### UNIVERSITATEA DIN BUCUREȘTI

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**PROIECT**

# Baze de date

### Student

**Georgescu Andrei**

**Profesor**

**Lect. dr. Iulia Banu – Demergianu**

**Anul I**

**Semestrul 2**

**Grupa 1**

**București, iunie 2024**

#### Tema

Proiectarea unei baze de date relaţională (SQL) pentru site-ul comercial de electronice GadgetHaven – acest site reprezintă proiectul personal pentru materia Tehnici Web.

#### Tehnologii utilizate

Pentru realizarea acestui proiect am folosit următoarele tehnologii:

**Docker** - v24.0.5 ( pentru containerizarea bazei de date )

**OracleDB** – v19.3 ( baza de date SQL )

**SQL Developer** - v23.1.1 ( client pentru baza de date – pentru inserare date/interogare )

sqlplus sys/oracle19\_3 as sysdba

sqlplus proiect/proiect@localhost:1521/ORCLPDB1

SELECT table\_name FROM all\_tables;

ORCLPDB1

**Admin ( SYSDBA )**

Username: sys

Password: oracle19\_3

**User ( default )**

Username: proiect

Password: proiect

Bibliografie

[Baze de date 1](#_Toc166539057)

[1. Descrierea modelului real, a utilităţii acestuia şi a regulilor de funcţionare. 5](#_Toc166539059)

[2. Prezentarea constrângerilor ( restricţii, reguli ) impuse asupra modelului. 5](#_Toc166539060)

[3. Descrierea entităţilor, incluzând precizarea cheii primare. 5](#_Toc166539061)

[4. Descrierea relaţiilor, incluzând precizarea cardinalităţii acestora. 5](#_Toc166539062)

[5. Descrierea atributelor, incluzând tipul de date şi evemtiaşeşe constrângeri, valori implicite, valori posibile ale atributelor. 5](#_Toc166539063)

[6. Realizarea diagramei entitate-relaţie corespunzătoare descrierii de la punctele 3-5 5](#_Toc166539064)

[7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relaţie proiectate la punctul 6. Diagrama conceptuală obţinută treubie să conţină minim 7 tabele ( fără considerarea subentităţilor), dintre cel puţin un table asociativ. 5](#_Toc166539065)

[8. Enumerarea schemelor relaţionale corespunzătoare diagramei conceptuale proiectate la punctul 7. 5](#_Toc166539066)

[9. Realizarea normalizării până la forma normală 3 (FN1-FN3). 5](#_Toc166539067)

[10. Crearea unei secvenţe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). 5](#_Toc166539068)

[11. Crearea tabelelor în SQL şi inserarea de date coerente în fiecare dintre acestea ( minim 5 înregistrări în fiecare tabel neasociativ; minim 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel). 5](#_Toc166539069)

[12. Formulaţi în limbaj natural şi implementaţi 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: 6](#_Toc166539070)

[13. Implementarea a 3 operaţii de actulizare şi de suprimare a datelor utilizând subcereri. 6](#_Toc166539078)

[14. Crearea unei vizualizări complexe. Daţi un exemplu de operaţie LMD permisă pe vizualizarea respectivă şi un exemplu de operaţie LMD nepermisă. 6](#_Toc166539079)

[15. Formulaţi în limbaj natural şi implementaţi în SQL: o cerere ce utilizează operaţia outer-join pe minim 4 tabele, o cerere ce utilizează operaţia division şi o cerere care implementează analiza top-n. 6](#_Toc166539080)

[16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietăţile operatorilor algebrei relaţionale. Cererea va fi exprimată prin expresie algebrică, arbore algebric şi limbaj (SQL), atât anterior cât şi ulterior optimizării. 6](#_Toc166539082)

[17. a) Realizarea normalizării BCNF, FN4, FN5. b) Aplicarea denormalizării, justificând necesitatea acesteia. 6](#_Toc166539083)

[Concluzii 8](#_Toc166539088)

### Descrierea modelului real, a utilităţii acestuia şi a regulilor de funcţionare.

**Modelul real:** GadgetHaven este un site de comerț electronic specializat în vânzarea de electronice și accesorii pentru electronice. Baza de date a acestui site trebuie să gestioneze informațiile despre produse, clienți, comenzi, stocuri și alte aspecte necesare pentru funcționarea magazinului online.

**Utilitatea:** Baza de date va permite gestionarea eficientă a stocurilor, urmărirea comenzilor, administrarea informațiilor despre clienți și furnizarea de rapoarte pentru analiza vânzărilor și a performanțelor produselor.

**Reguli de funcționare:**

* Fiecare produs trebuie să aibă un stoc asociat.
* Fiecare comandă trebuie să fie asociată unui client.
* Stocurile trebuie actualizate la fiecare achiziție.
* Produsele pot avea mai multe accesorii.
* Prețurile și descrierile produselor trebuie să fie precise și actualizate.

