

Proiect la Baze de Date
Managementul unui magazin de muzică

Proiect realizat de Robitu Riana Ioana
Grupa 144
Anul universitar 2024-2025

Cuprins

1.Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.....	2
2.Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.....	3
3.Descrierea entităților, incluzând precizarea cheii primare.....	4
4.Descrierea relațiilor, incluzând precizarea cardinalității acestora.....	5
5.Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.....	7
6.Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.....	11
7.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.....	11
8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.....	12
9.Realizarea normalizării până la forma normală 3 (FN1-FN3).....	13
10+11.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).....	15
12.Formulați în limbaj natural și implementați 5 cereri SQL complexe.....	25
13.Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri..	27
14.Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.....	29
15.Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.....	31
16.Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.....	34
17.Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării, justificând necesitatea acesteia.....	36
18.Exemplificarea isolation levels prin exemple de tranzacții care se execută în paralel în condiții de concurență, evidențiind efectele diferitelor niveluri de izolare asupra concurenței și integrității datelor.....	37

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

Această bază de date este concepută pentru a gestiona procesul de comercializare a produselor muzicale (albume, viniluri, instrumente muzicale), într-un sistem centralizat care implică clienți, comenzi, facturi și livrări. Este ideală pentru un magazin online de muzică sau o platformă de distribuție muzicală care colaborează cu case de discuri.

Aceasta este structurată în câteva părți, cu entități relevante pentru partea administrativă, cum ar fi Client, Comandă, Factură și Curier, și entități ajutătoare pentru organizarea produselor, precum Record Label.

Această bază de date vine în ajutorul managerilor magazinului de muzică, deoarece facilitează accesul la informații legate de toți clienții prin interogări simple (am menționat câteva exemple mai jos). În plus, baza de date este utilă și pentru clienți în procesul de căutare al produselor, putând fi filtrate cu ușurință tot datorită interogărilor precizate anterior.

Fiecare utilizator se poate înregistra o singură dată pentru a nu exista instanțe redundante în baza de date, iar acesta va introduce informații relevante, precum nume și date de contact (e-mail, telefon), dar și data de naștere, opțional, pentru diverse beneficii.

Fiecărei comenzi îi corespunde o factură în care se menționează prețul și metoda de plată preferată de fiecare client la momentul respectiv, iar în tabelul său va fi asociat câte un curier.

Prețul unei comenzi este calculat automat în funcție de cantitatea comandată și prețul fiecărui produs, și va fi preluat în factură.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

- Un client poate plasa mai multe comenzi;
- Fiecare comandă are propria ei factură, iar aceasta este unică;
- Un curier poate livra multiple comenzi;
- Un produs poate fi ori album, ori vinyl, ori instrument (numele se pot repeta, întrucât un album se poate vinde și drept vinyl, însă va fi prezentat în baza de date cu un id diferit și va fi organizat în subtabelul corespunzător)
- Fiecare album sau vinyl este asociat unei case de discuri (record label);
- O comandă poate conține mai multe albume, viniluri sau instrumente;
- Nucleul unei tranzacții (tabelul de asociere) este Order Window-ul;
- Fiecare client se va putea înregistra o singură dată (pentru a evita intrări redundante în baza noastră de date).
- Fiecare notă acordată produselor va fi strict între 1 și 10.
- Fiecare tabel ce nu este asociativ va fi identificat printr-o cheie unică PRIMARY KEY ce nu poate fi nulă sau 0, și va fi reprezentată printr-un număr întreg de la 1 la 30, după caz.
- Un client nu poate oferi mai mult de o notă unui singur produs, însă poate oferi note mai multor produse, iar un produs poate să primească note de la mai mulți clienți.
- Tabelele asociative sunt identificate prin cheile străine ale tabelelor relevante (în cazul notelor, vorbim despre client și produs)
- Fiecare produs este diferit, nu putem avea două produse identice în baza de date, de aceea avem câmpul de "cantitate" în cadrul acestuia.
- Nu pot fi doi curieri asociați aceleiași comenzi.

3.Descrierea entităților, incluzând precizarea cheii primare.

- **Client**
Primary key: ID_Client
Alte atribute: Nume, Email, Nr de telefon, Zi de nastere(optional)
- **Curier**
Primary key: ID_Angajat
Alte atribute: Nume, Ore De Lucru, Salariu
- **Comanda**
Primary key: ID_Comanda
Foreign keys: ID_Client
Alte atribute: Data
- **Factura**
Primary key: FK: ID_Comanda
Alte atribute: Metoda De Plata, Pret
- **Order Window**
Primary key: ID_Window
Foreign keys: ID_Comanda, ID_Produs
Alte atribute: Cantitate
- **Produs**
Primary key: ID_Produs
Alte atribute: Nume, Pret, Cantitate
- **Album**
Primary key: FK: ID_Produs
Foreign key: ID_Casa
Alte atribute: Nr Discuri
- **Vinyl**
Primary key: FK: ID_Produs
Foreign key: ID_Casa
Alte atribute: Variante
- **Record Label**
Primary key: ID_Label
Alte atribute: Nume, Tara
- **Instrument**
Primary key: ID_Instrument
Alte atribute: Tip, Material

4.Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Relatie	Entitati implicate	Cardinalitate
a_achitat	client-factura	one-to-many
a_comandat	client-comanda	one-to-many
a_facturat	comanda-factura	one-to-one
livreaza	curier-comanda	one-to-many
este_comandat	comanda-instrument	many-to-many (se face prin intermediul order window)
este_comandat	comanda-produs(album/vinyl)	many-to-many (se face prin intermediul order window)
lanseaza	record label-produs	one-to-many
acorda	client-produs	many-to-many(se face prin intermediul tabelului "nota")
-	produs-album	one-to-one (album este subentitate a produsului)
-	produs-vinyl	one-to-one(vinyl ul este subentitate a produsului)
-	produs-instrument	one-to-one(instrument ul este subentitate a produsului)

Observații:

- Nu este obligatoriu ca un client să fi efectuat o comandă;
- Este obligatoriu să existe cel puțin un produs comandat în fiecare comandă(implicit și order window).
- Este obligatoriu să existe o factură aferentă fiecărei comenzi;
- Nu este obligatoriu să existe un curier asociat fiecărei comenzi;
- Este obligatoriu ca fiecare album/vinyl să fie legat de un record label;
- Este obligatoriu ca fiecare instanță a entității “Produs” să fie sau album, vinyl sau instrument.

5.Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Client:

Atribut	Tip de date	Constrangeri	Valori implicite	Valori posibile	Observatii
ID_CLIENT	INT	PK		1	
Nume	VARCHAR(100)	NOT NULL		Andrei Popa	
NrTel	VARCHAR(20)			0724004004	
Email	VARCHAR(100)	UNIQUE		andrei.popa@gmail.com	
ZiNastere	DATE		<null>	<null>	

Factura:

Atribut	Tip de date	Constrangeri	Valori implicite	Valori posibile	Observatii
ID_COMANDA	INT	PK, FK		2	
ID_CLIENT	INT	FK		4	
Pret	DECIMAL(10, 2)	NOT NULL		300	
ModPlata	VARCHAR(10)	CHECK (mod_de_plata IN ('card', 'ramburs'))	ramburs	card	

Comanda:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_COMANDA	INT	PK		3	
ID_CLIENT	INT	FK		5	
ID_CURIER	INT	FK		5	
Data	DATE	NOT NULL		14.05.2025	

Curier:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_CURIER	INT	PK		4	
OreMunca	INT			20	
Salariu	INT			3000	
Nume	VARCHAR(100)	NOT NULL		Sandu Sandel	

Order Window:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_COMANDA	INT	FK		6	
ID_PRODUS	INT	FK		11	
Cantitate	INT			2	

Produs:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_PRODUS	INT	PK		27	
Nume	VARCHAR(100)	NOT NULL		Saxofon	
Pret	DECIMAL(10, 2)	NOT NULL		9000	
Cantitate	INT			40	

Album:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_PRODUS	INT	PK, FK		2	
ID_CASA	INT	FK		3	
Artist	VARCHAR(100)			Evanescence	
Nr_Discuri	INT	NOT NULL		2	

Vinyl:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_PRODUS	INT	PK, FK		5	
ID_CASA	INT	FK		7	
Artist	VARCHAR(100)			System Of a Down	
Variante	INT			3	

Instrument:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_PRODUS	INT	PK, FK		4	
Tip	VARCHAR(20)	NOT NULL		Suflat	
Material	VARCHAR(20)			Alama	

