 10) AS adresa_client,
UPPER(SUBSTR(ic.ume, 1, 1) || '.' || SUBSTR(ic.prenume, 1, 1)) AS initiale_client,
CONCAT('Data plata: ', TO_CHAR(p.data_plata, 'DD-MON-YYYY')) AS data_plata_formatata,
CONCAT('Total plata: ', p.suma) AS total_plata,
CASE
WHEN co.status = 'In procesare' THEN 'Comanda in procesare'
WHEN co.status = 'Finalizata' THEN 'Comanda finalizata'
ELSE 'Status necunoscut'
END AS status_comanda
FROM
Cont c
JOIN Info_cont ic ON ic.client_id = c.client_id

```

JOIN Comanda co ON co.cont\_id = c.cont\_id  
 JOIN Plata p ON co.comanda\_id = p.comanda\_id;

```

2 v SELECT
3     SUBSTR(ic.adresa_client, 1, 10) AS adresa_client,
4     UPPER(SUBSTR(ic.num, 1, 1) || '.' || SUBSTR(ic.prenume, 1, 1)) AS initiale_client,
5     CONCAT('Data plata: ', TO_CHAR(p.data_plata, 'DD-MON-YYYY')) AS data_plata_formatata,
6     TRUNC(MONTHS_BETWEEN(SYSDATE, ic.data_nasterii) / 12) AS varsta_client,
7     CASE
8         WHEN co.status = 'In procesare' THEN 'Comanda in procesare'
9         WHEN co.status = 'Finalizata' THEN 'Comanda finalizata'
10        ELSE 'Status necunoscut'
11    END AS status_comanda
12 FROM
13     Cont c
14 JOIN Info_cont ic ON ic.client_id = c.client_id
15 JOIN Comanda co ON co.cont_id = c.cont_id
16 JOIN Plata p ON co.comanda_id = p.comanda_id;
  
```

ADRESA_CLIENT	INITIALE_CLIENT	DATA_PLATA_FORMATATA	VARSTA_CLIENT	STATUS_COMANDA
123 Main S	J.D	Data plata: 01-JUN-2021	33	Comanda in procesare
456 High S	J.S	Data plata: 02-JUN-2020	34	Comanda in procesare
789 Rue de	M.J	Data plata: 03-JUN-2022	27	Comanda in procesare
10 Unter d	E.J	Data plata: 04-JUN-2023	30	Comanda finalizata
555 George	D.B	Data plata: 05-JUN-2021	32	Comanda finalizata
555 George	D.B	Data plata: 06-JUN-2022	32	Comanda finalizata
456 High S	J.S	Data plata: 07-JUN-2023	34	Comanda in procesare
789 Rue de	M.J	Data plata: 08-JUN-2023	27	Comanda finalizata
456 High S	J.S	Data plata: 09-JUN-2021	34	Comanda finalizata

## Cererea 5

### Utilizarea blocului de cerere (clauza WITH):

Sa se afişeze numele produsului şi cantitatea pentru toate livrările efectuate.

```

WITH Produse_Livrare AS (
  SELECT ol.produs_id, SUM(ol.cantitate) AS Cantitate_Totala
  FROM Obiecte_livrare ol
  GROUP BY ol.produs_id
  )
  
```

```

)
SELECT pr.numele_produsului, pl.Cantitate_Totala
FROM Produse pr
JOIN Produse_Livrare pl ON pr.produs_id = pl.produs_id
WHERE pl.Cantitate_Totala > (SELECT AVG(Cantitate_Totala) FROM Produse_Livrare);

```

```

21 v WITH Produse_Livrare AS (
22     SELECT ol.produs_id, SUM(ol.cantitate) AS Cantitate_Totala
23     FROM Obiecte_livrare ol
24     GROUP BY ol.produs_id
25 )
26 SELECT pr.numele_produsului, pl.Cantitate_Totala
27 FROM Produse pr
28 JOIN Produse_Livrare pl ON pr.produs_id = pl.produs_id
29 WHERE pl.Cantitate_Totala > (SELECT AVG(Cantitate_Totala) FROM Produse_Livrare);
30
<

```

NUMELE_PRODUSULUI	CANTITATE_TOTALA
iPhone 12	5
Dell XPS 13	3
Samsung Galaxy S21	3
Canon EOS R6	4
LG OLED CX Series	3

## 13. Operații de actualizare și de ștergere a datelor

### Operația 1

Să se crească cu 1234 maximul salariilor care încep cu "S" și "G".