### Prezentarea constrângerilor ( restricţii, reguli ) impuse asupra modelului.

**Constrangeri:**

* Chei primare unice pentru identificarea clară a fiecărei entități.
* Chei străine pentru menținerea integrității referențiale între tabele.
* Constrângeri de unicitate pentru anumite câmpuri (de exemplu, adresele de e-mail ale clienților).
* Constrângeri CHECK pentru validarea datelor (de exemplu, prețuri pozitive).
* Valori implicite pentru anumite câmpuri (de exemplu, data comenzii implicită să fie data curentă).

### Descrierea entităţilor, incluzând precizarea cheii primare.

|  |  |  |
| --- | --- | --- |
| **ENTITATE** | **CHEIE PRIMARA** | **OBSERVATII** |
| Client | client\_id | Informatii despre client ( nume, prenume,email) |
| Produs | produs\_id | Informatii despre produs ( descriere, pret, categorie ) |
| Categorie | categorie\_id | Numele categoriei |
| Furnizori | furnizor\_id | Informatii despre furnizor ( nume, adresa/contact ) |
| Stoc | stoc\_id | Informatii despre stoc ( cantitate, data ultimei aprovizionari ) |
| Comanda | comanda\_id | Informatii despre comanda ( data, total ) |
| DetaliiComanda | detaliucomanda\_id | Detalii legate de comanda ( cantitate, pret ) |
| Aprovizionari | aprovizionare\_id | Informatii despre aprovizionari ( cantitate, data ) |

### Descrierea relaţiilor, incluzând precizarea cardinalităţii acestora.

|  |  |  |
| --- | --- | --- |
| RELATIE | CARDINALITATE | OBSERVATII |
| are | Client – comanda ( one to many )  Categorie – produs ( one to many )  Produs – stoc ( one to one ) | Un client poate avea mai multe comenzi  O categorie poate avea mai multe produse  Un produs are un singur stoc |
| apartine | DetaliiComanda – Produs ( many to one ) | Fiecare detaliu de comanda apartine unui singur produs |
| furnizeaza | Furnizor – produs ( one to many ) | Un furnizor poate furniza mai multe produse |

### Descrierea atributelor, incluzând tipul de date şi eventualele constrângeri, valori implicite, valori posibile ale atributelor.

**ENTITATE: CLIENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **client\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Client”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume | VARCHAR |  |  | Numele clientului, NOT NULL |
| prenume | VARCHAR |  |  | Prenumele clientului, NOT NULL |
| email | VARCHAR |  |  | Emailul clientului, NOT NULL, UNIQUE |
| telefon | VARCHAR |  |  | Numarul de telefon al clientului, NOT NULL |
| adresa | VARCHAR |  |  | Adresa clientului, NOT NULL |

**ENTITATE: PRODUS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **produs\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Produs”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume | VARCHAR |  |  | Numele produsului, NOT NULL |
| descriere | VARCHAR |  |  | Descrierea produsului, NULL |
| pret | NUMBER |  |  | Pretul produsului, NOT NULL |
| categorie\_id | NUMBER |  |  | ID-ul categoriei asociat produsului, FOREIGN KEY, NOT NULL |

**ENTITATE: CATEGORIE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **categorie\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Categorie”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_categorie | VARCHAR |  |  | Numele categoriei de produse, NOT NULL |

**ENTITATE: FURNIZOR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **Furnizor\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Furnizori”, PRIMARY KEY, AUTO\_INCREMENT** |
| nume\_furnizor | VARCHAR |  |  | Numele furnizorului, NOT NULL |
| contact | VARCHAR |  |  | Datele de contact ale furnizorului, NOT NULL |

**ENTITATE: STOC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **stoc\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Stoc”, PRIMARY KEY, AUTO\_INCREMENT** |
| produs\_id | NUMBER |  |  | ID-ul produsului asociat stocului, FOREIGN KEY, NOT NULL |
| cantitate | NUMBER |  |  | Cantiatea stocului, NOT NULL |
| data\_stoc | DATE |  |  | Data ultimei aprovizionari, NOT NULL |

**ENTITATE: COMANDA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **comanda\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Comanda”, PRIMARY KEY, AUTO\_INCREMENT** |
| client\_id | NUMBER |  |  | ID-ul clientului asociat comenzii, FOREIGN KEY, NOT NULL |
| data\_comanda | DATE |  |  | Data cand s-a facut comanda, NOT NULL |
| total | NUMBER |  |  | Totalul de plata al comenzii, NOT NULL |