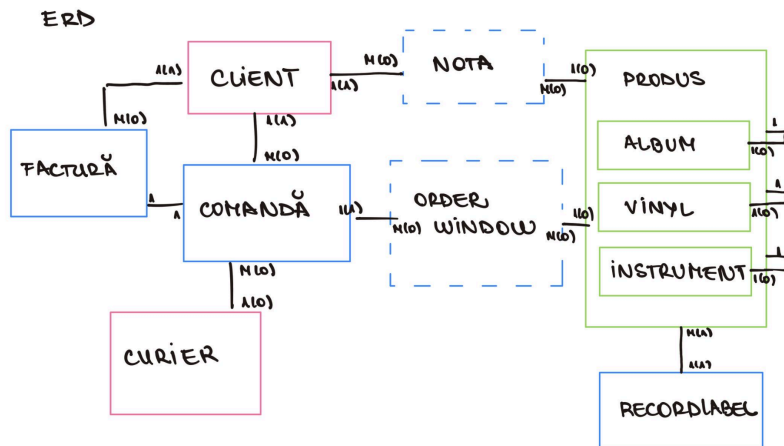
Record label:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_CASA	INT	PK		8	
Nume	VARCHAR(50)	NOT NULL		Sony Music	
Tara	VARCHAR(20)			Japonia	

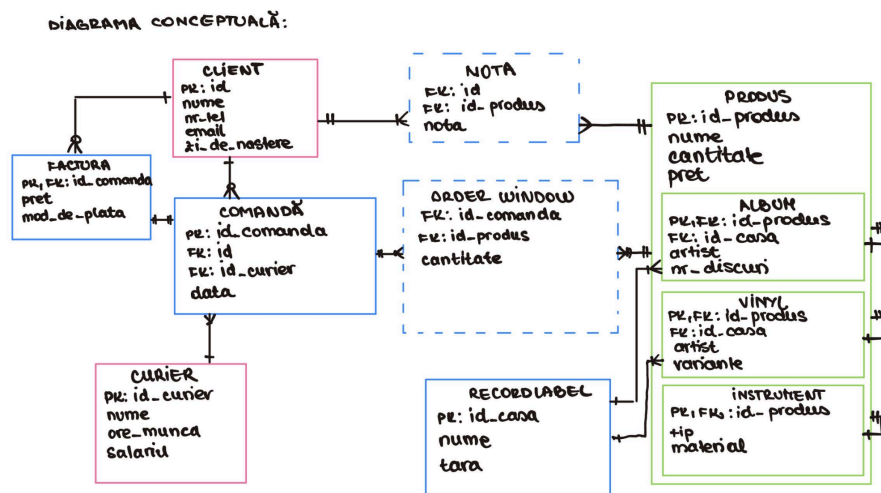
Rating:

Atribut	Tip de date	Constrângeri	Valori implicite	Valori posibile	Observații
ID_CLIENT	INT	FK		9	
ID_PRODUS	INT	FK		3	
Nota	INT	CHECK (nota > 0 AND nota <= 10)		8	

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.



8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

CLIENT(id#, nume, nr_telefon, email, zi_de_nastere)

COMANDA(id_comanda#, id#, id_curier#, data)

FACTURA(id_comanda#, pret, mod_de_plata)

PRODUS(id_produs#, nume, cantitate, pret)

ALBUM(id_produs#, id_casa#, artist, nr_discuri)

VINYL(id_produs#, id_casa#, artist, variante)

INSTRUMENT(id_produs#, tip, material)

CURIER(id_curier#, nume, ore_de_munca, salariu)

RECORDLABEL(id_casa#, nume, tara)

ORDERWINDOW(id_comanda#, id#, id_produs#, cantitate)

NOTA(id#, id_produs#, nota)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

În primul rând, doresc să menționez că diagrama conceptuală de la cerința 7, s-a aflat de la bun început în forma normală 3, astfel nu a fost nevoie să modific schema.

- Verificarea și aducerea la forma normală 1 (FN1):

Am verificat diagrama conceptuală de la cerința 7 pentru a nu exista intrări redundante și valori ce se repetă în schemă. În forma sa inițială, diagrama se află deja în acest stadiu, întrucât toate cheile primare sunt independente, iar datele și attributele din tabelele prezentate sunt unice și prezintă condiții pentru a le valida unicitatea. În plus, tabelele conțin doar date atomice, fără liste.

De exemplu, deși comanda și factura ar putea să aibă informații comune, le-am păstrat doar pe cele strict relevante pentru fiecare tabel.

Cu ajutorul tabelelor asociative am confirmat lipsa valorilor multiple în tabele, rezolvând problematica relațiilor de tip “many-to-many”.

Un exemplu de situație în care tabelul nu s-ar fi aflat în această formă ar fi fost cazul în care aş fi avut multiple produse într-o singură instanță a tabelului “comandă”, deoarece datele nu ar fi fost de tip atomic. Problema este rezolvată cu ajutorul tabelului asociativ “OrderWindow”.

- Aducerea la forma normală 2 (FN2):

Față de forma normală 1, cea de-a doua aduce în plus că fiecare atribut ce nu este cheie primară este dependent complet de acel primary key. Astfel, doar prin accesarea ID-ului instanței de care am nevoie, cu ajutorul cererilor și subcererilor putem afla orice informație dorită, fără riscul de a întâmpina date introduse de multiple ori sau attribute inaccesibile.

Spre exemplu, între tabelele client și comandă am reținut doar informațiile strict necesare, legându-le pe fiecare de ID-ul potrivit, putând accesa orice detaliu despre orice comandă a utilizatorilor, având certitudinea că vor fi afișate doar datele corecte.

Un exemplu în care FN2 nu ar fi fost respectat ar fi păstrarea numelui produsului în tabelul “OrderWindow”. Deși este relevant pentru comandă, această informație poate fi ușor aflată cu ajutorul unei interogări simple. Numele produsului nu depinde de ID-ul comenzii, așa că va fi păstrat în tabelul potrivit, și anume “Produs”.

- Aducerea la forma normală 3 (FN3):

Cea de-a 3-a formă normală aduce și lipsa dependenței atributelor non-cheie între ele. Astfel, unica dependență pe care o au attributele se află la nivelul cheii

principale din tabelul aferent. Redundanțele sunt excluse complet, facilitând interogările la nivelul bazei de date.

Un exemplu concret ar fi reprezentat de superentitatea produs și subentitățile ei, acestea fiind complet independente între ele, fiind legate doar de “id_produș”, intermediul prin care vom efectua în cerințele următoare cererile SQL.

Am evitat încălcarea acestei forme normale prin a nu reține informații irelevante, precum vârsta unui client, ce poate fi aflată prin calcul (după cum am prezentat în interogările din punctele următoare), aflându-se o relație tranzitivă de dependență, fără a reține nimic în plus.

10+11.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

- Am atașat atât fișierul cu cererile de creare tabele, cât și cele de inserare a datelor.

```

1 --Crearea tabelei
2 CREATE TABLE Client (
3     id NUMBER PRIMARY KEY,
4     nume VARCHAR2(100) NOT NULL,
5     nr_telefon VARCHAR2(20),
6     email VARCHAR2(100) NOT NULL UNIQUE,
7     zi_nasterii DATE
8 );
9
10 CREATE TABLE Curier (
11     id_curier NUMBER PRIMARY KEY,
12     nume VARCHAR2(100) NOT NULL,
13     nr_telefon VARCHAR2(20),
14     nr_p_munca NUMBER,
15     salariu NUMBER(10,2)
16 );
17
18 CREATE TABLE Comanda (
19     id_comanda NUMBER PRIMARY KEY,
20     id NUMBER NOT NULL,
21     id_curier NUMBER NOT NULL,
22     data DATE DEFAULT SYSDATE,
23
24     CONSTRAINT fk_client FOREIGN KEY (id) REFERENCES Client(id),
25     CONSTRAINT fk_curier FOREIGN KEY (id_curier) REFERENCES Curier(id_curier)
26 );
27
28 CREATE TABLE RecordLabel (

```

```

1 CREATE TABLE RecordLabel (
2     id_casa NUMBER PRIMARY KEY,
3     nume VARCHAR2(100) NOT NULL,
4     tara VARCHAR2(50)
5 );
6
7 CREATE TABLE Produs (
8     id_produs NUMBER PRIMARY KEY,
9     nume VARCHAR2(100) NOT NULL,
10    cantitate NUMBER NOT NULL,
11    pret NUMBER(10,2) NOT NULL
12 );
13
14 CREATE TABLE Albom (
15     id_produs NUMBER PRIMARY KEY,
16     id_casa NUMBER,
17     artist VARCHAR2(100),
18     nr_discuiri NUMBER,
19     FOREIGN KEY (id_produs) REFERENCES Produs(id_produs),
20     FOREIGN KEY (id_casa) REFERENCES RecordLabel(id_casa)
21 );
22
23 CREATE TABLE Vinyt (
24     id_produs NUMBER PRIMARY KEY,
25     id_casa NUMBER,
26     artist VARCHAR2(100),
27     nr_discuiri NUMBER,
28     FOREIGN KEY (id_produs) REFERENCES Produs(id_produs),
29     FOREIGN KEY (id_casa) REFERENCES RecordLabel(id_casa)
30 );

