**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 informaţii legate de organizarea şi funcţionarea corecta a unui magazin online. Aceasta baza de date este utilă deoarece permite stocarea informațiilor despre angajați, produse, review-urile și furnizorii acestora, stocurile produselor in magazine, categorii de produse, magazine fizice, comenzi, plăti, clienți și informațiile acestora.

Magazinul permite tuturor clienților să comande unul sau mai multe produse și fiecare comandă are o plată unică.Fiecare comanda 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 fac parte dintr-o singura categorie. Produsele sunt unicate 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 avand informații unicate. 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 exista unul sau mai mulți angajați, prin urmare, trebuie ca la un magazin sa lucreze măcar un angajat, iar un angajat trebuie sa 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 are atribuit un cont unic.
* Un produs poate aparține unei singure categorii și o categorie poate avea mai multe produse. Un produs trebuie sa aparține unei categorii unice iar o categorie poate să nu aibă niciun produs.
* Un furnizor poate avea mai multe produse, așadar fiecare furnizor trebuie sa 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 exista 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 putin 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 identificati dupa cheia primara client\_id.

**Angajat**

Aici se memoreaza toate informațiile despre angajați, ei fiind identificati dupa cheia primara 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 identificati dupa cheia primara furnizor\_id.

**Categorii**

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

**Review\_produs**

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 Categorii)

**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\_produs**

* 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

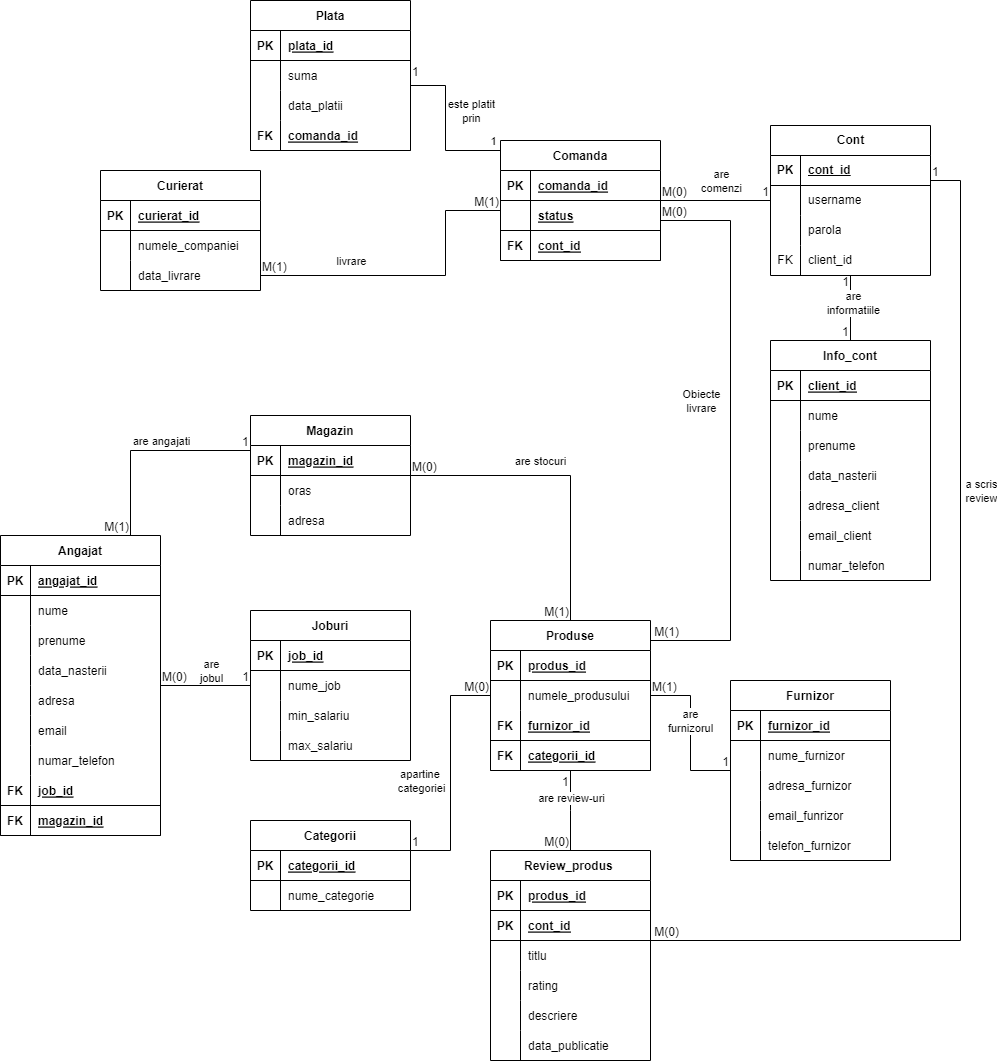
**Livrare**

* 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)