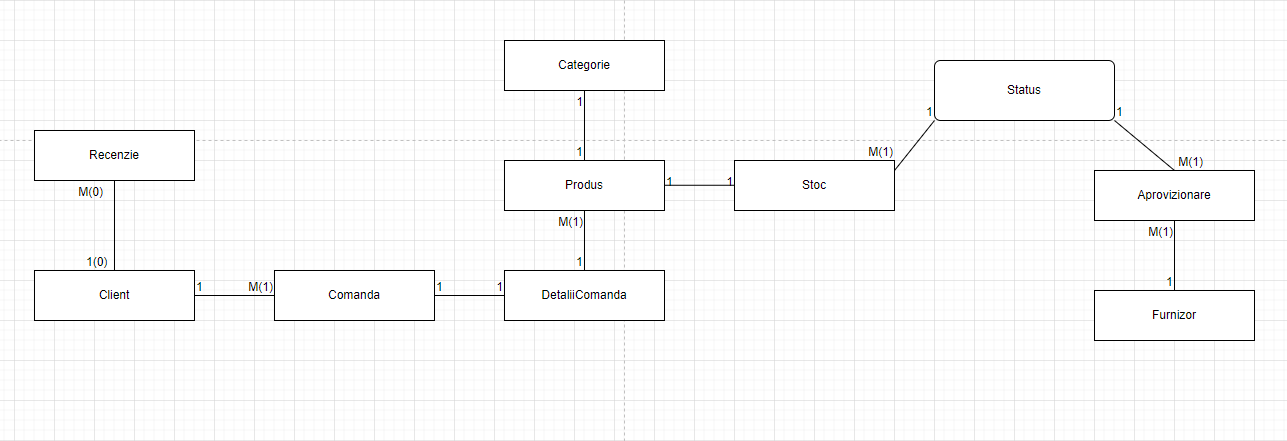
**ENTITATE: DETALIICOMANDA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **detalii\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „DetaliiComanda”, PRIMARY KEY, AUTO\_INCREMENT** |
| comanda\_id | NUMBER |  |  | FOREIN KEY, NOT NULL |
| produs\_id | NUMBER |  |  | FOREIN KEY, NOT NULL |
| cantitate | NUMBER |  |  | NOT NULL |
| pret\_unitar | NUMBER |  |  | NOT NULL |

**ENTITATE: APROVIZIONARE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune/precizie** | **Valori posibile si valori default** | **Observatii, obligatoriu/optional** |
| **aprovizionare\_id** | **NUMBER** |  |  | **Cheia de identificare unica a entitatii „Aprovizionare”, PRIMARY KEY, AUTO\_INCREMENT** |
| produs\_id | NUMBER |  |  | FOREIGN KEY, NOT NULL |
| furnizor\_id | NUMBER |  |  | FOREIGN KEY, NOT NULL |
| cantiate\_aprovizionare | NUMBER |  |  | NOT NULL |
| data\_aprovizionare | DATE |  |  | NOT NULL |

### Realizarea diagramei entitate-relaţie corespunzătoare descrierii de la punctele 3-5.



### Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relaţie proiectate la punctul 6. Diagrama conceptuală obţinută treubie să conţină minim 7 tabele ( fără considerarea subentităţilor), dintre cel puţin un table asociativ.

### Enumerarea schemelor relaţionale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Client(ClientID, Nume, Prenume, Email, Telefon, Adresă)

Produs(ProdusID, Nume, Descriere, Preț, CategorieID, FurnizorID)

Comandă(ComandaID, ClientID, DataComenzii, Total)

DetaliiComandă(DetaliiComandaID, ComandaID, ProdusID, Cantitate, PrețUnitar)

Stoc(ProdusID, CantitateDisponibilă)

Categorie(CategorieID, NumeCategorie)

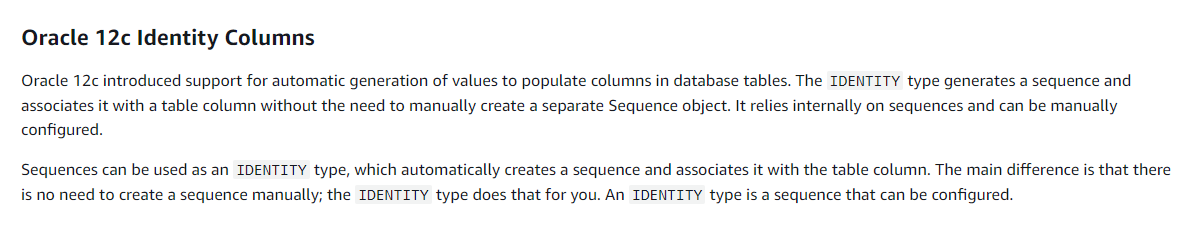
Furnizor(FurnizorID, NumeFurnizor, ContactFurnizor)

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

1. **FN1**: Toate tabelele sunt în FN1 deoarece fiecare atribut conține valori atomice.
2. **FN2**: Toate tabelele sunt în FN2 deoarece toate atributele non-cheie depind complet de cheia primară.
3. **FN3**: Toate tabelele sunt în FN3 deoarece nu există dependențe tranzitive între atributele non-cheie.

### Crearea unei secvenţe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

Pentru secvenţe am folosit **IDENTITY** – genereaza o secvenţă asociată coloanei fără a mai fi nevoie sa fie creat un obiect secvenţă separat.