```

```

59 CREATE TABLE Instrument (
60     id_produs NUMBER PRIMARY KEY,
61     tip VARCHAR2(50),
62     material VARCHAR2(50),
63     FOREIGN KEY (id_produs) REFERENCES Produs(id_produs)
64 );
65
66 CREATE TABLE Factura (
67     id_comanda INT PRIMARY KEY,
68     pret DECIMAL(10, 2) NOT NULL,
69     mod_de_plata VARCHAR2(20) DEFAULT 'ramburs' CHECK (mod_de_plata IN ('ramburs', 'card')),
70     FOREIGN KEY (id_comanda) REFERENCES comanda(id_comanda)
71 );
72
73 CREATE TABLE orderWindow (
74     id_comanda INT,
75     id_produs INT,
76     data DATE INT NOT NULL CHECK (cantitate >= 1),
77     PRIMARY KEY (id_comanda, id_produs),
78     FOREIGN KEY (id_comanda) REFERENCES comanda(id_comanda),
79     FOREIGN KEY (id_produs) REFERENCES produs(id_produs)
80 );
81
82 CREATE TABLE Nota (
83     id INT,
84     id_produs INT,
85     nota INT CHECK (nota BETWEEN 1 AND 10),
86     FOREIGN KEY (id) REFERENCES Client(id),

```

```

INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (10, 'nume' 'Andreea Maria', [nr_telefon] '0716778990', [email] 'andreea.maria@gmail.com');

INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu)
VALUES (10, 'nume' 'Adi ColTelivrea', [nr_telefon] '0725678645', [ore_munca] '25', [salariu] '3900');

INSERT INTO Comanda (id_comanda, id, id_curier, data)
VALUES (10, 10, 10, [data] TO_DATE('2025-04-15', 'YYYY-MM-DD'));

INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (10, 'nume' 'BMG Rights Management', [tara] 'Germania');

INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (1, 'nume' 'Alveolan', [cantitate] 160, [pret] 34.00);

```



```

--Create tabele
CREATE TABLE Client (
  id NUMBER PRIMARY KEY,
  nume VARCHAR2(100) NOT NULL,
  nr_telefon VARCHAR2(20),
  email VARCHAR2(100) NOT NULL UNIQUE,
  zi_nastere DATE
);

CREATE TABLE Curier (
  id_curier NUMBER PRIMARY KEY,
  nume VARCHAR2(100) NOT NULL,
  nr_telefon VARCHAR2(20),
  ore_munca NUMBER,
  salariu NUMBER(10,2)
);

CREATE TABLE Comanda (
  id_comanda NUMBER PRIMARY KEY,
  id NUMBER NOT NULL,
  id_curier NUMBER NOT NULL,
  data DATE DEFAULT SYSDATE,

  CONSTRAINT fk_client FOREIGN KEY (id) REFERENCES Client(id),
  CONSTRAINT fk_curier FOREIGN KEY (id_curier) REFERENCES Curier(id_curier)
);

CREATE TABLE RecordLabel (
  id_casa NUMBER PRIMARY KEY,
  nume VARCHAR2(100) NOT NULL,
  tara VARCHAR2(50)
);

CREATE TABLE Produs (
  id_produs NUMBER PRIMARY KEY,
  nume VARCHAR2(100) NOT NULL,
  cantitate NUMBER NOT NULL,
  pret NUMBER(10,2) NOT NULL
);

CREATE TABLE Album (
  id_produs NUMBER PRIMARY KEY,
  id_casa NUMBER,
  artist VARCHAR2(100),
  nr_discuri NUMBER,
  FOREIGN KEY (id_produs) REFERENCES Produs(id_produs),

```

```

    FOREIGN KEY (id_casa) REFERENCES RecordLabel(id_casa)
);

CREATE TABLE Vinyl (
    id_produș NUMBER PRIMARY KEY,
    id_casa NUMBER,
    artist VARCHAR(100),
    varianta NUMBER,
    FOREIGN KEY (id_produș) REFERENCES Produș(id_produș),
    FOREIGN KEY (id_casa) REFERENCES RecordLabel(id_casa)
);

CREATE TABLE Instrument (
    id_produș NUMBER PRIMARY KEY,
    tip VARCHAR2(50),
    material VARCHAR2(50),
    FOREIGN KEY (id_produș) REFERENCES Produș(id_produș)
);

CREATE TABLE Factura (
    id_comanda INT PRIMARY KEY,
    pret DECIMAL(10, 2) NOT NULL,
    mod_de_plata VARCHAR(20) DEFAULT 'ramburs' CHECK (mod_de_plata IN
('ramburs', 'card')),
    FOREIGN KEY (id_comanda) REFERENCES comanda(id_comanda)
);

CREATE TABLE orderWindow (
    id_comanda INT,
    id_produș INT,
    cantitate INT NOT NULL CHECK (cantitate >= 1),
    PRIMARY KEY (id_comanda, id_produș),
    FOREIGN KEY (id_comanda) REFERENCES comanda(id_comanda),
    FOREIGN KEY (id_produș) REFERENCES produș(id_produș)
);

CREATE TABLE Nota (
    id INT,
    id_produș INT,
    nota INT CHECK (nota BETWEEN 1 AND 10),
    FOREIGN KEY (id) REFERENCES Client(id),
    FOREIGN KEY (id_produș) REFERENCES Produș(id_produș)
);

--Inserare date
INSERT INTO Client (id, nume, nr_telefon, email, zi_nastere)

```

```

VALUES (1,'Gigel Ionescu', '0722123456', 'gigel.ionescu@gmail.com', 1983-01-01);
INSERT INTO Client (id, nume, nr_telefon, email, zi_nastere)
VALUES (2,'Petrica Marinescu', '0722233445', 'petrica.marinescu@gmail.com',
1999-12-26);
INSERT INTO Client (id, nume, nr_telefon, email, zi_nastere)
VALUES (3,'Strutul Modest', '0700556677', 'strutul.modest@gmail.com', 1981-12-02);
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (4,'Britney Spears', '412-222-5555', 'hitmebabyonemoretime@gmail.com');
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (5,'Batman', '0724504188', 'liliacul.salvator@gmail.com');
INSERT INTO Client (id, nume, nr_telefon, email, zi_nastere)
VALUES (6,'Ioana Grigore', '0769888444', 'ioana.grigore@gmail.com', 1974-05-12);
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (7,'Camil Petrescu', '0719601988', 'patullui.procust@gmail.com');
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (8,'Ion AlGlanetasului', '0754332331', 'ion.din.pripas@yahoo.com');
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (9,'Marius Andrei', '0711235813', 'marius.andrei@yahoo.com');
INSERT INTO Client (id, nume, nr_telefon, email)
VALUES (10,'Andreea Maria', '0716778990', 'andreea.maria@gmail.com');

INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(1, 'Alina Vasile', '0744566778', 25, 4000.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(2, 'Corleone Brat', '0783445112', 30, 4020.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(3, 'Maricica Paraschiv', '0722123456', 28, 4000.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(4, 'Eduardo CelMic', '0724500600', 24, 3800.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(5, 'Mickey Mouse', '0723888765', 26, 3900.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(6, 'Curierul Fan', '0724448765', 40, 4200.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(7, 'Andrei Ionescu', '0723123456', 40, 3900.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(8, 'Maria Popa', '0734987654', 35, 3000.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(9, 'Vlad Dumitrescu', '0755123987', 20, 1800.00);
INSERT INTO Curier (id_curier, nume, nr_telefon, ore_munca, salariu) VALUES
(10, 'Adi CelCeLivreaza', '0725678465', 25, 3900.00);

INSERT INTO Comanda (id_comanda, id, id_curier, data)
VALUES (1, 1, 10, TO_DATE('2025-04-15', 'YYYY-MM-DD'));
INSERT INTO Comanda (id_comanda, id, id_curier)

