Sima Mihai, grupa 241

**Proiect Baze de Date**

-primul fisier-

**1) Descrierea modelului real, a utilitatii acestuia si a regulilor de functionare**

Am conceptualizat urmatoarea baza de date pentru gestionarea unui lant de magazine de electronice. Modelul permite monitorizarea stocurilor, vanzarilor, comenzilor si alocarea responsabilitatilor angajatilor, oferind o imagine clara asupra interactiunilor dintre magazine, produse, furnizori si clienti.

**Principalele tabele din aceasta baza de date sunt:**

* **Magazin** – pentru stocarea informatiilor despre magazine.
* **Raion** – pentru organizarea produselor in sectiuni ale magazinului.
* **Electronic** – pentru produsele destinate vanzarii (telefon, laptop, televizor etc.).
* **Furnizor** – pentru furnizorii care livreaza electronicele.
* **Client** – pentru clientii care plaseaza comenzi.
* **Comanda** – pentru comenzile facute de clienti.
* **Detalii\_Comanda** – tabela asociativa pentru relatia M:M dintre Comanda si Electronic.
* **Angajat** – pentru gestionarea angajatilor magazinelor.

**2) Prezentarea constrângerilor (restrictii, reguli) impuse asupra modelului**

* Un **raion** apartine unui singur **magazin**.
* Un **electronic** provine dintr-un singur **raion**.
* Un **electronic** este asociat unui singur **furnizor** pentru vanzare.
* Un **client** poate plasa mai multe **comenzi**.
* O **comanda** poate contine mai multe **electronice,** iar mai multe **electronice** pot aparea intr-o **comanda**, gestionate prin tabela asociativa **Detalii\_Comanda**.
* Un **magazin** poate avea mai multi **angajati.**

**3) Descrierea entitatilor, incluzand precizarea cheii primare.**

|  |  |  |
| --- | --- | --- |
| Tabel | Cheie primară (PK) | Descriere |
| Magazin | id\_magazin | Magazin de electronice. |
| Raion | id\_raion | Raion dintr-un magazin (ex: televizoare, laptopuri). |
| Electronic | id\_electronic | Produs electronic destinat vânzării (TV, laptop, telefon). |
| Furnizor | id\_furnizor | Furnizor/partener care livrează electronice către magazine. |
| Client | id\_client | Client care face comenzi. |
| Comanda | id\_comanda | Comandă făcută de un client pentru produse electronice. |
| Detalii\_Comanda | id\_detalii | Detaliile unei comenzi (produs, cantitate, firmă de livrare). |
| Angajat | id\_angajat | Angajat care lucrează într-un magazin. |

**4) Descrierea relatiilor, incluzand precizarea cardinalitatii**

|  |  |  |  |
| --- | --- | --- | --- |
| Entitati | Relatie | Cardinalitate | Explicatie |
| Magazin → Raion | are | 1:M | Fiecare magazin are mai multe raioane |
| Raion → Electronic | contine | 1:M | Fiecare raion poate avea mai multe electronice |
| Electronic → Furnizor | provine | M:1 | Fiecare electronic este furnizat de un singur furnizor |
| Client → Comanda | face | 1:M | Un client poate plasa mai multe comenzi |
| Comanda → Electronic | include (via Detalii\_Comanda) | M:M | O comanda poate include mai multe electronice, iar un electronic poate fi in mai multe comenzi |
| Magazin → Angajat | angajeaza | 1:M | Un magazin poate avea mai multi angajati |

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

**MAGAZIN**

* id\_magazin : cheie primara si unica
* nume : numele magazinului
* oras: orasul in care este magazinul

**Observatii:** Stocheaza informatii despre fiecare magazin. Legatura cu RAION si ANGAJAT.

**RAION**

* id\_raion : cheie primara si unica
* id\_magazin : cheie straina catre MAGAZIN
* categorie : tipul produselor din raion (ex. televizoare, laptopuri)