CREATE TABLE Client (

    client\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY

CREATE TABLE Produs (

    produs\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Furnizor (

    furnizor\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Stoc (

    stoc\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY

CREATE TABLE Comanda (

    comanda\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Detalii\_Comanda (

    detalii\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

CREATE TABLE Aprovizionare (

    aprovizionare\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

### Crearea tabelelor în SQL şi inserarea de date coerente în fiecare dintre acestea ( minim 5 înregistrări în fiecare tabel neasociativ; minim 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

-- INSEREAZA DATE IN TABELE ( etapa 11 )

SET SCAN OFF

INSERT INTO Client(nume\_client, prenume\_client, parola, email, telefon, adresa, cont\_premium)

SELECT \*

FROM

(

    SELECT 'Popescu', 'Ion', 'somepassword', 'ion.popescu@example.com', '0712345678', 'Str. Libertatii, nr. 1', 'nu' FROM DUAL UNION ALL

    SELECT 'Ionescu', 'Maria', 'chinchilla', 'maria.ionescu@example.com', '0723456789', 'Str. Independentei, nr. 10', 'nu' FROM DUAL UNION ALL

    SELECT 'Georgescu', 'Andrei', 'masterpassword', 'andrei.georgescu@example.com', '0734567890', 'Str. Victoriei, nr. 23', 'da' FROM DUAL UNION ALL

    SELECT 'Dumitrescu', 'Elena', 'P@ssw0rd!G5hVq#1K8j2Rz&7Tc', 'elena.dumitrescu@example.com', '0745678901', 'Str. Unirii, nr. 56', 'da' FROM DUAL UNION ALL

    SELECT 'Stanescu', 'Mihai', 'X9f#T@lM5$e@uY1W7Q8g\*4rI2v', 'mihai.stanescu@example.com', '0756789012', 'Str. Revolutiei, nr. 78', 'da' FROM DUAL UNION ALL

    SELECT 'Popa', 'Ana', 'b8@Q\*1N6yT#4vWpZ5$hJ2!rX7kE', 'ana.popa@example.com', '0767890123', 'Str. Eroilor, nr. 12', 'nu' FROM DUAL UNION ALL

    SELECT 'Tudor', 'Florin', 'Y\*8hT1@eG2$9Rz6wQ#5@v3LpI4U', 'florin.tudor@example.com', '0778901234', 'Str. Primaverii, nr. 34', 'nu' FROM DUAL UNION ALL

    SELECT 'Marin', 'Alina', 'm6@J#1R2\*8yG$7L5W!4fQz3P9Vk', 'alina.marin@example.com', '0789012345', 'Str. Sperantei, nr. 22', 'da' FROM DUAL UNION ALL

    SELECT 'Radu', 'George', 'r1$Z@4G8!k#3Q2L7V6\*9Tp5eWyU', 'george.radu@example.com', '0790123456', 'Str. Pacii, nr. 19', 'nu' FROM DUAL UNION ALL

    SELECT 'Iliescu', 'Simona', 'H5$3@Q1!Rz4v\*2Y6K7pX8T#9gLf', 'simona.iliescu@example.com', '0701234567', 'Str. Prieteniei, nr. 44', 'nu' FROM DUAL UNION ALL

    SELECT 'Vasilescu', 'Lucian', 'v8\*9Q4@2W1#T5$yR3!Lz6p7GmXk', 'lucian.vasilescu@example.com', '0712345679', 'Str. Pacea, nr. 11', 'nu' FROM DUAL UNION ALL

    SELECT 'Petrescu', 'Ioana', 'Q3$z@6R1!L4p\*8G7V5T#2y9kXWe', 'ioana.petrescu@example.com', '0723456790', 'Str. Libertatii, nr. 20', 'nu' FROM DUAL UNION ALL

    SELECT 'Munteanu', 'Stefan', 'Rz6\*4Q1@T5!9L2#3y7G$8vWkpXm', 'stefan.munteanu@example.com', '0734567901', 'Str. Florilor, nr. 33', 'nu' FROM DUAL UNION ALL

    SELECT 'Constantin', 'Laura', 'V5@8Rz4#3Q2!9G1y6L\*7T$pWkXm', 'laura.constantin@example.com', '0745679012', 'Str. Crinului, nr. 45', 'nu' FROM DUAL UNION ALL

    SELECT 'Dragomir', 'Catalin', '2Q5@T1#8v3Rz!4L\*7$9yG6pXWk', 'catalin.dragomir@example.com', '0756789123', 'Str. Bucegi, nr. 66', 'nu' FROM DUAL

);

INSERT INTO Produs (nume\_produs, descriere\_produs, pret\_produs, categorie\_produs, oferta)

SELECT \*

FROM (

    SELECT 'Laptop HP Spectre x360', 'Laptop ultrabook convertibil cu ecran tactil', 6500, 'Electronice', 'nu' FROM DUAL UNION ALL

    SELECT 'iPhone 14 Pro Max', 'Smartphone de ultimă generație cu cameră triplă', 7500, 'Electronice', 'da' FROM DUAL UNION ALL