```

```

VALUES (2, 2, 7);
INSERT INTO Comanda (id_comanda, id, id_curier)
VALUES (3, 1, 7);
INSERT INTO Comanda (id_comanda, id, id_curier, data)
VALUES (4, 4, 2, TO_DATE('2024-12-01', 'YYYY-MM-DD'));
INSERT INTO Comanda (id_comanda, id, id_curier)
VALUES (5, 5, 8);
INSERT INTO Comanda (id_comanda, id, id_curier)
VALUES (6, 6, 1);
INSERT INTO Comanda (id_comanda, id, id_curier)
VALUES (7, 7, 10);
INSERT INTO Comanda (id_comanda, id, id_curier)
VALUES (8, 8, 6);
INSERT INTO Comanda (id_comanda, id, id_curier, data)
VALUES (9, 3, 7, TO_DATE('2025-05-01', 'YYYY-MM-DD'));
INSERT INTO Comanda (id_comanda, id, id_curier, data)
VALUES (10, 6, 2, TO_DATE('2025-04-15', 'YYYY-MM-DD'));

INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (1, 'Universal Music', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (2, 'Sony Music', 'Japonia');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (3, 'Warner Music', 'UK');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (4, 'EMI', 'UK');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (5, 'Island Records', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (6, 'Columbia Records', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (7, 'Atlantic Records', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (8, 'Capitol Records', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (9, 'Def Jam Recordings', 'USA');
INSERT INTO RecordLabel (id_casa, nume, tara)
VALUES (10, 'BMG Rights Management', 'Germania');

INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (1, 'Alveolar', 160, 34.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (2, 'Visatori cu PLumb in Ochi', 30, 40.50);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (3, 'Arhitectul din Babel', 32, 44.55);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (4, 'ZABA', 60, 85.00);

```

```

INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (5, 'How To Be a Human Being', 57, 80.19);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (6, 'Dreamland', 70, 80.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (7, 'ILYSFM', 68, 82.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (8, 'The Open Door', 68, 28.35);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (9, 'Mezmerize', 90, 59.13);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (10, 'This is All Yours', 30, 51.03);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (11, 'Hot Pink', 20, 97.20);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (12, 'Sweetener', 80, 194.40);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (13, 'Dangerous Woman', 40, 162.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (14, 'MAYHEM', 200, 210.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (15, 'Artpop', 190, 200.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (16, 'IMPERA', 30, 105.30);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (17, 'Dreamland', 20, 110.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (18, 'Toxicity', 12, 105.30);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (19, 'Fear of the Dark', 70, 129.60);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (20, 'So Close To What', 5, 110.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (21, 'Chitara Electrica', 3, 1620.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (22, 'Chitara Acustica', 5, 1900.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (23, 'Chitara Electroacustica', 7, 1701.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (24, 'Chitara Clasica', 2, 1900.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (25, 'Bas Electric', 6, 1782.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (26, 'Vioara', 8, 4050.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (27, 'Clarinet', 2, 9720.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (28, 'Trompeta', 3,

```

```

6480.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (29, 'Saxofon', 4,
9000.00);
INSERT INTO Produs (id_produs, nume, cantitate, pret) VALUES (30, 'Pianina', 2,
10500.00);

INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (1, 2, 'E-An-Na', 1);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (2, 2, 'Alternosfera',
2);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (3, 1, 'Alternosfera',
1);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (4, 3, 'Glass
Animals', 1);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (5, 6, 'Glass
Animals', 1);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (6, 3, 'Glass
Animals', 3);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (7, 8, 'Glass
Animals', 2);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (9, 4, 'System of a
Down', 1);
INSERT INTO Album (id_produs, id_casa, artist, nr_discuri) VALUES (10, 2, 'Alt-J', 1);

INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (11, 2, 'Doja Cat',
NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (12, 1, 'Ariana Grande',
3);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (13, 1, 'Ariana Grande',
NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (14, 4, 'Lady Gaga', 6);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (15, 6, 'Lady Gaga',
NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (16, 8, 'Ghost', NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (17, 10, 'Glass Animals',
2);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (18, 3, 'System of a
Down', NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (19, 5, 'Iron Maiden',
NULL);
INSERT INTO Vinyl (id_produs, id_casa, artist, varianta) VALUES (20, 7, 'Tate McRae',
NULL);

INSERT INTO instrument (id_produs, tip, material) VALUES (21, 'Corzi', 'Lemn, Plastic');
INSERT INTO instrument (id_produs, tip, material) VALUES (22, 'Corzi', 'Lemn');
INSERT INTO instrument (id_produs, tip, material) VALUES (23, 'Corzi', 'Lemn');
INSERT INTO instrument (id_produs, tip, material) VALUES (24, 'Corzi', 'Lemn');

```

```

INSERT INTO instrument (id_produș, tip, material) VALUES (25, 'Corzi', 'Lemn, Plastic');
INSERT INTO instrument (id_produș, tip, material) VALUES (26, 'Corzi', 'Lemn');
INSERT INTO instrument (id_produș, tip, material) VALUES (27, 'Suflat', 'Otel');
INSERT INTO instrument (id_produș, tip, material) VALUES (28, 'Suflat', 'Alama');
INSERT INTO instrument (id_produș, tip, material) VALUES (29, 'Suflat', NULL);
INSERT INTO instrument (id_produș, tip, material) VALUES (30, 'Corzi', NULL);

```

```

INSERT INTO Nota (id, id_produș, nota) VALUES (3, 3, 6);
INSERT INTO Nota (id, id_produș, nota) VALUES (4, 13, 5);
INSERT INTO Nota (id, id_produș, nota) VALUES (1, 20, 10);
INSERT INTO Nota (id, id_produș, nota) VALUES (5, 16, 1);
INSERT INTO Nota (id, id_produș, nota) VALUES (6, 14, 2);
INSERT INTO Nota (id, id_produș, nota) VALUES (6, 8, 8);
INSERT INTO Nota (id, id_produș, nota) VALUES (10, 21, 9);
INSERT INTO Nota (id, id_produș, nota) VALUES (3, 2, 9);
INSERT INTO Nota (id, id_produș, nota) VALUES (7, 5, 7);
INSERT INTO Nota (id, id_produș, nota) VALUES (32, 19, 8);

```

```

INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (1, 29, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (2, 1, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (3, 6, 3);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (4, 14, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (4, 15, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (5, 22, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (5, 17, 2);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (6, 4, 4);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (7, 15, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (8, 24, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (9, 7, 2);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (9, 30, 1);
INSERT INTO OrderWindow (id_comanda, id_produș, cantitate) VALUES (10, 20, 3);

```

```

INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (1, 9000, 'ramburs');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (2, 65, 'ramburs');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (3, 240, 'card');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (4, 390, 'ramburs');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (7, 190, 'ramburs');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (8, 1900, 'card');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (9, 10664, 'ramburs');
INSERT INTO Factura (id_comanda, pret, mod_de_plata) VALUES (10, 330, 'ramburs');

```

--Verificare

```
SELECT * FROM Produș;
```

```
COMMIT;
```

console					COMANDA	FACTURA	CLIENT	ALBUM	PRODUS
WHERE					ORDER BY				
ID_PRODUS	NUME	CANTITATE	PRET						
1	1 Alveolar	20	65.00						
2	2 Visatori cu Plumb in Ochi	30	40.50						
3	3 Arhitectul din Babel	32	44.55						
4	4 ZABA	60	85.00						
5	5 How To Be a Human Being	57	80.19						
6	6 Dreamland	70	80.00						
7	7 ILYSM	68	82.00						
8	8 The Open Door	60	28.35						
9	9 Mezeerize	90	59.13						
10	10 This is All Yours	30	51.03						
11	11 Hot Pink	20	97.20						
12	12 Sweetener	80	194.40						
13	13 Dangerous Woman	40	162.00						
14	14 MAYHEM	200	210.00						
15	15 Artpop	190	200.00						
16	16 IMPERA	30	105.30						
17	17 Dreamland	20	110.00						
18	18 Toxicity	12	105.30						
19	19 Fear of the Dark	70	129.60						
20	20 So Close To What	5	110.00						
21	21 Chitara Electrica	3	1620.00						
22	22 Chitara Acustica	5							
23	23 Chitara Electroatistica	7							

console					COMANDA	FACTURA	CLIENT	ALBUM	PRODUS
WHERE					ORDER BY				
ID_PRODUS	ID_CASA	ARTIST	NR_DISCURI						
1	1	2 E-An-Na	1						
2	2	2 Alternosfera	2						
3	3	1 Alternosfera	1						
4	4	3 Glass Animals	1						
5	5	6 Glass Animals	1						
6	6	3 Glass Animals	3						
7	7	8 Glass Animals	2						
8	8	4 System of a Down	1						
9	9	2 Alt-J	1						
10	10								