**Observatii:** Fiecare raion apartine unui MAGAZIN si contine ELECTRONICE.

**ELECTRONIC**

* id\_electronic : cheie primara si unica
* id\_raion : cheie straina catre RAION
* id\_furnizor : cheie straina catre FURNIZOR
* nume : modelul produsului
* culoare : culoarea dispozitivului
* cantitate\_stoc: pentru a sti daca sunt destule pentru comanda clientului
* pret\_recomandat : pretul recomandat de vanzare

**Observatii:** Electronicele sunt destinate vanzarii si apartin unui raion si unui furnizor.

**FURNIZOR**

* id\_furnizor : cheie primara si unica
* nume : denumirea furnizorului
* telefon : numar de contact
* mail : adresa de e-mail

**Observatii:** Importa electronice. Legat numai de ELECTRONIC.

**ANGAJAT**

* id\_angajat : cheie primara si unica
* id\_magazin : cheie straina catre MAGAZIN
* nume : numele angajatului
* departament : departamentul in care lucreaza
* data\_angajare : data angajarii
* salariu : salariul angajatului

**Observatii:** Lucreaza la MAGAZIN.

**CLIENT**

* id\_client : cheie primara si unica
* nume : numele clientului
* telefon : numarul de telefon

**Observatii:** Poate plasa mai multe comenzi.

**COMANDA**

* id\_comanda : cheie primara si unica
* id\_client : cheie straina catre CLIENT
* data : data plasarii comenzii

**Observatii:** Fiecare comanda apartine unui client.

**DETALII\_COMANDA**

* id\_comanda : cheie straina catre COMANDA
* id\_electronic : cheie straina catre ELECTRONIC
* cantitate : numarul de unitati comandate
* firma\_livrare : numele firmei de livrare

**Observatii:** **Tabel asociativ(punctul 7)** pentru produsele din comanda si electronicele vandute.

○○

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

A computer screen shot of a computer

AI-generated content may be incorrect.

**7) Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.**

A computer screen shot of a diagram

AI-generated content may be incorrect.

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

* **RAION**(#id\_raion, id\_magazin(FK), categorie)
* **ELECTRONIC**(#id\_electronic, id\_raion(FK), id\_furnizor(FK), nume, culoare, pret\_recomandat,cantitate\_stoc)
* **FURNIZOR**(#id\_furnizor, nume, telefon, mail)
* **ANGAJAT**(#id\_angajat, id\_magazin(FK), nume, departament, data\_angajare, salariu)
* **CLIENT**(#id\_client, nume, telefon)
* **COMANDA**(#id\_comanda, id\_client(FK), data)
* **DETALII\_COMANDA**(#id\_comanda(FK), #id\_electronic(FK), cantitate, firma\_livrare)
* **MAGAZIN**(#id\_magazin, nume, oras)

**9)Realizarea normalizarii pana la forma actuala 3.(FN1, FN2,FN3).**

**FN1 (atomicitate)**

**Tabel non-FN1: DETALII\_COMANDA**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_comanda | id\_electronic | cantitate | firma\_livrare |
| 1 | 1,2 | 2,1 | DHL,DHL |
| 2 | 3 | 1 | FanCourier |
| 3 | 2,3 | 3,1 | GLS,GLS |

* Coloanele id\_electronic, cantitate si firma\_livrare contin mai multe valori într-o singură celulă.
* FN1 cere valori atomice, adică o singură valoare pe celulă.

Tabel normalizat FN1: DETALII\_COMANDA

|  |  |  |  |
| --- | --- | --- | --- |
| id\_comanda | id\_electronic | cantitate | firma\_livrare |
| 1 | 1 | 2 | DHL |
| 1 | 2 | 1 | DHL |
| 2 | 3 | 1 | FanCourier |
| 3 | 2 | 3 | GLS |
| 3 | 3 | 1 | GLS |

**FN2 (dependenta completa de cheia primara)**

**Tabel non-FN2: ELECTRONIC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_electronic | nume | id\_raion | id\_furnizor | pret\_recomandat | cantitate\_stoc |
| 1 | MacBook Pro | 1 | 1 | 12000 | 12 |
| 2 | MacBook Air | 1 | 1 | 9500 | 20 |
| 3 | iPhone 14 | 2 | 2 | 6000 | 3 |
| 4 | iPhone 13 | 2 | 2 | 5000 | 1 |
| 5 | LG OLED 55" | 3 | 3 | 5000 | 5 |
| 6 | LG NanoCell 65" | 3 | 3 | 6000 | 10 |

* id\_furnizor depinde tranzitiv de id\_electronic prin id\_raion → dacă schimbăm furnizorul unui raion, trebuie să modificăm mai multe rânduri.
* Deci există o dependență tranzitivă și tabelul nu respectă FN2 complet, pentru că unele atribute nu depind strict de cheia primară (id\_electronic).