    SELECT 'Samsung Galaxy Tab S8', 'Tableta performantă cu ecran AMOLED', 4200, 'Electronice', 'nu' FROM DUAL UNION ALL

    SELECT 'Sony WH-1000XM4', 'Căști wireless cu anulare activă a zgomotului', 1200, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'GoPro HERO10', 'Cameră de acțiune cu rezoluție 5K', 3000, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Kindle Paperwhite', 'E-book reader cu ecran iluminat și anti-reflexie', 600, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Apple Watch Series 7', 'Ceas inteligent cu funcții avansate de sănătate', 2200, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Roku Streaming Stick 4K', 'Dispozitiv de streaming media 4K', 400, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'DJI Mavic Air 2', 'Dronă pliabilă cu cameră 4K', 4800, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Logitech MX Master 3', 'Mouse wireless ergonomic pentru productivitate', 450, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Anker PowerCore 20000', 'Baterie externă portabilă cu capacitate mare', 300, 'Accesorii', 'da' FROM DUAL UNION ALL

    SELECT 'Samsung T7 Touch', 'SSD extern cu cititor de amprentă', 900, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Fitbit Charge 5', 'Brățară fitness cu monitorizare a ritmului cardiac', 600, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'JBL Charge 5', 'Boxă portabilă cu bluetooth și funcție powerbank', 700, 'Accesorii', 'nu' FROM DUAL UNION ALL

    SELECT 'Oculus Quest 2', 'Set VR autonom cu controlere touch', 2500, 'Accesorii', 'da' FROM DUAL

);

INSERT INTO Furnizor (nume\_furnizor, adresa\_furnizor, email\_furnizor, fax\_furnizor, produs\_furnizat)

SELECT \*

FROM (

    SELECT 'TechDistribution', 'Str. Tehnologiei, nr. 12', 'contact@techdistribution.com', 0211234567, 'Laptopuri' FROM DUAL UNION ALL

    SELECT 'MobileWorld', 'Str. Mobilei, nr. 8', 'sales@mobileworld.com', 0212345678, 'Smartphone-uri' FROM DUAL UNION ALL

    SELECT 'TabletHub', 'Str. Tabletelor, nr. 22', 'info@tablethub.com', 0213456789, 'Tablete' FROM DUAL UNION ALL

    SELECT 'AudioMasters', 'Str. Audio, nr. 34', 'support@audiomasters.com', 0214567890, 'Căști' FROM DUAL UNION ALL

    SELECT 'CameraExperts', 'Str. Camerei, nr. 18', 'contact@cameraexperts.com', 0215678901, 'Camere de acțiune' FROM DUAL UNION ALL

    SELECT 'EbookReaders', 'Str. Ebook, nr. 45', 'info@ebookreaders.com', 0216789012, 'E-book readere' FROM DUAL UNION ALL

    SELECT 'SmartWatches', 'Str. Ceasului, nr. 3', 'sales@smartwatches.com', 0217890123, 'Ceasuri inteligente' FROM DUAL UNION ALL

    SELECT 'StreamingDevices', 'Str. Streaming, nr. 14', 'support@streamingdevices.com', 0218901234, 'Dispozitive de streaming' FROM DUAL UNION ALL

    SELECT 'DroneZone', 'Str. Dronelor, nr. 27', 'info@dronezone.com', 0219012345, 'Drone' FROM DUAL UNION ALL

    SELECT 'AccessoryCenter', 'Str. Accesoriilor, nr. 9', 'contact@accessorycenter.com', 0210123456, 'Mouse-uri' FROM DUAL UNION ALL

    SELECT 'PowerSolutions', 'Str. Puterii, nr. 5', 'support@powersolutions.com', 0211234568, 'Baterii externe' FROM DUAL UNION ALL

    SELECT 'StoragePro', 'Str. Stocării, nr. 11', 'sales@storagepro.com', 0212345679, 'SSD-uri' FROM DUAL UNION ALL

    SELECT 'FitnessGear', 'Str. Fitness, nr. 7', 'info@fitnessgear.com', 0213456780, 'Brățări fitness' FROM DUAL UNION ALL

    SELECT 'SoundWave', 'Str. Sunetului, nr. 6', 'contact@soundwave.com', 0214567891, 'Boxe portabile' FROM DUAL UNION ALL

    SELECT 'VRInnovations', 'Str. Virtuală, nr. 10', 'sales@vrinnovations.com', 0215678902, 'Seturi VR' FROM DUAL

);

INSERT INTO Stoc (produs\_id, cantitate, data\_ultima\_aprovizionare)

SELECT \*

FROM (

    SELECT 1, 20, TO\_DATE('2024-03-01', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 2, 30, TO\_DATE('2024-03-02', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 3, 25, TO\_DATE('2024-03-03', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 4, 40, TO\_DATE('2024-03-04', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 5, 15, TO\_DATE('2024-03-05', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 6, 35, TO\_DATE('2024-03-06', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 7, 20, TO\_DATE('2024-03-07', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 8, 50, TO\_DATE('2024-03-08', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 9, 10, TO\_DATE('2024-03-09', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 10, 25, TO\_DATE('2024-03-10', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 11, 45, TO\_DATE('2024-03-11', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 12, 30, TO\_DATE('2024-03-12', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 13, 40, TO\_DATE('2024-03-13', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 14, 50, TO\_DATE('2024-03-14', 'YYYY-MM-DD') FROM DUAL UNION ALL