console					COMANDA	FACTURA	CLIENT	ALBUM	PRODUS
WHERE					ORDER BY				
ID	NUME	NR_TELEFON	EMAIL	ZI_MASTERE					
1	1 Gigel Ionescu	0722123456	gigel.ionescu@gmail.com	1992-08-15					
2	2 Petrica Marinescu	0722233445	petrica.marinescu@gmail.com	1983-01-01					
3	3 Strutul Modest	0700556677	strutul.modest@gmail.com	1999-12-26					
4	4 Britney Spears	412-222-5555	hitmebabyonemoretina@gmail.com	1981-12-02					
5	5 Batman	0724504188	liliascul.salvator@gmail.com	<null>					
6	6 Ioana Grigore	0769888444	ioana.grigore@gmail.com	1974-05-12					
7	7 Camil Petrescu	0719601988	patullui.procust@gmail.com	<null>					
8	8 Ion Alolanetasului	0754332331	ion.din.pripas@yahoo.com	<null>					
9	9 Marius Andrei	0711235813	marius.andrei@yahoo.com	<null>					
10	10 Andreea Maria	0716778990	andreea.maria@gmail.com	<null>					

console

COMANDA

FACTURA

CLIENT

ALBUM

PRODUS

<

console

COMANDA

FACTURA

CLIENT

ALBUM

PRODUS

Tx: Manual

DDL

WHERE

ORDER BY

ID_COMANDA	ID	ID_CURIER	DATA
1	1	1	10 2025-05-12 19:47:12
2	2	2	7 2025-05-12 19:47:33
3	3	1	7 2024-12-01
4	4	4	2 2025-05-12 19:49:32
5	5	5	8 2025-05-12 19:49:36
6	6	6	1 2025-05-12 19:49:40
7	7	7	10 2025-05-12 19:49:44
8	8	8	6 2025-05-12 19:49:50
9	9	3	7 2025-05-01
10	10	6	2 2025-04-15

console					CURIER	INSTRUMENT	NOTA	ORDERWINDOW	RECORDLABEL	VINYL
WHERE					ORDER BY					
ID_CURIER	NUME	NR_TELEFON	ORE_MUNCA	SALARIU						
1	1 Alina Vasile	0744566778	25	4000.00						
2	2 Corleone Brat	0783445112	30	4020.00						
3	3 Maricica Paraschiv	0722123456	28	4000.00						
4	4 Eduardo CelMic	0724500400	24	3800.00						
5	5 Mickey Mouse	0723888765	26	3900.00						
6	6 Curierul Fan	0724448765	40	4200.00						
7	7 Andrei Ionescu	0723123456	40	3900.00						
8	8 Maria Popa	0734987654	35	3600.00						
9	9 Vlad Dumitrescu	0755123987	20	1800.00						
10	10 Adi CelCelLivraaza	0725678465	25	3900.00						

console INSTRUMENT x NOTA ORDERWINDOW RECORDLABEL VINYL

WHERE ORDER BY

ID_PRODUS	TIP	MATERIAL
21	Corzi	Lemn, Plastic
22	Corzi	Lemn
23	Corzi	Lemn
24	Corzi	Lemn
25	Corzi	Lemn, Plastic
26	Corzi	Lemn
27	Suflat	Otel
28	Suflat	Alama
29	Suflat	<null>
30	Corzi	<null>

console NOTA x ORDERWINDOW RECORDLABEL VINYL

WHERE ORDER BY

ID	ID_PRODUS	NOTA
3	3	6
4	4	5
1	20	10
5	16	1
6	14	2
6	8	8
10	21	9
3	2	9
7	5	7
2	19	8

console ORDERWINDOW x RECORDLABEL VINYL

WHERE ORDER BY

ID_COMANDA	ID_PRODUS	CANTITATE
1	1	29
2	2	1
3	3	6
4	4	14
5	4	15
6	5	22
7	5	17
8	6	4
9	7	15
10	8	24
11	9	7
12	9	30
13	10	20

console RECORDLABEL x VINYL

WHERE ORDER BY

ID_CASA	NUME	TARA
1	Universal Music	USA
2	Sony Music	Japonia
3	Warner Music	UK
4	EMI	UK
5	Island Records	USA
6	Columbia Records	USA
7	Atlantic Records	USA
8	Capitol Records	USA
9	Def Jam Recordings	USA
10	BMG Rights Management	Germania

console VINYL x

WHERE ORDER BY

ID_PRODUS	ID_CASA	ARTIST	VARIANTA
11	2	Doja Cat	<null>
12	1	Ariana Grande	3
13	1	Ariana Grande	<null>
14	4	Lady Gaga	6
15	6	Lady Gaga	<null>
16	8	Ghost	<null>
17	10	Glass Animals	2
18	3	System of a Down	<null>
19	5	Iron Maiden	<null>
20	7	Tate McRae	<null>

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe.

(Am atașat separat și documentul cu cererile SQL. Sunt grupate după cerințele a), b), c), d), respectiv e) și f) în ultima interogare)

```
--selecteaza toate comenzile cu albume livrate de curieri cu salariul de peste 3000 de lei
SELECT c.ume
FROM Client c
JOIN Comanda com ON c.id = com.id
JOIN OrderWindow ow ON com.id_comanda = ow.id_comanda
JOIN Album a ON ow.id_produs = a.id_produs
JOIN Curier cu ON com.id_curier = cu.id_curier
WHERE cu.salariu > 3000;
```

```
-- selecteaza si sorteaza descrescatori comenzile in functie de numarul de produse
comandate in fiecare
SELECT o.id_comanda
FROM (
    SELECT id_comanda, SUM(cantitate) AS total_produce
    FROM OrderWindow
    GROUP BY id_comanda
) o
ORDER BY o.total_produce DESC;
```

```
--id ul clientilor care au comandat in total mai mult de 5 produse
SELECT c.id, COUNT(DISTINCT com.id_comanda) AS nr_comenzi
FROM Client c
JOIN Comanda com ON c.id = com.id
JOIN (
    SELECT id_comanda, SUM(cantitate) AS total
    FROM OrderWindow
    GROUP BY id_comanda
) ow ON com.id_comanda = ow.id_comanda
GROUP BY c.id
HAVING SUM(ow.total) > 5;
```

```
--selecteaza varstele pentru fiecare client sau arata "necunoscut" pt cei ce nu o au
inregistrata
SELECT
    id AS "ID Client",
    INITCAP(ume) AS "Nume Client",
    NVL(
        TO_CHAR(zi_nastere, 'DD Month YYYY', 'NLS_DATE_LANGUAGE = ENGLISH'),
        'Necunoscută'
    ) AS "Zi de Naștere",
    CASE
```

```

    WHEN zi_nastere IS NULL THEN 'Vârstă necunoscută'
    ELSE TO_CHAR(TRUNC(MONTHS_BETWEEN(SYSDATE, zi_nastere) / 12)) || ' ani'
END AS "Vârstă"
FROM Client
ORDER BY "Nume Client";