**Tabele normalizate FN2:**

**În FN2 eliminăm dependențele parțiale, păstrând doar atribute care depind strict de cheia primară (id\_electronic). În cazul nostru:**

**ELECTRONIC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_electronic | nume | id\_raion | cantitate\_stoc | pret\_recomandat |
| 1 | MacBook Pro | 1 | 12 | 12000 |
| 2 | MacBook Air | 1 | 20 | 9500 |
| 3 | iPhone 14 | 2 | 3 | 6000 |
| 4 | iPhone 13 | 2 | 1 | 5000 |
| 5 | LG OLED 55" | 3 | 5 | 5000 |
| 6 | LG NanoCell 65" | 3 | 10 | 6000 |

**FURNIZOR**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_furnizor | nume | telefon | mail |
| 1 | SAMSUNG | 0212345678 | [contact@samsung.ro](mailto:contact@samsung.ro) |
| 2 | APPLE | 0218765432 | [contact@apple.ro](mailto:contact@apple.ro) |
| 3 | LG | 0264123456 | [contact@lg.ro](mailto:contact@lg.ro) |

**FN3 (fara dependente tranzitive)**

**Tabel non-FN3: ELECTRONIC**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_electronic | nume | id\_raion | id\_furnizor |
| 1 | A24 | 1 | 1 |
| 2 | S10 | 1 | 1 |
| 3 | X200 | 2 | 2 |

* id\_furnizor depinde tranzitiv de id\_electronic prin id\_raion.
* Daca un raion isi schimba furnizorul, toate electronicele din acel raion ar trebui actualizate.

**Tabele normalizate FN3:**

**ELECTRONIC**

|  |  |  |
| --- | --- | --- |
| id\_electronic | nume | id\_raion |
| 1 | A24 | 1 |
| 2 | S10 | 1 |
| 3 | X200 | 2 |

**RAION**

|  |  |  |
| --- | --- | --- |
| id\_raion | categorie | id\_furnizor |
| 1 | Laptopuri | 1 |
| 2 | Telefoane | 2 |

**10) Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).**

-- Creare secvente noi

CREATE SEQUENCE magazin\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE raion\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE electronic\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE furnizor\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE angajat\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE client\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE comanda\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE detalii\_comanda\_seq START WITH 1 INCREMENT BY 1;

**11) 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).**  
  
--CREAREA TABELELOR

CREATE TABLE MAGAZIN (

id\_magazin INT PRIMARY KEY,

nume VARCHAR2(100),

oras VARCHAR2(50)

);

CREATE TABLE RAION (

id\_raion INT PRIMARY KEY,

id\_magazin INT NOT NULL REFERENCES MAGAZIN(id\_magazin),

categorie VARCHAR2(50)

);

CREATE TABLE FURNIZOR (

id\_furnizor INT PRIMARY KEY,

nume VARCHAR2(100),

telefon VARCHAR2(20),

mail VARCHAR2(100)

);

CREATE TABLE ELECTRONIC (

id\_electronic INT PRIMARY KEY,

id\_raion INT NOT NULL REFERENCES RAION(id\_raion),

id\_furnizor INT NOT NULL REFERENCES FURNIZOR(id\_furnizor),

nume VARCHAR2(100),

culoare VARCHAR2(30),

pret\_recomandat NUMBER

);

CREATE TABLE ANGAJAT (

id\_angajat INT PRIMARY KEY,

id\_magazin INT REFERENCES MAGAZIN(id\_magazin),

nume VARCHAR2(100),

departament VARCHAR2(50),

data\_angajare DATE,

salariu NUMBER

);