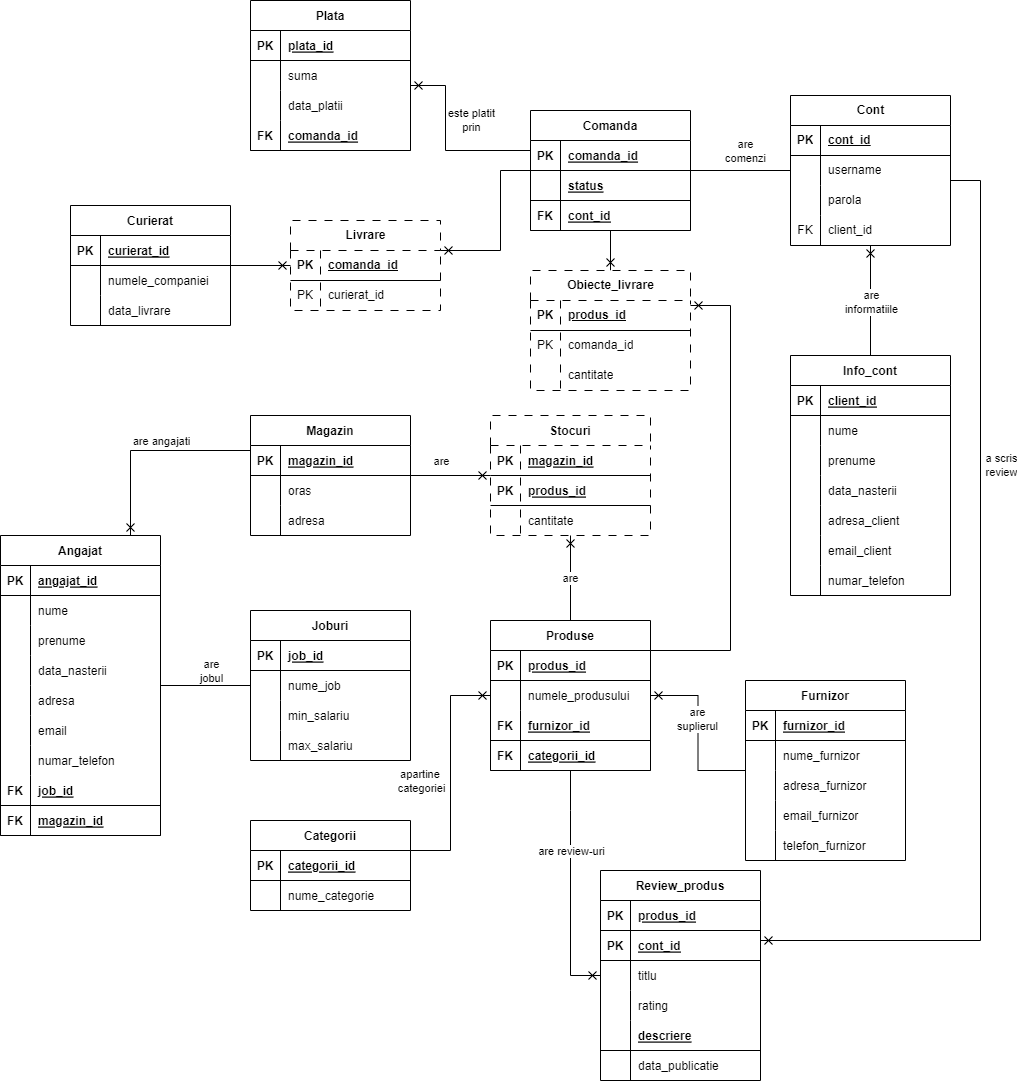
**Obiecte\_livrare**

* 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

Acestea 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 poatea 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');



**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)

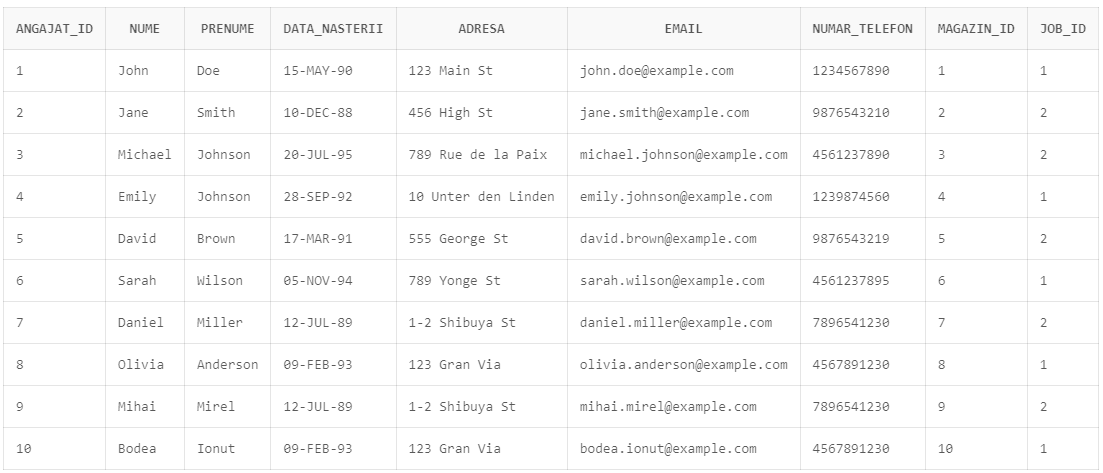
VALUES (seq\_angajat.nextval, 'Olivia', 'Anderson', DATE '1993-02-09', '123 Gran Via', 'olivia.anderson@example.com', '4567891230', 8, 1);