    SELECT 15, 20, TO\_DATE('2024-03-15', 'YYYY-MM-DD') FROM DUAL

);

INSERT INTO Comanda (client\_id, data\_comanda, data\_scadenta, pret\_total, mod\_plata)

SELECT \*

FROM (

    SELECT 1, TO\_DATE('2024-04-01', 'YYYY-MM-DD'), TO\_DATE('2024-05-01', 'YYYY-MM-DD'), 6500, 'Card' FROM DUAL UNION ALL

    SELECT 2, TO\_DATE('2024-04-02', 'YYYY-MM-DD'), TO\_DATE('2024-05-02', 'YYYY-MM-DD'), 7500, 'Card' FROM DUAL UNION ALL

    SELECT 3, TO\_DATE('2024-04-03', 'YYYY-MM-DD'), TO\_DATE('2024-05-03', 'YYYY-MM-DD'), 4200, 'Cash' FROM DUAL UNION ALL

    SELECT 4, TO\_DATE('2024-04-04', 'YYYY-MM-DD'), TO\_DATE('2024-05-04', 'YYYY-MM-DD'), 1200, 'Card' FROM DUAL UNION ALL

    SELECT 5, TO\_DATE('2024-04-05', 'YYYY-MM-DD'), TO\_DATE('2024-05-05', 'YYYY-MM-DD'), 3000, 'Cash' FROM DUAL UNION ALL

    SELECT 6, TO\_DATE('2024-04-06', 'YYYY-MM-DD'), TO\_DATE('2024-05-06', 'YYYY-MM-DD'), 600, 'Card' FROM DUAL UNION ALL

    SELECT 7, TO\_DATE('2024-04-07', 'YYYY-MM-DD'), TO\_DATE('2024-05-07', 'YYYY-MM-DD'), 2200, 'Card' FROM DUAL UNION ALL

    SELECT 8, TO\_DATE('2024-04-08', 'YYYY-MM-DD'), TO\_DATE('2024-05-08', 'YYYY-MM-DD'), 400, 'Cash' FROM DUAL UNION ALL

    SELECT 9, TO\_DATE('2024-04-09', 'YYYY-MM-DD'), TO\_DATE('2024-05-09', 'YYYY-MM-DD'), 4800, 'Card' FROM DUAL UNION ALL

    SELECT 10, TO\_DATE('2024-04-10', 'YYYY-MM-DD'), TO\_DATE('2024-05-10', 'YYYY-MM-DD'), 450, 'Cash' FROM DUAL UNION ALL

    SELECT 11, TO\_DATE('2024-04-11', 'YYYY-MM-DD'), TO\_DATE('2024-05-11', 'YYYY-MM-DD'), 300, 'Card' FROM DUAL UNION ALL

    SELECT 12, TO\_DATE('2024-04-12', 'YYYY-MM-DD'), TO\_DATE('2024-05-12', 'YYYY-MM-DD'), 900, 'Cash' FROM DUAL UNION ALL

    SELECT 13, TO\_DATE('2024-04-13', 'YYYY-MM-DD'), TO\_DATE('2024-05-13', 'YYYY-MM-DD'), 700, 'Card' FROM DUAL UNION ALL

    SELECT 14, TO\_DATE('2024-04-14', 'YYYY-MM-DD'), TO\_DATE('2024-05-14', 'YYYY-MM-DD'), 2500, 'Cash' FROM DUAL UNION ALL

    SELECT 15, TO\_DATE('2024-04-15', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'), 2500, 'Card' FROM DUAL

);

INSERT INTO Detalii\_Comanda (comanda\_id, produs\_id, cantitate, pret\_unitar, pret\_fara\_tva, pret\_cu\_tva, tva, cod\_promotional)