CREATE TABLE CLIENT (

id\_client INT PRIMARY KEY,

nume VARCHAR2(100),

telefon VARCHAR2(20)

);

CREATE TABLE COMANDA (

id\_comanda INT PRIMARY KEY,

id\_client INT NOT NULL REFERENCES CLIENT(id\_client),

data DATE

);

CREATE TABLE DETALII\_COMANDA (

id\_detalii INT PRIMARY KEY,

id\_comanda INT NOT NULL REFERENCES COMANDA(id\_comanda),

id\_electronic INT NOT NULL REFERENCES ELECTRONIC(id\_electronic),

cantitate NUMBER,

firma\_livrare VARCHAR2(50)

);

-- MAGAZIN (5)

INSERT INTO MAGAZIN VALUES (1, 'ElectroMall', 'Bucuresti');

INSERT INTO MAGAZIN VALUES (2, 'TechCenter', 'Cluj');

INSERT INTO MAGAZIN VALUES (3, 'Electronix', 'Timisoara');

INSERT INTO MAGAZIN VALUES (4, 'GigaTech', 'Iasi');

INSERT INTO MAGAZIN VALUES (5, 'SmartShop', 'Constanta');

-- RAION (5)

INSERT INTO RAION VALUES (1, 1, 'Laptopuri');

INSERT INTO RAION VALUES (2, 1, 'Telefoane');

INSERT INTO RAION VALUES (3, 2, 'Televizoare');

INSERT INTO RAION VALUES (4, 3, 'Electrocasnice mici');

INSERT INTO RAION VALUES (5, 4, 'Tablete');

-- FURNIZOR (5)

INSERT INTO FURNIZOR VALUES (1, 'SAMSUNG', '0212345678', 'contact@samsung.ro');

INSERT INTO FURNIZOR VALUES (2, 'APPLE', '0218765432', 'contact@apple.ro');

INSERT INTO FURNIZOR VALUES (3, 'LG', '0264123456', 'contact@lg.ro');

INSERT INTO FURNIZOR VALUES (4, 'HP', '0265987456', 'contact@hp.ro');

INSERT INTO FURNIZOR VALUES (5, 'LENOVO', '0219988776', 'contact@lenovo.ro');

-- ELECTRONIC (10)

INSERT INTO ELECTRONIC VALUES (1, 1, 1, 'MacBook Pro', 'Argintiu', 12000,12);

INSERT INTO ELECTRONIC VALUES (2, 2, 2, 'iPhone 14', 'Negru', 6000),3;

INSERT INTO ELECTRONIC VALUES (3, 3, 3, 'LG OLED 55"', 'Gri', 5000,5);

INSERT INTO ELECTRONIC VALUES (4, 1, 4, 'HP Spectre x360', 'Negru', 8500,4);

INSERT INTO ELECTRONIC VALUES (5, 5, 5, 'Lenovo Yoga', 'Argintiu', 7000,0);

INSERT INTO ELECTRONIC VALUES (6, 2, 2, 'iPhone 13', 'Alb', 5000,1);

INSERT INTO ELECTRONIC VALUES (7, 3, 3, 'LG NanoCell 65"', 'Negru', 6000,10);

INSERT INTO ELECTRONIC VALUES (8, 4, 4, 'Mixer Philips', 'Rosu', 350,9);

INSERT INTO ELECTRONIC VALUES (9, 5, 5, 'Lenovo Tab P11', 'Gri', 1500,3);

INSERT INTO ELECTRONIC VALUES (10, 1, 1, 'MacBook Air', 'Aur', 9500,20);

-- ANGAJAT (5)

INSERT INTO ANGAJAT VALUES (1, 1, 'Ion Popescu', 'Vanzari', TO\_DATE('2022-03-15','YYYY-MM-DD'), 3500);

INSERT INTO ANGAJAT VALUES (2, 2, 'Maria Ionescu', 'Marketing', TO\_DATE('2021-07-20','YYYY-MM-DD'), 4000);