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);



**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)

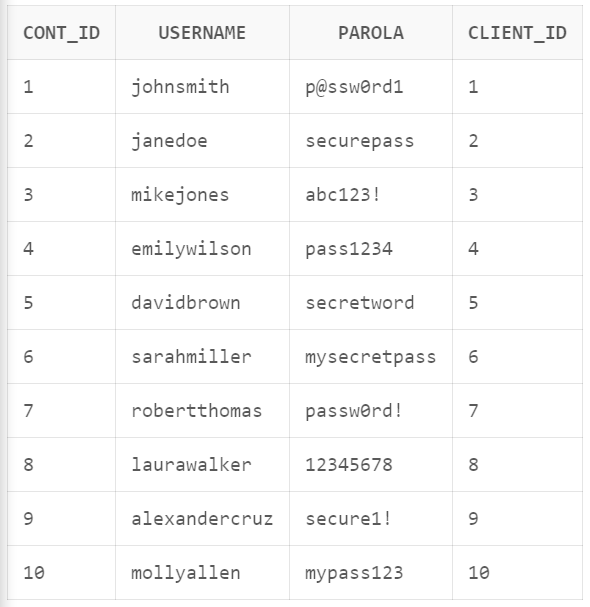
VALUES (seq\_cont.nextval, 'laurawalker', '12345678', 8);

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);



**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');



**Stocuri**

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);

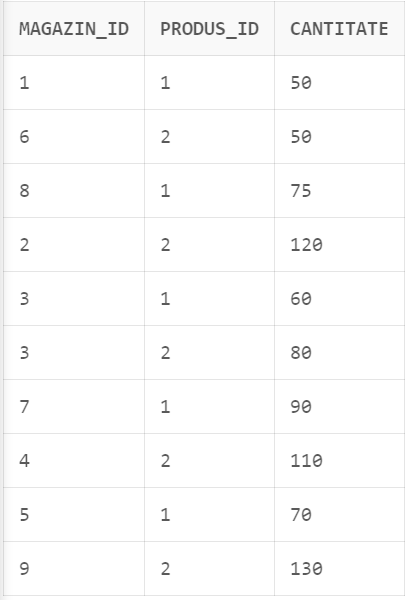
INSERT INTO Stocuri (magazin\_id, produs\_id, cantitate) VALUES (3, 2, 80);

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);



**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');

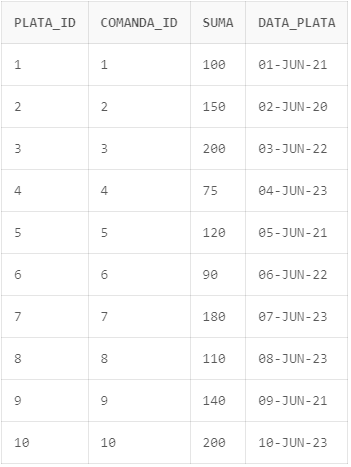
INSERT INTO Plata (plata\_id, comanda\_id, suma, data\_plata) VALUES (seq\_plata.nextval, 6, 90, DATE '2022-06-06');

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');



**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);



**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');

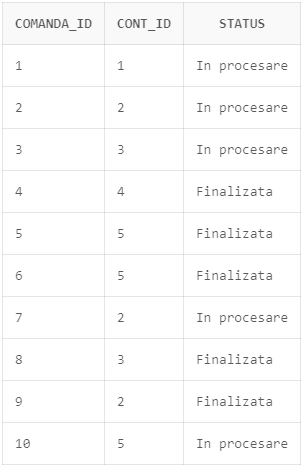
INSERT INTO Comanda (comanda\_id, cont\_id,status) VALUES (seq\_comanda.nextval, 5, 'Finalizata');

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');



**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)

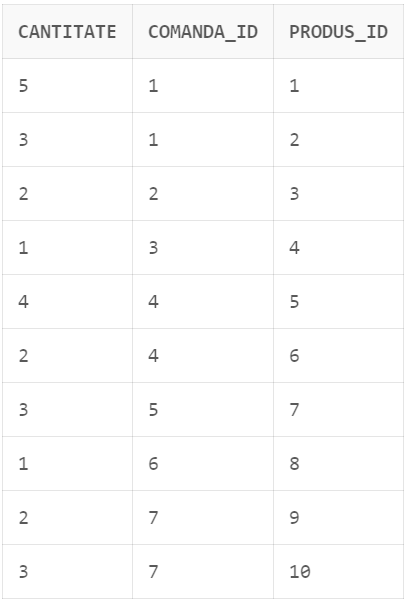
VALUES (1, 6, 8);