SELECT \*

FROM (

    SELECT 1 AS comanda\_id, 1 AS produs\_id, 1 AS cantitate, 6500 AS pret\_unitar, 5416.67 AS pret\_fara\_tva, 6500 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 1 AS comanda\_id, 4 AS produs\_id, 1 AS cantitate, 1200 AS pret\_unitar, 960 AS pret\_fara\_tva, 1200 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 1 AS comanda\_id, 2 AS produs\_id, 2 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 2 AS comanda\_id, 2 AS produs\_id, 1 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, 'da' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 3 AS comanda\_id, 3 AS produs\_id, 1 AS cantitate, 4200 AS pret\_unitar, 3500 AS pret\_fara\_tva, 4200 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 4 AS comanda\_id, 4 AS produs\_id, 1 AS cantitate, 1200 AS pret\_unitar, 1000 AS pret\_fara\_tva, 1200 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 5 AS comanda\_id, 5 AS produs\_id, 1 AS cantitate, 3000 AS pret\_unitar, 2500 AS pret\_fara\_tva, 3000 AS pret\_cu\_tva, 20 AS tva, 'da' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 6 AS comanda\_id, 6 AS produs\_id, 1 AS cantitate, 600 AS pret\_unitar, 500 AS pret\_fara\_tva, 600 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 7 AS comanda\_id, 7 AS produs\_id, 1 AS cantitate, 2200 AS pret\_unitar, 1833.33 AS pret\_fara\_tva, 2200 AS pret\_cu\_tva, 20 AS tva, 'da' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 8 AS comanda\_id, 8 AS produs\_id, 1 AS cantitate, 400 AS pret\_unitar, 333.33 AS pret\_fara\_tva, 400 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 9 AS comanda\_id, 9 AS produs\_id, 1 AS cantitate, 4800 AS pret\_unitar, 4000 AS pret\_fara\_tva, 4800 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 10 AS comanda\_id, 10 AS produs\_id, 1 AS cantitate, 450 AS pret\_unitar, 375 AS pret\_fara\_tva, 450 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 10 AS comanda\_id, 2 AS produs\_id, 1 AS cantitate, 7500 AS pret\_unitar, 6250 AS pret\_fara\_tva, 7500 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 11 AS comanda\_id, 11 AS produs\_id, 1 AS cantitate, 300 AS pret\_unitar, 250 AS pret\_fara\_tva, 300 AS pret\_cu\_tva, 20 AS tva, 'da' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 12 AS comanda\_id, 12 AS produs\_id, 1 AS cantitate, 900 AS pret\_unitar, 750 AS pret\_fara\_tva, 900 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 13 AS comanda\_id, 13 AS produs\_id, 1 AS cantitate, 600 AS pret\_unitar, 500 AS pret\_fara\_tva, 600 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 14 AS comanda\_id, 14 AS produs\_id, 1 AS cantitate, 700 AS pret\_unitar, 583.33 AS pret\_fara\_tva, 700 AS pret\_cu\_tva, 20 AS tva, 'nu' AS cod\_promotional FROM DUAL UNION ALL

    SELECT 15 AS comanda\_id, 15 AS produs\_id, 1 AS cantitate, 2500 AS pret\_unitar, 2083.33 AS pret\_fara\_tva, 2500 AS pret\_cu\_tva, 20 AS tva, 'da' AS cod\_promotional FROM DUAL

);

INSERT INTO Aprovizionare (produs\_id, furnizor\_id, cantitate, data\_aprovizionare)

SELECT \*

FROM (

    SELECT 1 AS produs\_id, 1 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2024-03-01', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 2 AS produs\_id, 2 AS furnizor\_id, 30 AS cantitate, TO\_DATE('2024-03-02', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 3 AS produs\_id, 3 AS furnizor\_id, 25 AS cantitate, TO\_DATE('2024-03-03', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 4 AS produs\_id, 4 AS furnizor\_id, 40 AS cantitate, TO\_DATE('2024-03-04', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 5 AS produs\_id, 5 AS furnizor\_id, 15 AS cantitate, TO\_DATE('2024-03-05', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 6 AS produs\_id, 6 AS furnizor\_id, 35 AS cantitate, TO\_DATE('2024-03-06', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 7 AS produs\_id, 7 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2024-03-07', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 8 AS produs\_id, 8 AS furnizor\_id, 50 AS cantitate, TO\_DATE('2024-03-08', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 9 AS produs\_id, 9 AS furnizor\_id, 10 AS cantitate, TO\_DATE('2024-03-09', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 10 AS produs\_id, 10 AS furnizor\_id, 25 AS cantitate, TO\_DATE('2024-03-10', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 11 AS produs\_id, 11 AS furnizor\_id, 45 AS cantitate, TO\_DATE('2024-03-11', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 12 AS produs\_id, 12 AS furnizor\_id, 30 AS cantitate, TO\_DATE('2024-03-12', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 13 AS produs\_id, 13 AS furnizor\_id, 40 AS cantitate, TO\_DATE('2024-03-13', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 14 AS produs\_id, 14 AS furnizor\_id, 50 AS cantitate, TO\_DATE('2024-03-14', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL UNION ALL

    SELECT 15 AS produs\_id, 15 AS furnizor\_id, 20 AS cantitate, TO\_DATE('2024-03-15', 'YYYY-MM-DD') AS data\_aprovizionare FROM DUAL

);

### Formulaţi în limbaj natural şi implementaţi 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

* 1. subcereri sincronizate în care intervin cel puțin 3 tabele – done

SELECT cl.nume\_client, cl.prenume\_client, COUNT(dc.produs\_id) AS numar\_produse

FROM Client cl

JOIN Comanda c ON cl.client\_id = c.client\_id

JOIN Detalii\_Comanda dc ON c.comanda\_id = dc.comanda\_id

JOIN Produs p ON dc.produs\_id = p.produs\_id

GROUP BY cl.nume\_client, cl.prenume\_client

HAVING COUNT(DISTINCT p.categorie\_produs) >= 2;