```

--se afiseaza clientii ce au plasat cel putin 2 comenzi si se afiseaza varsta lor (daca au data de nastere inregistrata), ultima comanda si gradul de fidelitate

```

WITH clienti_comenzi AS (
    SELECT
        c.id,
        INITCAP(c.numa) AS nume,
        COUNT(com.id_comanda) AS nr_comenzi,
        MAX(com.data) AS ultima_comanda,
        EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM c.zi_nastere) AS varsta
    FROM Client c
    JOIN Comanda com ON c.id = com.id
    GROUP BY c.id, c.numa, c.zi_nastere
)
SELECT
    id AS "ID Client",
    nume AS "Nume Client",
    nr_comenzi AS "Număr Comenzi",
    TO_CHAR(ultima_comanda, 'DD Mon YYYY') AS "Ultima Comandă",
    varsta AS "Vârstă",
    CASE
        WHEN nr_comenzi >= 10 THEN 'VIP'
        WHEN nr_comenzi >= 5 THEN 'Frecvent'
        ELSE 'Ocazional'
    END AS "Categorie Client"
FROM clienti_comenzi
WHERE nr_comenzi >= 2
ORDER BY nr_comenzi DESC;

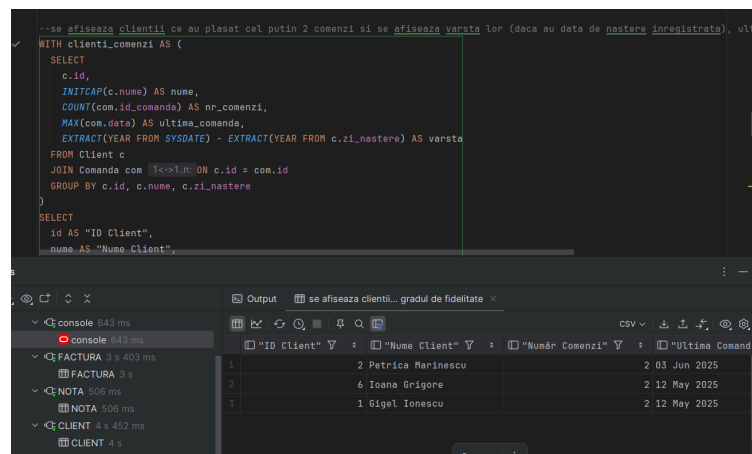
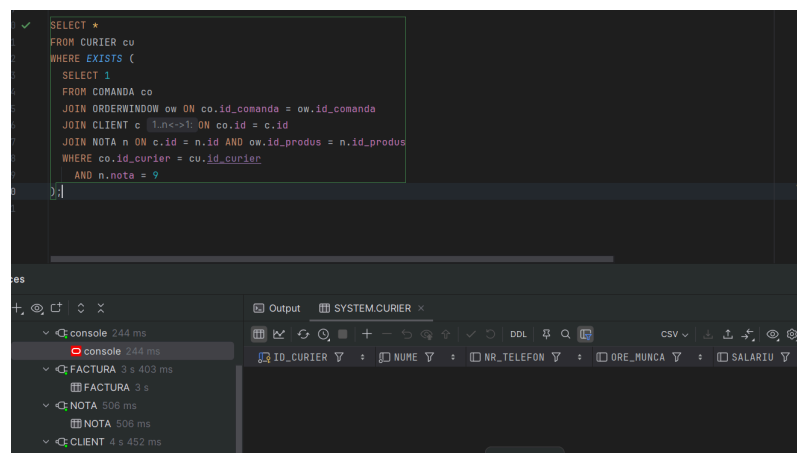
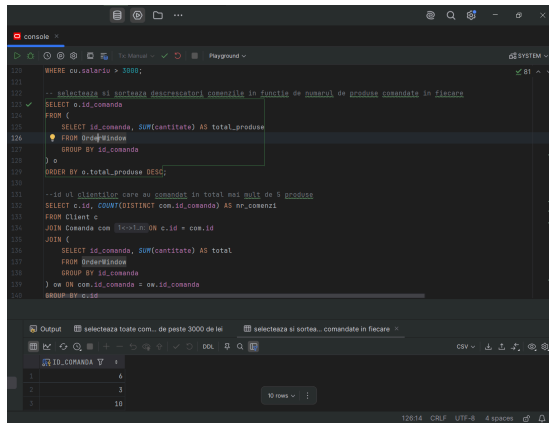
```

--in urma feedback-ului, cerinta a) ce contine si o subcerere cu EXISTS

```

SELECT *
FROM CURIER cu
WHERE EXISTS (
    SELECT 1
    FROM COMANDA co
    JOIN ORDERWINDOW ow ON co.id_comanda = ow.id_comanda
    JOIN CLIENT c ON co.id = c.id
    JOIN NOTA n ON c.id = n.id AND ow.id_produs = n.id_produs
    WHERE co.id_curier = cu.id_curier
    AND n.nota = 9
);

```



```
--id ul clientilor care au comandat in total mai mult de 5 produse
SELECT c.id, COUNT(DISTINCT com.id_comanda) AS nr_comenzi
FROM Client c
JOIN Comanda com 1<->1..n: ON c.id = com.id
JOIN (
  SELECT id_comanda, SUM(cantitate) AS total
  FROM OrderWindow
  GROUP BY id_comanda
) ow ON com.id_comanda = ow.id_comanda
GROUP BY c.id
HAVING SUM(ow.total) > 5;

--se afiseaza clientii ce au plasat cel putin 2 comenzi si se afiseaza varsta lor (dac
WITH clienti_comenzi AS (
```

Output: id ul clientilor car...mai mult de 5 produse

ID	NR_COMENZI
1	6
2	2

console 275 ms
 console 275 ms
 FACTURA 3 s 403 ms
 FACTURA 3 s
 NOTA 506 ms
 NOTA 506 ms

```
SELECT
  id AS "ID Client",
  INITCAP(ume) AS "Nume Client",
  NVL(
    TO_CHAR(zi_nastere, 'DD Month YYYY', 'NLS_DATE_LANGUAGE = ENGLISH'),
    'Necunoscută'
  ) AS "Zi de Naștere",
  CASE
    WHEN zi_nastere IS NULL THEN 'Vârstă necunoscută'
    ELSE TO_CHAR(TRUNC(MONTHS_BETWEEN(SYSDATE, zi_nastere) / 12)) || ' ani'
  END AS "Vârstă"
FROM Client
ORDER BY "Nume Client";

--update uri si delete uri

--se ofera o reducere de 10% la produsele ce nu au fost comandate
```

Output: am cautat pe interne...pentru interogari sql

ID Client	Nume Client	Zi de Naștere	Vârstă
1	10 Andreea Maria	Necunoscută	Vârstă necunoscută
2	5 Batman	Necunoscută	Vârstă necunoscută
3	4 Britney Spears	02 December 1981	43 ani
4	7 Camil Petrescu	unoscute	Vârstă necunoscută
5	1 Gigel Ionescu	August 1992	32 ani

console 501 ms
 console 501 ms
 FACTURA 3 s 403 ms
 FACTURA 3 s
 NOTA 506 ms
 NOTA 506 ms
 CLIENT 4 s 452 ms
 CLIENT 4 s

13.Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

```
--se scade cu 10% pretul produselor care nu au fost comandate niciodata
UPDATE Produs
SET pret = pret * 0.9
WHERE id_produs NOT IN (
    SELECT DISTINCT id_produs
    FROM OrderWindow
);
```

```
--se vor sterge albumele ce au fost produse de o casa de discuri bazata in Germania
DELETE FROM Album
WHERE id_produs IN (
    SELECT a.id_produs
    FROM Album a
    JOIN RecordLabel rl ON a.id_casa = rl.id_casa
    WHERE rl.tara = 'Germania'
);
```

```
--schimba modul de plata in "card" pentru clientii cu ziua de nastere in luna curenta
UPDATE Factura
SET mod_de_plata = 'card'
WHERE id_comanda IN(
    SELECT id
    FROM Client
    WHERE EXTRACT(MONTH FROM zi_nastere) = EXTRACT(MONTH FROM SYSDATE)
);
```

*de menționat că am dat run de 2 ori din greșeală la update.

```
console PRODUK
FROM Client
ORDER BY "Nume Client";
--update uri si delete uri
UPDATE Produs
SET pret = pret * 0.9
WHERE id_produs NOT IN (
  SELECT DISTINCT id_produs
  FROM OrderWindow
);
select * from produs;
DELETE FROM Produs
WHERE id_produs IN (
  SELECT a.id_produs
  FROM Album a
  JOIN RecordLabel r1 ON a.id_casa = r1.id_casa
  WHERE r1.terra = 'Germania'
```

Output selecteaza toate com... de peste 3000 de lei SYSTEMPRODUK

ID_PRODUS	NUME	CANTITATE	PRET
27	Clarinet	2	9720.00
28	Trompeta	3	4480.00
29	Saxofon	4	9000.00

```
console ALBUM PRODUK
ORDER BY "Nume Client";
--update uri si delete uri
UPDATE Produs
SET pret = pret * 0.9
WHERE id_produs NOT IN (
  SELECT DISTINCT id_produs
  FROM OrderWindow
);
select * from produs;
DELETE FROM Album
WHERE id_produs IN (
  SELECT a.id_produs
  FROM Album a
  JOIN RecordLabel r1 ON a.id_casa = r1.id_casa
  WHERE r1.terra = 'Germania'
```

Output selecteaza toate com... de peste 3000 de lei SYSTEMALBUM

ID_PRODUS	ID_CASA	ARTIST	NR_DISCUR
7	7	Glass Animals	2
9	9	System of a Down	1
10	10	Alt-J	1

WHERE		ORDER BY			
ID_PRODUS	ID_CASA	ARTIST	NR_DISCUR		
1	1	2 E-An-Ma	1		
2	2	2 Alternosfera	2		
3	3	1 Alternosfera	1		
4	4	3 Glass Animals	1		
5	5	6 Glass Animals	1		
6	6	3 Glass Animals	3		
7	7	8 Glass Animals	2		
8	8	10 Evanescence	1		
9	9	4 System of a Down	1		
10	10	2 Alt-J	1		

--Cerințe opționale 14-19:

14.Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

→Un exemplu de operație de tip LMD permisă pe vizualizarea complexă este “select”, iar una nepermisă este “update”.

```
233
234 UPDATE raport_clienti
235 SET total_cheltuit = 0
236 WHERE id_client = 1;
237 commit;
```

[42000][1732]
ORA-01732: data manipulation operation not legal on this view
<https://docs.oracle.com/error-help/db/ora-01732/>
Position: 7

```
229 ✓ SELECT nume_client, total_cheltuit
230 FROM raport_clienti
231 WHERE total_cheltuit > 0
232 ORDER BY total_cheltuit ASC;
```