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);



**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');



**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)

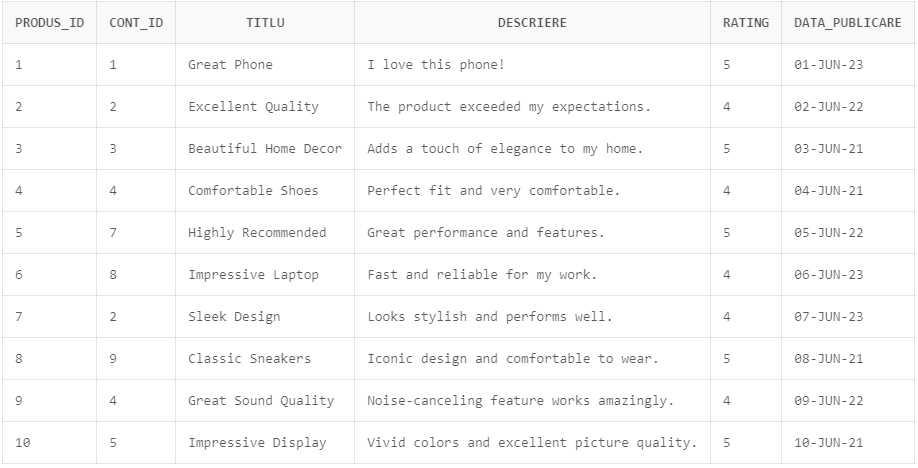
VALUES (8, 9, 'Classic Sneakers', 'Iconic design and comfortable to wear.', 5, DATE '2021-06-08');

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');



**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');



**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);



**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)

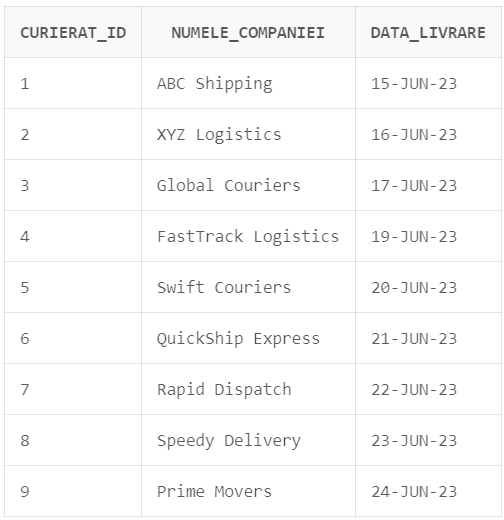
VALUES (seq\_curierat.nextval, 'Rapid Dispatch', DATE '2023-06-22' );

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');



**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)

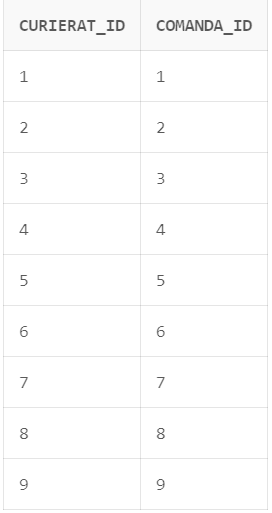
VALUES (8, 8);

INSERT INTO Livrare (curierat\_id, comanda\_id)

VALUES (9, 9);

INSERT INTO Livrare (curierat\_id, comanda\_id)

VALUES (10, 10);

****

**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

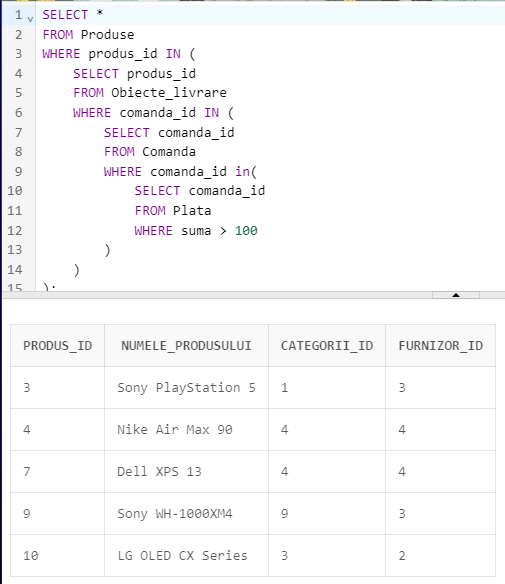
FROM Plata

WHERE suma > 100

)

)

);