```

update jobs
set max_salariu = max_salariu + 100
where nume_job in (select nume_job
from jobs
where nume_job like 'S%' or nume_job like 'G%');

```

Tabela Jobs înainte:

JOB_ID	NUME_JOB	MIN_SALARIU	MAX_SALARIU
1	Software Engineer	50000	100100
2	Sales Manager	60000	120100
3	Marketing Specialist	45000	90000
4	Financial Analyst	55000	110000
5	HR Coordinator	40000	80000
6	Operations Supervisor	48000	96000
7	Customer Service Representative	35000	70000
8	Graphic Designer	42000	84100
9	Quality Assurance Engineer	52000	104000
10	Administrative Assistant	38000	76000

Tabela Jobs după update(3 rows updated):

```

22 v update jobs
23 set max_salariu = max_salariu + 1234
24 where nume_job in (select nume_job
25                     from jobs
26                     where nume_job like 'S%' or nume_job like 'G%');
27

```

JOB_ID	NUME_JOB	MIN_SALARIU	MAX_SALARIU
1	Software Engineer	50000	101334
2	Sales Manager	60000	121334
3	Marketing Specialist	45000	90000
4	Financial Analyst	55000	110000
5	HR Coordinator	40000	80000
6	Operations Supervisor	48000	96000
7	Customer Service Representative	35000	70000
8	Graphic Designer	42000	85334
9	Quality Assurance Engineer	52000	104000
10	Administrative Assistant	38000	76000

## Operația 2

Să se șteargă din tabela Furnizor furnizorii care au o adresă care începe cu numărul 1 sau 2 și numele cu 'B'.

```

delete from Furnizor
where adresa_furnizor in(select adresa_furnizor from Furnizor
                          where adresa_furnizor like '1%' or adresa_furnizor like '2%' or adresa_furnizor like '3%')
and nume_furnizor in (select nume_furnizor from Furnizor
                      where nume_furnizor like 'B%');

```

Tabela Furnizor înainte:

FURNIZOR_ID	NUME_FURNIZOR	ADRESA_FURNIZOR	EMAIL_FURNIZOR	TELEFON_FURNIZOR
1	ABC Electronics	123 Main Street	abc@example.com	1234567890
2	XYZ Fashion	456 Elm Avenue	xyz@example.com	9876543210
3	Global Home Decor	789 Oak Lane	global@example.com	5678901234
4	ShoeMart	321 Maple Road	shoemart@example.com	8901234567
5	Appliance World	654 Pine Street	appliance@example.com	3456789012
6	Furniture Direct	987 Cedar Avenue	furniture@example.com	6789012345
7	BookWorm Publishers	234 Birch Lane	bookworm@example.com	9012345678
8	Beauty Essentials	567 Walnut Street	beauty@example.com	4321098765
9	Sports Unlimited	890 Oak Avenue	sports@example.com	7654321098
10	JewelCraft	123 Elm Lane	jewelry@example.com	2109876543

Tabela Furnizor după update(1 row deleted):

```
18_v delete from Furnizor
19 where adresa_furnizor in(select adresa_furnizor from Furnizor
20 where adresa_furnizor like '1%' or adresa_furnizor like '2%' or adresa_furnizor like '3%')
21 and nume_furnizor in (select nume_furnizor from Furnizor
22 where nume_furnizor like '8%');
23
```

FURNIZOR_ID	NUME_FURNIZOR	ADRESA_FURNIZOR	EMAIL_FURNIZOR	TELEFON_FURNIZOR
1	ABC Electronics	123 Main Street	abc@example.com	1234567890
2	XYZ Fashion	456 Elm Avenue	xyz@example.com	9876543210
3	Global Home Decor	789 Oak Lane	global@example.com	5678901234
4	ShoeMart	321 Maple Road	shoemart@example.com	8901234567
5	Appliance World	654 Pine Street	appliance@example.com	3456789012
6	Furniture Direct	987 Cedar Avenue	furniture@example.com	6789012345
8	Beauty Essentials	567 Walnut Street	beauty@example.com	4321098765
9	Sports Unlimited	890 Oak Avenue	sports@example.com	7654321098
10	JewelCraft	123 Elm Lane	jewelry@example.com	2109876543

### Operația 3

Să se transforme adresa de email a angajatilor într-o adresa de muncă pentru angajații născuți înainte de anul 1992(se va înlocui 'example' cu 'storemail')