	NUME_CLIENT	TOTAL_CHELTUIT
1	Petrica Marinescu	65
2	Camil Petrescu	190
3	Ioana Grigore	670
4	Britney Spears	780
5	Ion Alblanetasului	1900
6	Batman	4240
7	Gigel Ionescu	9240
8	Strutul Modest	21328

→Am ales să creez un tabel cu statisticile tuturor clienților ce au efectuat o comandă, afișând numărul total de comenzi efectuate și suma de bani plătită. Pentru o vizualizare mai ordonată, recomand sortarea datelor din tabelul creat după suma de bani plătită sau numărul de comenzi efectuate (asemeni exemplului de select de mai sus).

```
--creare tabel
CREATE VIEW raport_clienti AS
SELECT
    c.id AS id_client,
    c.nume AS nume_client,
    COUNT(DISTINCT o.id_comanda) AS nr_comenzi,
    SUM(f.pret) AS total_cheltuit
FROM CLIENT c
JOIN COMANDA o ON c.id = o.id
JOIN FACTURA f ON o.id_comanda = f.id_comanda
JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
```



```

WHERE f.pret > 0
GROUP BY c.id, c.numa
HAVING SUM(f.pret) > 0;

--ordonarea in functie de pret
SELECT nume_client, total_cheltuit
FROM raport_clienti
WHERE total_cheltuit > 0
ORDER BY total_cheltuit ASC;

--exemplu pentru operatia nepermisa
UPDATE raport_clienti
SET total_cheltuit = 0
WHERE id_client = 1;

```

```

CREATE VIEW raport_clienti AS
SELECT
    c.id AS id_client,
    c.numa AS nume_client,
    COUNT(DISTINCT o.id_comanda) AS nr_comenzi,
    SUM(f.pret) AS total_cheltuit
FROM CLIENT c
JOIN COMANDA o 1<->1.n: ON c.id = o.id
JOIN FACTURA f ON o.id_comanda = f.id_comanda
JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
WHERE f.pret > 0
GROUP BY c.id, c.numa
HAVING SUM(f.pret) > 0;

```

console RAPORT_CLIENTI x ALBUM COMANDA

Tx: Manual DDL CSV

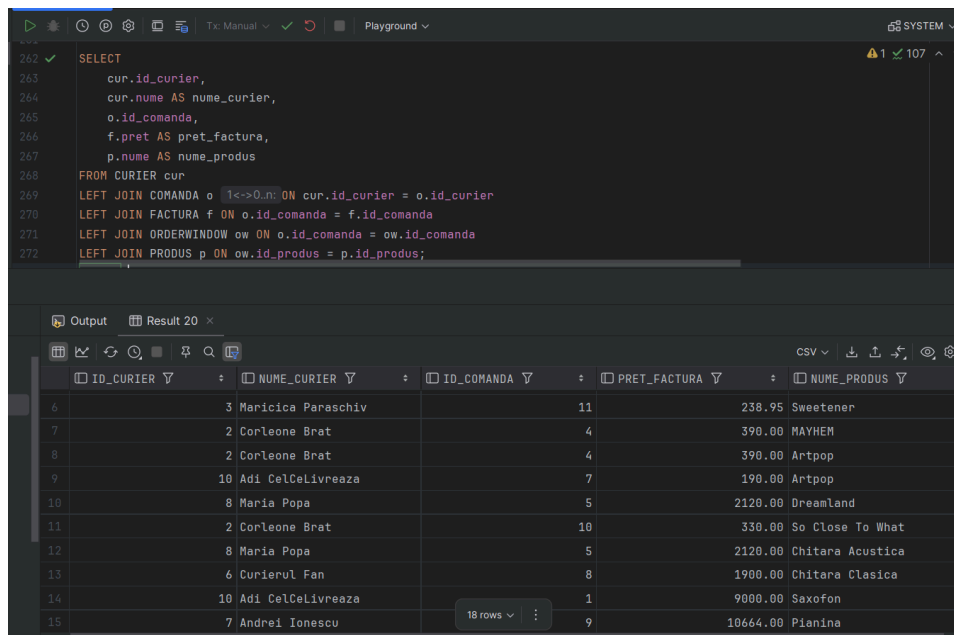
WHERE ORDER BY

	ID_CLIENT	NUME_CLIENT	NR_COMENZI	TOTAL_CHELTUIT
1	2	Petrica Marinescu	1	65
2	8	Ion AlGlanetasului	1	1900
3	7	Camil Petrescu	1	190
4	3	Strutul Modest	1	21328
5	6	Ioana Grigore	2	670
6	1	Gigel Ionescu	2	9240
7	4	Britney Spears	1	780
8	5	Batman	1	4240

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

→outer-join:

Pentru operația cu join-uri am plecat de la tabelul cu curieri pentru a afla id-ul comenzii, prețul și produsul comandat în comanda livrată de aceștia. Interogarea funcționează și pentru curierii ce nu au livrat nicio comandă, afișând <null>.



The screenshot shows a SQL playground with a query and its results. The query is as follows:

```
SELECT
  cur.id_curier,
  cur.num AS nume_curier,
  o.id_comanda,
  f.pret AS pret_factura,
  p.num AS nume_produs
FROM CURIER cur
LEFT JOIN COMANDA o ON cur.id_curier = o.id_curier
LEFT JOIN FACTURA f ON o.id_comanda = f.id_comanda
LEFT JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
LEFT JOIN PRODUS p ON ow.id_produs = p.id_produs;
```

The results table has 5 columns: ID_CURIER, NUME_CURIER, ID_COMANDA, PRET_FACTURA, and NUME_PRODUS. It contains 15 rows of data, with the last row showing a null value for ID_COMANDA.

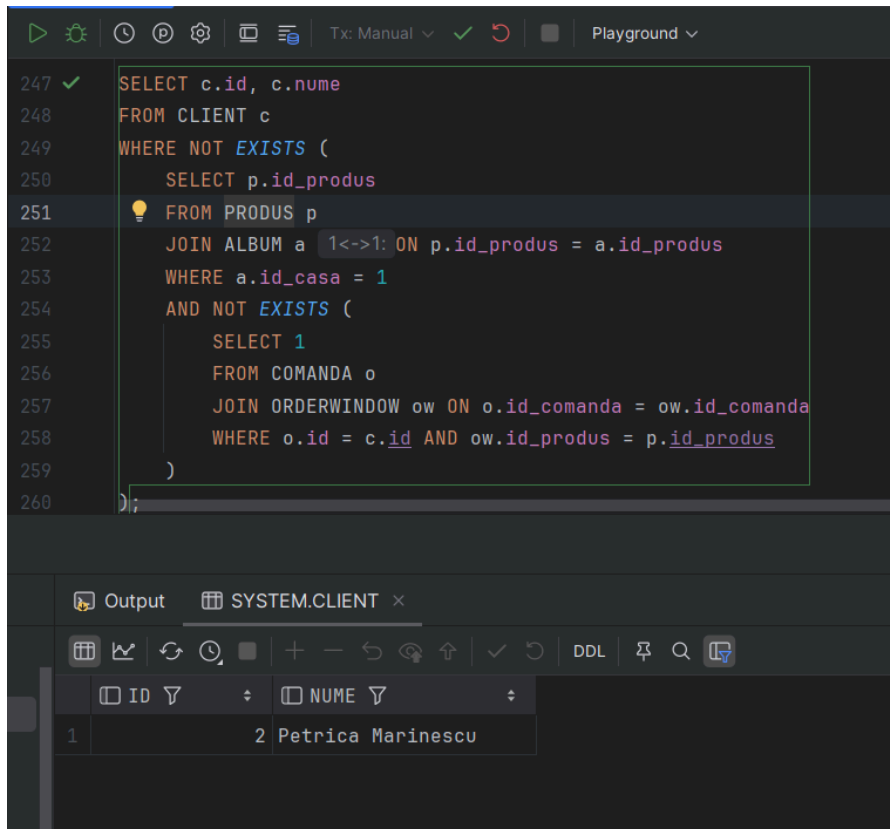
ID_CURIER	NUME_CURIER	ID_COMANDA	PRET_FACTURA	NUME_PRODUS
3	Maricica Paraschiv	11	238.95	Sweetener
2	Corleone Brat	4	390.00	MAYHEM
2	Corleone Brat	4	390.00	Artpop
10	Adi CelCelivreaza	7	190.00	Artpop
8	Maria Popa	5	2120.00	Dreamland
2	Corleone Brat	10	330.00	So Close To What
8	Maria Popa	5	2120.00	Chitara Acustica
6	Curierul Fan	8	1900.00	Chitara Clasica
10	Adi CelCelivreaza	1	9000.00	Saxofon
7	Andrei Ionescu	9	10664.00	Pianina

```
SELECT
  cur.id_curier,
  cur.num AS nume_curier,
  o.id_comanda,
  f.pret AS pret_factura,
  p.num AS nume_produs
FROM CURIER cur
LEFT JOIN COMANDA o ON cur.id_curier = o.id_curier
LEFT JOIN FACTURA f ON o.id_comanda = f.id_comanda
LEFT JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
LEFT JOIN PRODUS p ON ow.id_produs = p.id_produs;
```

→division:

În linii mari, operația de tip “division” este o interogare ce afișează toate instanțele de un anume tip “A” ce sunt legate de toate instanțele de tip “B”. În baza mea de date, inițial nu aveam un astfel de exemplu, însă am introdus o comandă nouă, cu id-ul 11, cu produsele cu id-ul 3 și 12.