**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 afiseze dacă acestea sunt disponibile sau nu.

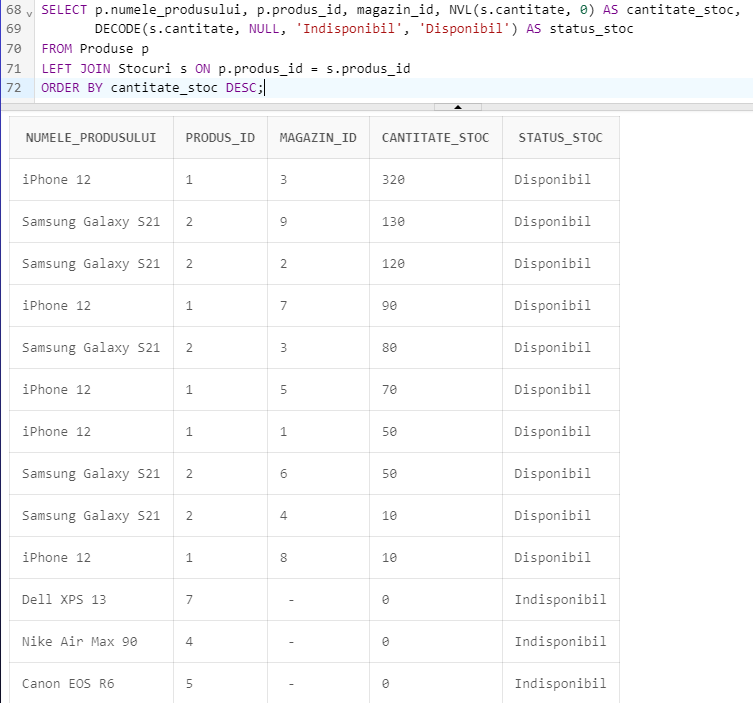
SELECT p.numele\_produsului, p.produs\_id, magazin\_id, NVL(s.cantitate, 0) AS cantitate\_stoc,

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;

****

**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.nume, ' '), ic.prenume) as nume\_client, COUNT(p.suma) AS numar\_plati, SUM(p.suma) suma\_totala

FROM Cont c

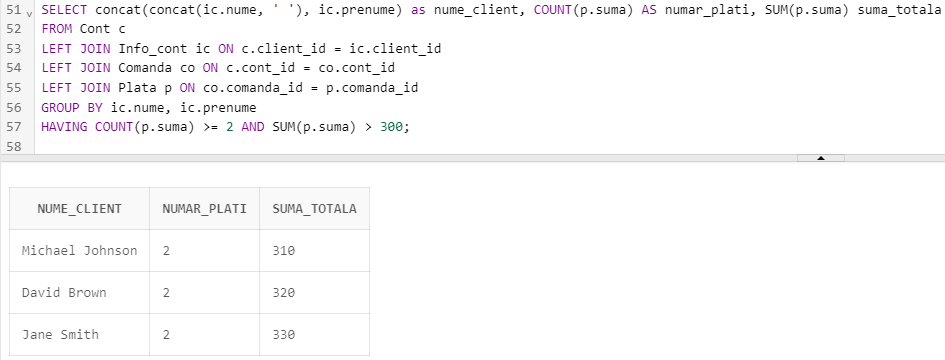
LEFT JOIN Info\_cont ic ON c.client\_id = ic.client\_id

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.nume, ic.prenume

HAVING COUNT(p.suma) >= 2 AND SUM(p.suma) > 300;

****

**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 efectuata plata, vârsta clientului și statusul acesteia.

SELECT

SUBSTR(ic.adresa\_client, 1, 10) AS adresa\_client,

UPPER(SUBSTR(ic.nume, 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

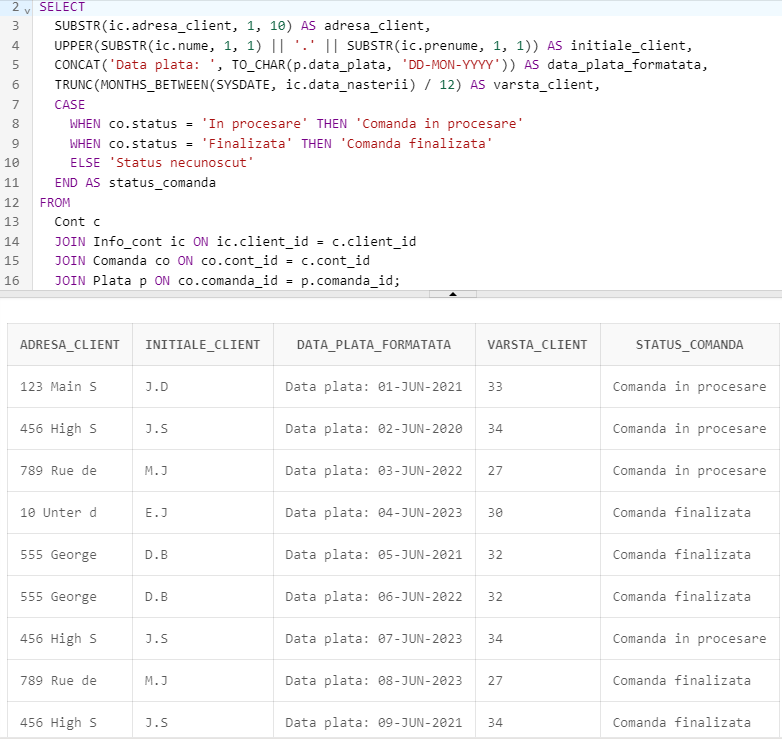
FROM

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;