INSERT INTO ANGAJAT VALUES (3, 3, 'Rares Georgescu', 'IT', TO\_DATE('2023-01-10','YYYY-MM-DD'), 4500);

INSERT INTO ANGAJAT VALUES (4, 4, 'Ana Marinescu', 'Vanzari', TO\_DATE('2022-11-05','YYYY-MM-DD'), 3800);

INSERT INTO ANGAJAT VALUES (5, 5, 'Dan Popa', 'Logistica', TO\_DATE('2023-02-18','YYYY-MM-DD'), 3700);

-- CLIENT (5)

INSERT INTO CLIENT VALUES (1, 'Daniel Vasile', '0723456789');

INSERT INTO CLIENT VALUES (2, 'Laura Marin', '0741234567');

INSERT INTO CLIENT VALUES (3, 'Andrei Ionescu', '0732123456');

INSERT INTO CLIENT VALUES (4, 'Mihaela Pop', '0765432198');

INSERT INTO CLIENT VALUES (5, 'Radu Georgescu', '0756789123');

-- COMANDA (10)

INSERT INTO COMANDA VALUES (1, 1, TO\_DATE('2025-09-07','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (2, 2, TO\_DATE('2025-09-07','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (3, 3, TO\_DATE('2025-09-06','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (4, 4, TO\_DATE('2025-09-05','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (5, 5, TO\_DATE('2025-09-04','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (6, 1, TO\_DATE('2025-09-03','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (7, 2, TO\_DATE('2025-09-02','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (8, 3, TO\_DATE('2025-09-01','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (9, 4, TO\_DATE('2025-08-31','YYYY-MM-DD'));

INSERT INTO COMANDA VALUES (10, 5, TO\_DATE('2025-08-30','YYYY-MM-DD'));

-- DETALII\_COMANDA (10)

INSERT INTO DETALII\_COMANDA VALUES (1, 1, 1, 1, 'DHL');

INSERT INTO DETALII\_COMANDA VALUES (2, 2, 2, 2, 'Fan Courier');

INSERT INTO DETALII\_COMANDA VALUES (3, 3, 3, 3, 'UPS');

INSERT INTO DETALII\_COMANDA VALUES (4, 4, 4, 4, 'DPD');

INSERT INTO DETALII\_COMANDA VALUES (5, 5, 5, 5, 'DHL');

INSERT INTO DETALII\_COMANDA VALUES (6, 6, 6, 6, 'Fan Courier');

INSERT INTO DETALII\_COMANDA VALUES (7, 7, 7, 7, 'UPS');

INSERT INTO DETALII\_COMANDA VALUES (8, 8, 8, 8, 'DPD');

INSERT INTO DETALII\_COMANDA VALUES (9, 9, 9, 9, 'DHL');

INSERT INTO DETALII\_COMANDA VALUES (10, 10, 10, 10, 'Fan Courier');

A screenshot of a computer

AI-generated content may be incorrect.  
**12) Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

**a) subcereri sincronizate în care intervin cel puțin 3 tabele**

Afișează numele clienților și produsele comandate de aceștia, împreună cu numele furnizorului și magazinul de unde provine produsul.