```
update Angajat
set email = replace(email,'@example','@storemail')
where angajat_id in (select angajat_id
                     from Angajat
                     where data_nasterii < '1-jan-1992');
```

Tabela Angajat înainte:

ANGAJAT_ID	NUME	PRENUME	DATA_NASTERII	ADRESA	EMAIL	NUMAR_TELEFON	MAGAZIN_ID	JOB_ID
1	John	Doe	15-MAY-90	123 Main St	john.doe@example.com	1234567890	1	1
2	Jane	Smith	10-DEC-88	456 High St	jane.smith@example.com	9876543210	2	2
3	Michael	Johnson	20-JUL-95	789 Rue de la Paix	michael.johnson@example.com	4561237890	1	2
4	Emily	Johnson	28-SEP-92	10 Unter den Linden	emily.johnson@example.com	1239874560	2	1
5	David	Brown	17-MAR-91	555 George St	david.brown@example.com	9876543219	1	2
6	Sarah	Wilson	05-NOV-94	789 Yonge St	sarah.wilson@example.com	4561237895	2	1
7	Daniel	Miller	12-JUL-89	1-2 Shibuya St	daniel.miller@example.com	7896541230	1	2
8	Olivia	Anderson	09-FEB-93	123 Gran Via	olivia.anderson@example.com	4567891230	2	1

Tabela Angajat după update(4 rows updated):

```
74 v update Angajat
75 set email = replace(email,'@example','@storemail')
76 where angajat_id in (select angajat_id
77 from Angajat
78 where data_nasterii < '1-jan-1992');
79
80
```

ANGAJAT_ID	NUME	PRENUME	DATA_NASTERII	ADRESA	EMAIL	NUMAR_TELEFON	MAGAZIN_ID	JOB_ID
1	John	Doe	15-MAY-90	123 Main St	john.doe@storemail.com	1234567890	1	1
2	Jane	Smith	10-DEC-88	456 High St	jane.smith@storemail.com	9876543210	2	2
3	Michael	Johnson	20-JUL-95	789 Rue de la Paix	michael.johnson@example.com	4561237890	1	2
4	Emily	Johnson	28-SEP-92	10 Unter den Linden	emily.johnson@example.com	1239874560	2	1
5	David	Brown	17-MAR-91	555 George St	david.brown@storemail.com	9876543219	1	2
6	Sarah	Wilson	05-NOV-94	789 Yonge St	sarah.wilson@example.com	4561237895	2	1
7	Daniel	Miller	12-JUL-89	1-2 Shibuya St	daniel.miller@storemail.com	7896541230	1	2
8	Olivia	Anderson	09-FEB-93	123 Gran Via	olivia.anderson@example.com	4567891230	2	1

## 14. CERERI SQL: OUTER-JOIN ȘI DIVISION

### OUTER-JOIN

Sa se afișeze numele produsului, furnizorul, categoria, și review-ul acestuia.

```
select Produse.numele_produsului, Furnizor.nume_furnizor, Categories.nume_categorie,
Review_produs.descriere
FROM Produse, Furnizor,Categories, Review_produs
where Produse.produs_id(+) = Review_produs.produs_id and
Categories.categorii_id(+) = Produse.categorii_id and
Furnizor.furnizor_id(+) = Produse.furnizor_id;
```





```

7 v SELECT DISTINCT comanda_id
8 FROM Livrare a
9 WHERE NOT EXISTS(SELECT 1
10                  FROM Curierat p
11                  WHERE numele_companiei like 'R%'
12                  AND NOT EXISTS
13                      (SELECT 'x' Comanda
14                       FROM Livrare b
15                       WHERE p.curierat_id = b.curierat_id
16                          AND b.comanda_id = a.comanda_id));

```

COMANDA_ID
7

## Analiza top-n

Să se selecteze numele, prenumele, id-ul angajatului, adresa de email și data nașterii ai primilor 3 angajați ordonați după data\_nasterii.