**Cererea 5**

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

Sa se afișeze numele produsului și 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

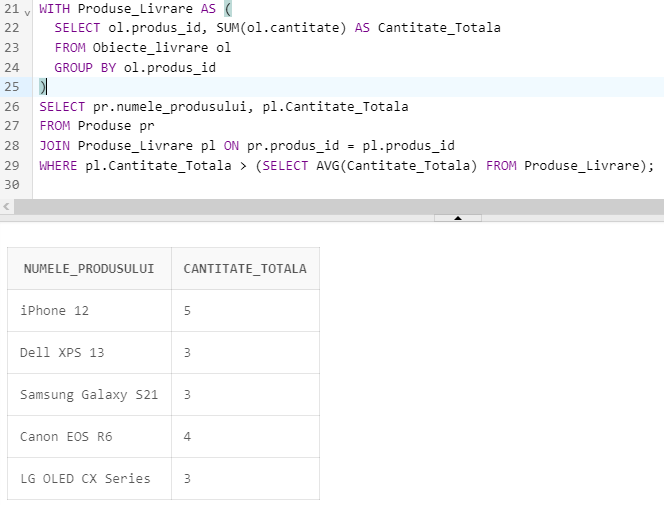
)

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);

****

**13. Operații de actualizare și de suprimare 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

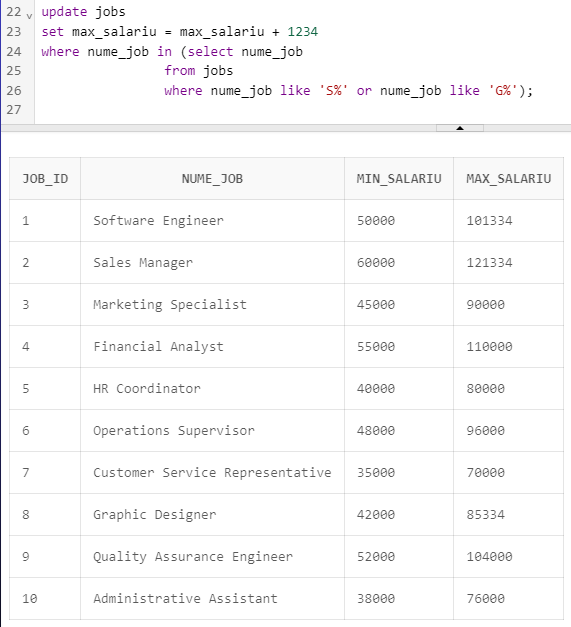
from jobs

where nume\_job like 'S%' or nume\_job like 'G%');

Tabela Jobs înainte:



Tabela Jobs după update(3 rows updated):



**Operația 2**

Să se ștearga din tabela Furnizor furnizorii care au o adresa care începe cu numărul 1 2 sau 3 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:



Tabela Furnizor după update(1 row deleted):



**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

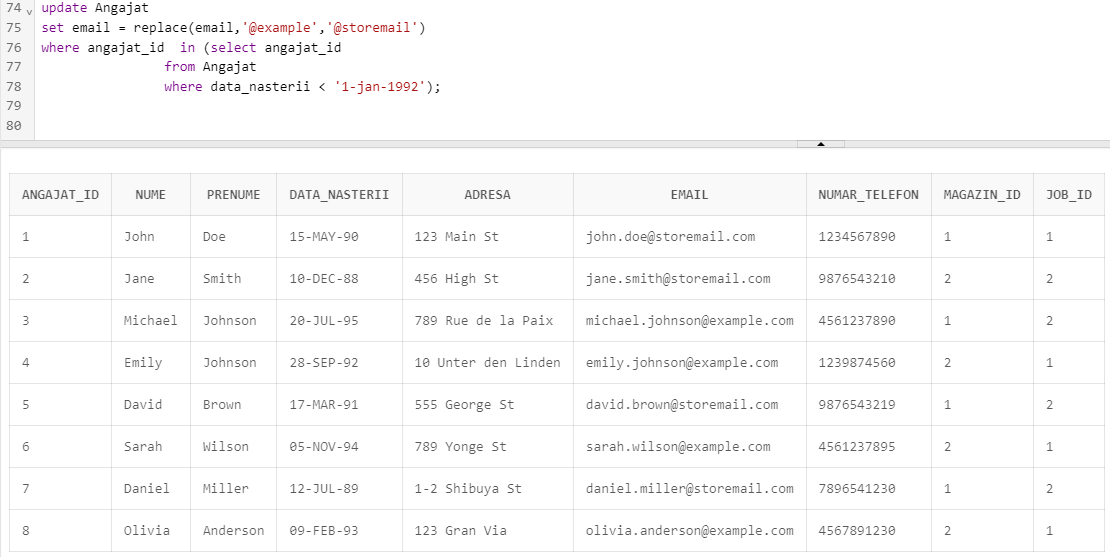
from Angajat

where data\_nasterii < '1-jan-1992');