SELECT c.nume AS client, e.nume AS produs, f.nume AS furnizor, m.nume AS magazin

FROM CLIENT c

JOIN COMANDA co ON c.id\_client = co.id\_client

JOIN DETALII\_COMANDA dc ON co.id\_comanda = dc.id\_comanda

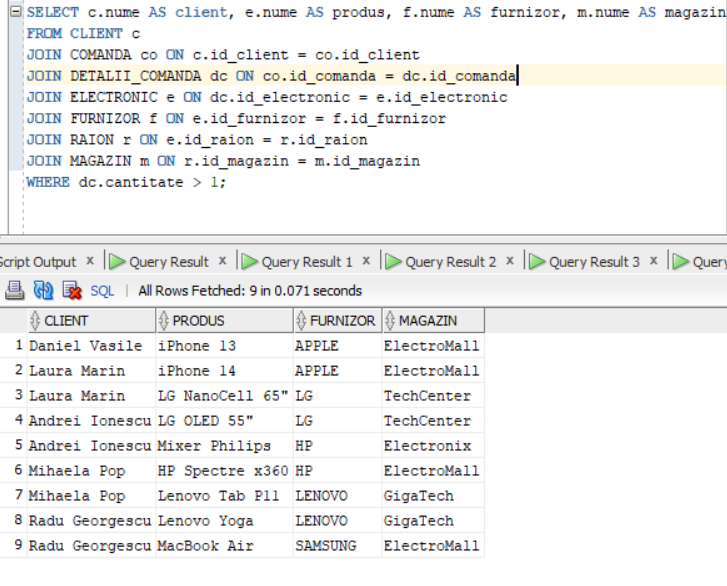
JOIN ELECTRONIC e ON dc.id\_electronic = e.id\_electronic

JOIN FURNIZOR f ON e.id\_furnizor = f.id\_furnizor

JOIN RAION r ON e.id\_raion = r.id\_raion

JOIN MAGAZIN m ON r.id\_magazin = m.id\_magazin

WHERE dc.cantitate > 1;



**b) subcereri nesincronizate în clauza FROM**

Afișează pentru fiecare produs dacă stocul său este mai mare decât media tuturor stocurilor.

SELECT e.nume AS produs, e.cantitate\_stoc,

(SELECT AVG(cantitate\_stoc) FROM ELECTRONIC) AS media\_stocurilor,

CASE

WHEN e.cantitate\_stoc > (SELECT AVG(cantitate\_stoc) FROM ELECTRONIC) THEN 'STOC MARE'

ELSE 'STOC MIC'

END AS situatie\_stoc

FROM ELECTRONIC e;

A screenshot of a computer

AI-generated content may be incorrect.

**c) 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)**

Se cere să se afișeze pentru fiecare magazin totalul produselor vândute (sumă a cantităților din comenzile clienților). Rezultatele trebuie să fie filtrate la nivel de grup folosind clauza HAVING: se vor afișa doar magazinele în care totalul produselor vândute este mai mare decât media vânzărilor pe comenzi.

SELECT m.nume AS magazin, SUM(dc.cantitate) AS total\_vandute

FROM DETALII\_COMANDA dc

JOIN ELECTRONIC e ON dc.id\_electronic = e.id\_electronic

JOIN RAION r ON e.id\_raion = r.id\_raion

JOIN MAGAZIN m ON r.id\_magazin = m.id\_magazin

GROUP BY m.nume

HAVING SUM(dc.cantitate) >

(SELECT AVG(total) FROM (

SELECT SUM(dc2.cantitate) AS total

FROM DETALII\_COMANDA dc2

GROUP BY dc2.id\_comanda

));

A screenshot of a computer

AI-generated content may be incorrect.

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

Afișează comenzile cu numele clientului, data comenzii și firma de livrare. Daca firma nu e specificata, afiseaza NEPRECIZATA. Daca e Fan-Courier, afiseaza LIVRARE RAPIDA. Altfel, afiseaza STANDARD. Ordoneaza descrescator, dupa data comenzii.

SELECT

c.id\_comanda, cl.nume AS nume\_client, c.data,

NVL(dc.firma\_livrare, 'NEPRECIZATA') AS firma\_livrare,

DECODE(dc.firma\_livrare,

'Fan Courier', 'LIVRARE RAPIDA',

'STANDARD') AS tip\_livrare

FROM comanda c

JOIN client cl ON c.id\_client = cl.id\_client

JOIN detalii\_comanda dc ON c.id\_comanda = dc.id\_comanda

ORDER BY c.data DESC;

A screenshot of a computer

AI-generated content may be incorrect.

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

Afisati angajatii care au fost angajati intr-o anumita zi si luna, crea-ti-i email folosind functii, si clasifica-ti-l in functie de vechime.