```

SELECT ROWNUM as RANK, nume, prenume, angajat_id, email, data_nasterii
FROM (SELECT nume, prenume, angajat_id, email, data_nasterii
      FROM Angajat
      ORDER BY data_nasterii)
WHERE ROWNUM <= 3;

```

```

35 v SELECT ROWNUM as RANK, nume, prenume, angajat_id, email, data_nasterii
36 FROM (SELECT nume, prenume, angajat_id, email, data_nasterii
37        FROM Angajat
38        ORDER BY data_nasterii)
39 WHERE ROWNUM <= 3;
40

```

RANK	NUME	PRENUME	ANGAJAT_ID	EMAIL	DATA_NASTERII
1	Jane	Smith	2	jane.smith@example.com	10-DEC-88
2	Daniel	Miller	7	daniel.miller@example.com	12-JUL-89
3	John	Doe	1	john.doe@example.com	15-MAY-90

## 15. Optimizarea cererii

Sa se obtina id-ul magazinului, numele oraşului în care se afla acesta, id-ul produsului şi cantitatea acestuia cantitate pentru magazinele care au în adresa 'St' şi au

**Magazin** (magazin\_id#, oras, adresa)

**Stocuri** (magazin\_id#, produs\_id#, cantitate)

R1 = PROJECT(MAGAZIN, id\_magazin, oras, adresa)

R2 = SELECT(R1, adresa like '%str%')

R3 = PROJECT(R2, id\_magazin, oras)

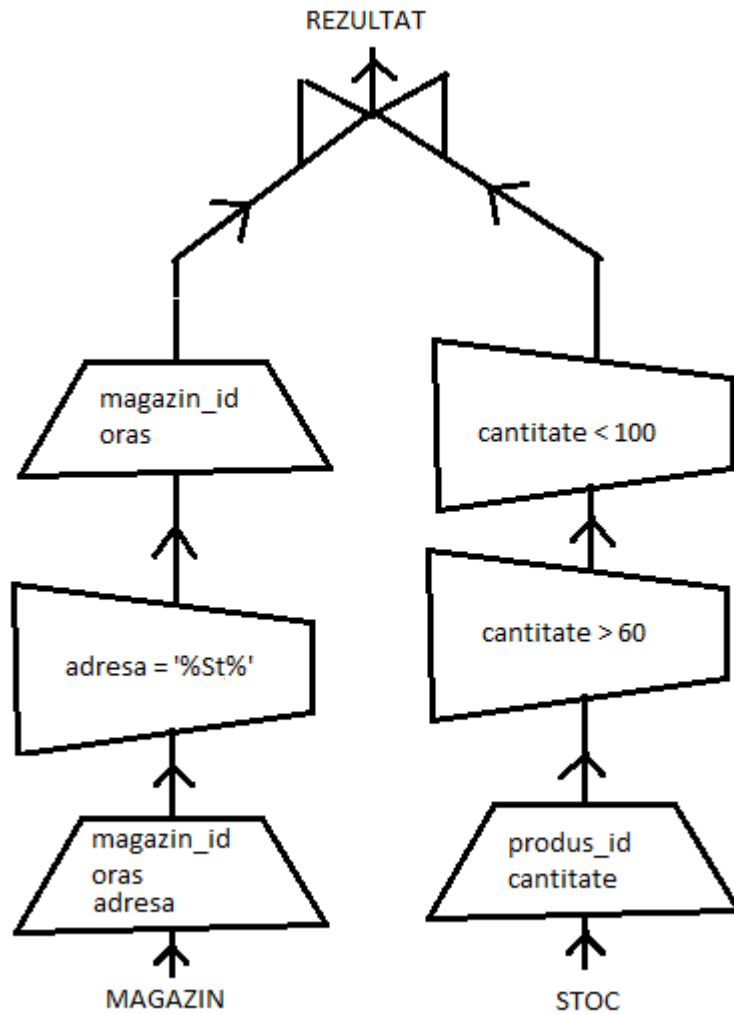
R4 = PROJECT(STOCURI, id\_produs, cantitate)

R5 = SELECT(R4, cantitate > 60)

R6 = SELECT(R5, cantitate < 100)

Rezultat = JOIN(R3, R6)

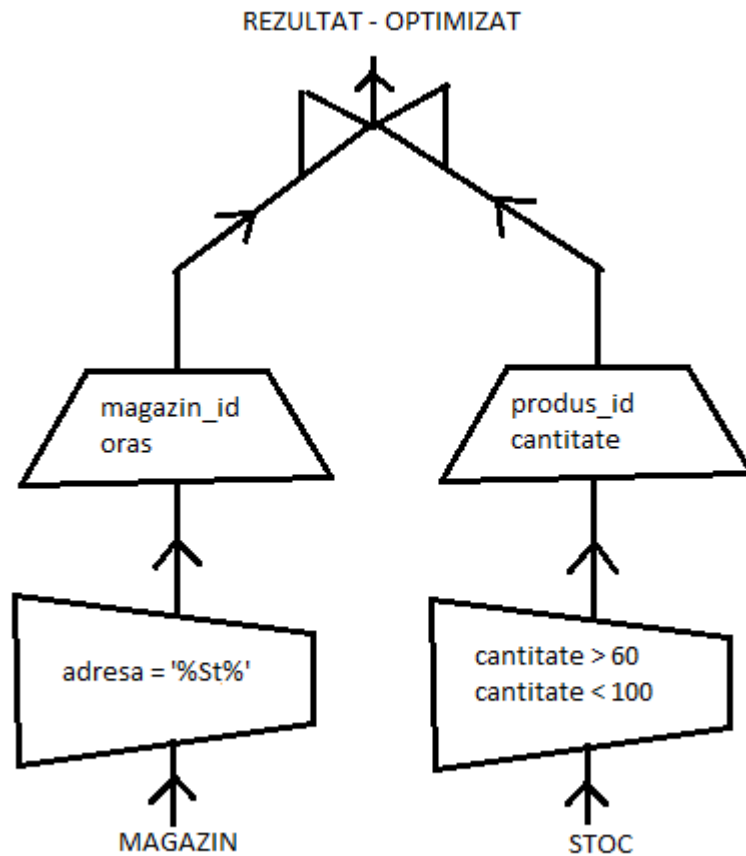
Arborele algebric iniţial:



Optimizarea cererii:

$$\begin{aligned}
 \text{Rezultat}_{(1)} &= \pi_{\text{magazin\_id}, \text{oras}} \left( \sigma_{\text{adresa} = \% \text{str} \%} \left( \pi_{\text{magazin\_id}, \text{oras}, \text{adresa}} (\text{MAGAZIN}) \right) \right) \bowtie \sigma_{\text{cantitate} < 100} \left( \sigma_{\text{cantitate} < 60} \left( \pi_{\text{produs\_id}, \text{cantitate}} (\text{STOC}) \right) \right) \\