Tabela Angajat înainte:



Tabela Angajat după update(4 rows updated):



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

**OUTER-JOIN**

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

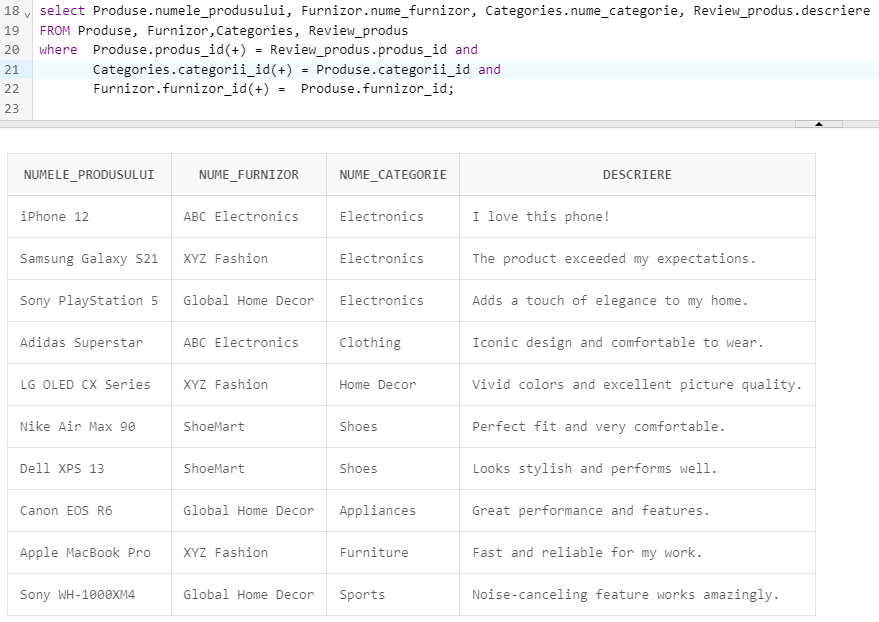
select Produse.numele\_produsului, Furnizor.nume\_furnizor, Categories.nume\_categorie, Review\_produs.descriere

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;



**DIVISION**

Sa se afișeze id-ul comenzilor care sunt livrate de companii care încep cu litera ‘R’.

SELECT DISTINCT comanda\_id

FROM Livrare a

WHERE NOT EXISTS(SELECT 1

FROM Curierat p

WHERE numele\_companiei like 'R%'

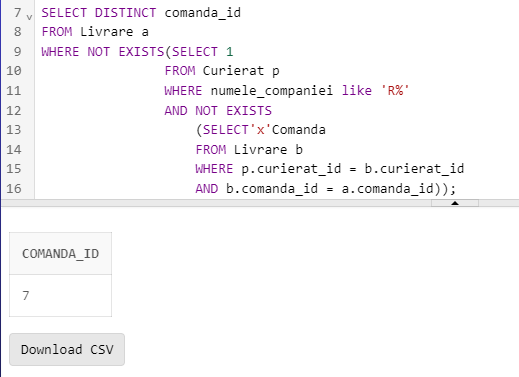
AND NOT EXISTS

(SELECT'x'Comanda

FROM Livrare b

WHERE p.curierat\_id = b.curierat\_id

AND b.comanda\_id = a.comanda\_id));



**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.

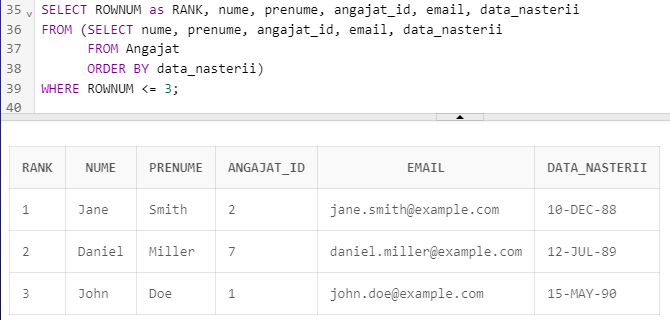
SELECT ROWNUM as RANK, nume, prenume, angajat\_id, email, data\_nasterii

FROM (SELECT nume, prenume, angajat\_id, email, data\_nasterii

FROM Angajat

ORDER BY data\_nasterii)

WHERE ROWNUM <= 3;