SELECT

A.id\_angajat, A.nume,

EXTRACT(MONTH FROM SYSDATE) - EXTRACT(MONTH FROM A.data\_angajare)

+ 12 \* (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM A.data\_angajare)) AS luni\_lucrate,

LOWER(A.nume) || '@magazin' || A.id\_magazin || '.com' AS email,

CASE

WHEN MONTHS\_BETWEEN(SYSDATE, A.data\_angajare) >= 24 THEN 'VETERAN'

ELSE 'NOU'

END AS status\_angajat,

SUBSTR(A.nume, 1, 3) AS prefix\_nume

FROM ANGAJAT A

WHERE EXTRACT(DAY FROM A.data\_angajare) = 15

AND EXTRACT(MONTH FROM A.data\_angajare) = 3;

**A screenshot of a computer program

AI-generated content may be incorrect.**

**f)** **Afișează pentru fiecare client numărul total de comenzi și valoarea totală a produselor comandate, filtrând doar clienții cu valoare totală > 10.000.**

WITH total\_comenzi\_client AS (

SELECT

c.id\_client, c.nume AS client,

SUM(e.pret\_recomandat \* dc.cantitate) AS valoare\_totala,

COUNT(DISTINCT cm.id\_comanda) AS nr\_comenzi

FROM CLIENT c

JOIN COMANDA cm ON c.id\_client = cm.id\_client

JOIN DETALII\_COMANDA dc ON cm.id\_comanda = dc.id\_comanda

JOIN ELECTRONIC e ON dc.id\_electronic = e.id\_electronic

GROUP BY c.id\_client, c.nume

)

SELECT \*

FROM total\_comenzi\_client

WHERE valoare\_totala > 10000

ORDER BY valoare\_totala DESC;

A screenshot of a computer

AI-generated content may be incorrect.

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

a) UPDATE – creștem salariul angajaților din magazinele din Cluj

UPDATE ANGAJAT

SET salariu = salariu \* 1.1

WHERE id\_magazin = (

SELECT id\_magazin

FROM MAGAZIN

WHERE oras = 'Cluj'

);

b)UPDATE – schimbăm firma de livrare la detalii\_comanda, pentru comenzile plasate dupa 1 septembrie

UPDATE DETALII\_COMANDA

SET firma\_livrare = 'GLS'

WHERE id\_comanda IN (

SELECT id\_comanda

FROM COMANDA

WHERE data > TO\_DATE('2025-09-01','YYYY-MM-DD')

);

A table with numbers and symbols

AI-generated content may be incorrect.

c)Sa se stearga comenzile mai vechi de o luna care nu contin niciun produs in DETALII\_COMANDA.

DELETE FROM COMANDA c

WHERE NOT EXISTS (

SELECT dc.id\_comanda

FROM DETALII\_COMANDA dc

WHERE dc.id\_comanda = c.id\_comanda

)

AND c.data < ADD\_MONTHS(SYSDATE, -1);

**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.**

**a) OUTER JOIN pe minimum 4 tabele**

Afișați comenzile plasate de clienți, împreună cu detaliile produselor comandate, magazinul și furnizorul fiecărui produs.

SELECT c.id\_comanda, cl.nume AS nume\_client, e.nume AS nume\_produs,

m.nume AS nume\_magazin, f.nume AS furnizor, dc.firma\_livrare

FROM COMANDA c

LEFT OUTER JOIN CLIENT cl ON c.id\_client = cl.id\_client

LEFT OUTER JOIN DETALII\_COMANDA dc ON c.id\_comanda = dc.id\_comanda

LEFT OUTER JOIN ELECTRONIC e ON dc.id\_electronic = e.id\_electronic

LEFT OUTER JOIN RAION r ON e.id\_raion = r.id\_raion

LEFT OUTER JOIN MAGAZIN m ON r.id\_magazin = m.id\_magazin

LEFT OUTER JOIN FURNIZOR f ON e.id\_furnizor = f.id\_furnizor;

**c)TOP-N**

Afișați cele mai scumpe 3 produse electronice din magazin, ordonate descrescător după preț.

SELECT \*

FROM (

SELECT e.id\_electronic, e.nume, e.pret\_recomandat

FROM ELECTRONIC e

ORDER BY e.pret\_recomandat DESC

)

WHERE ROWNUM <= 3;

A screenshot of a computer

AI-generated content may be incorrect.