 \text{Rezultat} &= \pi_{\text{magazin\_id}, \text{oras}} \left( \sigma_{\text{adresa} = \% \text{str} \%} \left( \pi_{\text{magazin\_id}, \text{oras}, \text{adresa}} (\text{MAGAZIN}) \right) \right) \bowtie \sigma_{\text{cantitate} < 100 \wedge \text{cantitate} > 60} \left( \pi_{\text{produs\_id}, \text{cantitate}} (\text{STOC}) \right) \\
 \text{Rezultat} &= \pi_{\text{magazin\_id}, \text{oras}, \text{adresa}} \left( \pi_{\text{magazin\_id}, \text{oras}, \text{adresa}} \left( \sigma_{\text{adresa} = \% \text{str} \%} (\text{MAGAZIN}) \right) \right) \bowtie \pi_{\text{produs\_id}, \text{cantitate}} \left( \sigma_{\text{cantitate} < 100 \wedge \text{cantitate} > 60} (\text{STOC}) \right) \\
 \text{Rezultat} &= \pi_{\text{magazin\_id}, \text{oras}} \left( \sigma_{\text{adresa} = \% \text{str} \%} (\text{MAGAZIN}) \right) \bowtie \pi_{\text{produs\_id}, \text{cantitate}} \left( \sigma_{\text{cantitate} < 100 \wedge \text{cantitate} > 60} (\text{STOC}) \right) = \text{Rezultat optim}
 \end{aligned}$$

Arborele algebric rezultat:



```

select m.magazin_id, m.oras, s.cantitate, s.produs_id
from stocuri s, magazin m
where m.adresa like '%St%' and m.magazin_id = s.magazin_id and s.cantitate > 60 and
s.cantitate < 100;

```

```

3 v select m.magazin_id, m.oras, s.cantitate, s.produs_id
4   from stocuri s, magazin m
5   where m.adresa like '%St%' and m.magazin_id = s.magazin_id and s.cantitate > 60 and s.cantitate < 100;

```

MAGAZIN_ID	ORAS	CANTITATE	PRODUS_ID
7	Tokyo	90	1
5	Sydney	70	1