**15. Optimizarea cererii**

Sa se obtina id-ul magazinului, numele orașului în care se afla acesta, id-ul produsului și cantitatea acestuia cantitate pentru magazinele care au în adresa ‘St’ și 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)

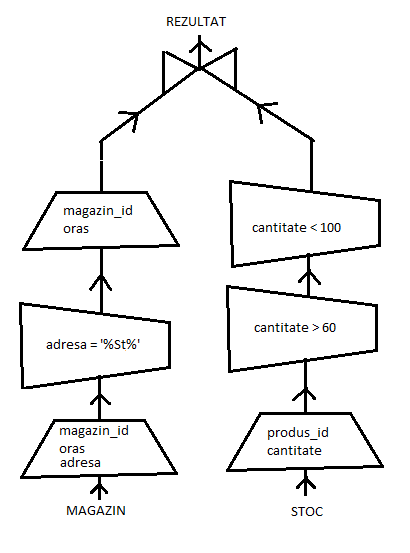
R4 = PROJECT(STOCURI, id\_produs, cantitate)

R5 = SELECT(R4, cantitate > 60)

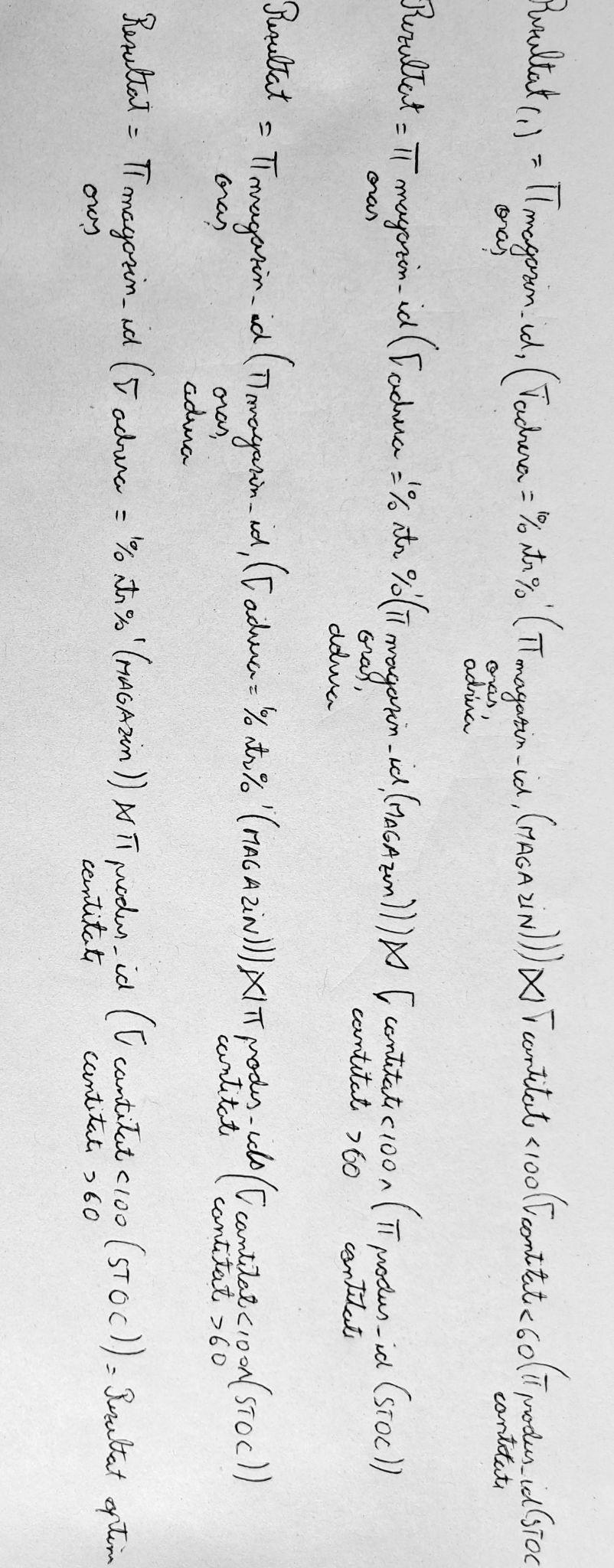
R6 = SELECT(R5, cantitate < 100)

Rezultat = JOIN(R3, R6)

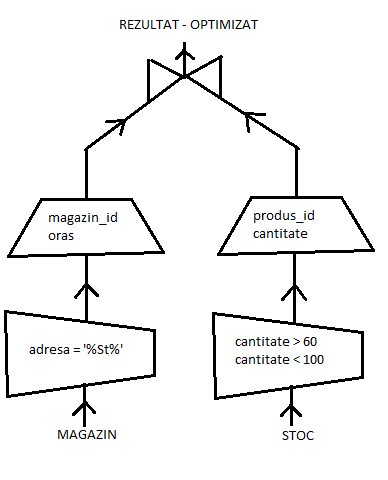
Arborele algebric inițial:



Optimizarea cererii:



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;