* 1. subcereri nesincronizate în clauza FROM

SELECT p.nume\_produs, comenzi\_totale, venit\_total

FROM Produs p,

    (SELECT produs\_id, COUNT(\*) AS comenzi\_totale, SUM(pret\_cu\_tva) AS venit\_total

     FROM Detalii\_Comanda

     GROUP BY produs\_id

     HAVING COUNT(\*) >= 3 AND SUM(pret\_cu\_tva) > 10000) dc

WHERE p.produs\_id = dc.produs\_id;

* 1. grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

SELECT p.categorie\_produs, cl.nume\_client, SUM(dc.pret\_cu\_tva) AS venit\_total,

       TRUNC(SYSDATE - MAX(c.data\_comanda)) AS zile\_de\_la\_ultima\_comanda,

       CASE

           WHEN TRUNC(SYSDATE - MAX(c.data\_comanda)) <= 30 THEN 'Recent'

           ELSE 'Nu este recent'

       END AS statut\_comanda

FROM Produs p

JOIN Detalii\_Comanda dc ON p.produs\_id = dc.produs\_id

JOIN Comanda c ON dc.comanda\_id = c.comanda\_id

JOIN Client cl ON c.client\_id = cl.client\_id

GROUP BY p.categorie\_produs, cl.nume\_client

HAVING SUM(dc.pret\_cu\_tva) > 7000;

* 1. ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

SELECT c.comanda\_id, c.data\_comanda, NVL(dc.cod\_promotional, 'FARA\_COD') AS cod\_promotional,

       DECODE(c.mod\_plata, 'CC', 'Card Credit', 'PP', 'PayPal', 'NB', 'Net Banking', 'Alta', 'Card', 'Cash') AS mod\_plata,

       UPPER(cl.nume\_client) || ', ' || INITCAP(cl.prenume\_client) AS nume\_complet

FROM Comanda c

JOIN Detalii\_Comanda dc ON c.comanda\_id = dc.comanda\_id

JOIN Client cl ON c.client\_id = cl.client\_id

ORDER BY c.data\_comanda;

* 1. utilizarea a cel puțin 2 funcții pe șiruri de caractere – in 4 si 5
  2. 2 funcții pe date calendaristice – in 3 si 5
  3. a cel puțin unei expresii CASE

SELECT c.comanda\_id, c.data\_comanda, NVL(dc.cod\_promotional, 'FARA\_COD') AS cod\_promotional,

       DECODE(c.mod\_plata, 'CC', 'Card Credit', 'PP', 'PayPal', 'NB', 'Net Banking', 'Alta', 'Card', 'Cash') AS mod\_plata,

       UPPER(cl.nume\_client) || ', ' || INITCAP(cl.prenume\_client) AS nume\_complet

FROM Comanda c

JOIN Detalii\_Comanda dc ON c.comanda\_id = dc.comanda\_id

JOIN Client cl ON c.client\_id = cl.client\_id

ORDER BY c.data\_comanda;

* 1. utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

WITH Produse\_Comandate AS (

    SELECT produs\_id, COUNT(\*) AS numar\_comenzi, MIN(c.data\_comanda) AS prima\_comanda

    FROM Detalii\_Comanda dc

    JOIN Comanda c ON dc.comanda\_id = c.comanda\_id

    GROUP BY produs\_id

    HAVING COUNT(\*) > 2

)

SELECT p.nume\_produs, pc.numar\_comenzi,

       TRUNC(SYSDATE - pc.prima\_comanda) AS zile\_de\_la\_prima\_comanda,

       LOWER(p.descriere\_produs) AS descriere\_produs\_mic

FROM Produs p

JOIN Produse\_Comandate pc ON p.produs\_id = pc.produs\_id;

Observaţie: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus astfel încât cele 5 cereri să cuprindă pe toate.

### Implementarea a 3 operaţii de actulizare şi de suprimare a datelor utilizând subcereri.

### Crearea unei vizualizări complexe. Daţi un exemplu de operaţie LMD permisă pe vizualizarea respectivă şi un exemplu de operaţie LMD nepermisă.

### Formulaţi în limbaj natural şi implementaţi în SQL: o cerere ce utilizează operaţia outer-join pe minim 4 tabele, o cerere ce utilizează operaţia division şi o cerere care implementează analiza top-n.

Observatie: Cele 3 cereri sunt diferite de cererile de la exerciţiul 12.

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

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

# Concluzii

[1] Donald E. Knuth, „Structured Programming with *Go to* Statements”, în *ACM Com- put. Surv.* 6.4 (Dec. 1974), pp. 261–301, issn: 0360-0300, doi: [10.1145/356635.](https://doi.org/10.1145/356635.356640) [356640](https://doi.org/10.1145/356635.356640), url: <https://doi.org/10.1145/356635.356640>.