Operația mea afișează clienții ce au comandat toate produsele asociate cu casa de discuri cu id-ul 1.



The screenshot shows a SQL playground interface. The top part displays a SQL query with line numbers 247 to 260. The query is a division query. The bottom part shows the results of the query in a table with two columns: ID and NUME. The table has one row with the value 1 in the ID column and Petrica Marinescu in the NUME column.

```
247 ✓ SELECT c.id, c.nume
248 FROM CLIENT c
249 WHERE NOT EXISTS (
250     SELECT p.id_produs
251     FROM PRODUS p
252     JOIN ALBUM a 1<->1: ON p.id_produs = a.id_produs
253     WHERE a.id_casa = 1
254     AND NOT EXISTS (
255         SELECT 1
256         FROM COMANDA o
257         JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
258         WHERE o.id = c.id AND ow.id_produs = p.id_produs
259     )
260 );
```

ID	NUME
1	Petrica Marinescu

```
SELECT c.id, c.nume
FROM CLIENT c
WHERE NOT EXISTS (
    SELECT p.id_produs
    FROM PRODUS p
    JOIN ALBUM a ON p.id_produs = a.id_produs
    WHERE a.id_casa = 1
    AND NOT EXISTS (
        SELECT 1
        FROM COMANDA o
        JOIN ORDERWINDOW ow ON o.id_comanda = ow.id_comanda
        WHERE o.id = c.id AND ow.id_produs = p.id_produs
    )
);
```

→analiza top-n:

Urmărind modelul de la subpunctul rezolvat anterior, voi afișa top 3 clienți ce au plătit cele mai mari sume de bani magazinului nostru. Am folosit condiția “rownum<=3” pentru a opri interogarea după primele 3 rezultate. De data asta, pentru a “simula” un top real, am sortat sumele descrescător.

```
237
238 ✓ SELECT id, nume, total_cheltuit FROM (
239     SELECT c.id, c.nume, SUM(f.pret) AS total_cheltuit
240     FROM CLIENT c
241     JOIN COMANDA o 1<->1.n: ON c.id = o.id
242     JOIN FACTURA f ON o.id_comanda = f.id_comanda
243     GROUP BY c.id, c.nume
244     ORDER BY total_cheltuit DESC
245 ) WHERE ROWNUM <= 3;
246
```

Output Result 17 x

	ID	NUME	TOTAL_CHELTUIT
1	3	Strutul Modest	10664
2	1	Gigel Ionescu	9240
3	5	Batman	2120

```
SELECT id, nume, total_cheltuit FROM (
    SELECT c.id, c.nume, SUM(f.pret) AS total_cheltuit
    FROM CLIENT c
    JOIN COMANDA o ON c.id = o.id
    JOIN FACTURA f ON o.id_comanda = f.id_comanda
    GROUP BY c.id, c.nume
    ORDER BY total_cheltuit DESC
) WHERE ROWNUM <= 3;
```

16.Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

Cererea pe care am efectuat-o va afișa totalul plasat de către fiecare client ce a făcut comenzi cu valoare adunată de peste 100 de lei. Am ales o cerere simplă pentru a putea evidenția în termeni mai mici folosirea expresiei și arborelui algebric aferent.

```

273
274 ✓ SELECT c.num, SUM(f.pret) AS total_cheltuit
275 FROM CLIENT c
276 JOIN COMANDA o 1<->1.n: ON c.id = o.id
277 JOIN FACTURA f ON o.id_comanda = f.id_comanda
278 GROUP BY c.num
279 HAVING SUM(f.pret) > 100;
280
281 commit;

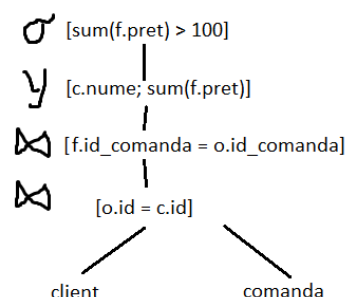
```

NUME	TOTAL_CHELTUIT
1 Britney Spears	390
2 Ioana Grigore	670
3 Ion AlGlanetasului	1900
4 Gigel Ionescu	9240
5 Batman	2120
6 Strutul Modest	10664
7 Petrica Marinescu	303.95
8 Camil Petrescu	190

Voi explica întâi arborele: În primul rând, el este format de “jos în sus”, începând cu informația finală pe care o vom afișa, iar frunzele sale vor fi tabelele de la care pornim. În acest caz, frunzele sunt tabelele “client” și “comanda”, pe care le unim prin operația “join”, urmând apoi încă un join pentru a ajunge la prețul unei facturi. (operația de JOIN între comanda și factură). Următorul pas este gruparea elementelor, în acest caz după nume și sumă, iar ultimul, rădăcina arborelui este selecția în sine, asociată condiției de a fi strict mai mare decât 100.

- σ (sigma) - selectia
- γ (gamma) - gruparea
- \bowtie (join) - alaturarea tabelor

Arborele algebric al interogarii



Expresia algebrică este mai puțin sugestivă decât arborele, însă reprezintă același lucru în esență:

$$\sigma_{\sum(f.pret)>100} (\gamma_{c.numc:\sum(f.pret)} ((CLIENT \bowtie_{c.id=o.id} COMANDA) \bowtie_{o.id=comanda=f.id.comanda} FACTURA))$$

17.Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării, justificând necesitatea acesteia.

→ **BCNF** (Boyce-Codd) implică că pentru fiecare dependență de la un tabel la altul, obiectul sau informația “de legătură” trebuie să fie o cheie de identificare. Baza mea de date se află deja în această formă, deoarece informațiile strict legate de fiecare tabel nu sunt chei primare sau secundare, iar cheile alese sunt strict cele relevante pentru operațiile de “join”.

Un exemplu bun în care tabelele nu se aflau în această formă ar fi fost cazul în care un atribut ar fi fost împărțit într-un tabel în care nu-și avea locul, cum ar fi salariul unui curier să se afle în tabelul cu comanda pe care o livrează. Aceasta dependență ar prezenta redundanțe, deoarece salariul unui curier nu depinde de comanda pe care o livrează, ci de curier în sine și contractul său de muncă, fiind legat de “id_curier”.

→**FN4:** Prima condiție pentru a se afla în această formă este verificarea normalizării la BCNF, pe care am validat-o la punctul anterior. Apoi, această formă presupune în plus să nu existe două tabele independente ce depind de o singură cheie primară. În alte cuvinte, ar încălca FN4 un tabel ce are mai multe atribute independente între ele, însă care depind de același PK.

În tabelul “client”, spre exemplu, atributele independente de alte chei, cum ar fi datele efective ale clientului sunt păstrate într-un singur tabel, fără să existe mai multe legate de același PK, eliminând redundanțele.

→**FN5:** Cele două condiții pentru aceasta sunt ca baza de date să se afle deja în FN4 și să nu avem apeluri de JOIN redundante, adică să avem un număr minim de tabele în care atributele sunt reținute doar în tabelul de care avem nevoie și care este potrivit pentru acea informație. Atributele legate de un PK se vor găsi în tabelul său și numai acolo.

Pentru a oferi un exemplu în care nu s-ar fi respectat această regulă, am putea spune că mutăm un atribut al clientului în tabelul facturilor. Acest lucru ar aduce atât repetiții la nivelul acelei informații (în cazul în care clientul efectuează mai mult de o comandă), cât și un apel în plus al operației JOIN.

Aplicarea **denormalizării** înseamnă să renunțăm la principiile de mai sus în favoarea unor interogări mai ușoare, cu mai puține join-uri. Spre exemplu, am putea uni tabelul de Factură cu cel de Comandă, ele având o relație one-to-one exclusivă. Totuși, avantajul ar fi un JOIN în minus și am renunța la o cheie de identificare primară. Un alt avantaj al denormalizării ar fi în cazul migrării la o bază de date NoSQL document-oriented, precum MongoDB. Am detaliat acest aspect în cerința opțională 19.