Proiect Baze de Date 2024

Proiectarea unei baze de date pentru gestiunea unui lanț de magazine cosmetice

De: Irina Coman

Grupa: 133

Cuprins

[**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.** 3](#_Toc168057110)

[**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.** 4](#_Toc168057111)

[**3. Descrierea entităților, incluzând precizarea cheii primare.** 5](#_Toc168057112)

[**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.** 6](#_Toc168057113)

[**5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.** 8](#_Toc168057114)

[**6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.** 12](#_Toc168057115)

[**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.** 13](#_Toc168057116)

[**8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.** 14](#_Toc168057117)

[**9. Realizarea normalizării până la forma normală 3 (FN1-FN3).** 14](#_Toc168057118)

[ Forma normală 1 (FN1) 14](#_Toc168057119)

[ Forma normală 2 (FN2) 15](#_Toc168057120)

[ Forma normală 3 (FN3) 15](#_Toc168057121)

[**10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).** 16](#_Toc168057122)

[**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).** 17](#_Toc168057123)

[**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:** 43](#_Toc168057124)

[**Exercițiul 1:** 44](#_Toc168057125)

[**Exercițiul 2:** 45](#_Toc168057126)

[**Exercițiul 3:** 46](#_Toc168057127)

[**Exercițiul 4:** 47](#_Toc168057128)

[**Exercițiul 5:** 49](#_Toc168057129)

[**13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.** 50](#_Toc168057130)

[**Exercițiul 1:** 50](#_Toc168057131)

[**Exercițiul 2:** 50](#_Toc168057132)

[**Exercițiul 3:** 51](#_Toc168057133)

[**14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.** 52](#_Toc168057134)

[**Operație LMD Permisă** 53](#_Toc168057135)

[**Operație LMD Nepermisă** 53](#_Toc168057136)

[**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.** 54](#_Toc168057137)

[**Outer join:** 54](#_Toc168057138)

[**Division:** 55](#_Toc168057139)

[**Analiza top-n:** 55](#_Toc168057140)

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

În acest paragraf voi prezenta modelul de date pentru un lanț de magazine de cosmetice, utilitățile acestuia și regulile sale de funcționare.

Baza de date conține informații despre magazinele de cosmetice din cadrul acestui lanț, orașele în care sunt amplasate aceste magazine, managerii care le gestionează, produsele vândute, angajații care lucrează în aceste magazine, furnizorii de la care se comandă produsele pentru menținerea stocului, comenzile cu detaliile lor, plasate către furnizori, clienții care achiziționează produse din aceste magazine și vânzările ce au avut loc în cadrul acestor magazine. Scopul creării acestei baze de date este de a putea ține evidența stocurilor de produse, a comenzilor către furnizori, a angajaților și a distribuției acestora în diverse magazine și a vânzărilor efectuate în magazine, cu produsele achiziționate de diverși clienți.

Fiecare magazin are o locație specifică și un program de funcționare și poate avea mai mulți angajați care îndeplinesc diverse funcții, cum ar fi manageri sau vânzători. Fiecare magazin plasează comenzi către furnizori pentru a-și reînnoi stocurile de produse. Aceste comenzi conțin detalii despre produsele comandate, cantitățile dorite și datele de livrare. De asemenea, fiecare magazin contorizează și vânzările efectuate, în acest fel ținând cont și de detaliile tranzacțiilor.

Furnizorii sunt responsabili de livrarea produselor comandate de magazine, iar produsele sunt clasificate în diverse categorii, precum machiaj, îngrijirea pielii, îngrijirea părului etc. Relația dintre comenzi și furnizori este gestionată prin intermediul unui tabel asociativ care permite înregistrarea datelor despre fiecare comandă.

Angajații lucrează în aceste magazine de cosmetice și fiecare are un set specific de responsabilități, cum ar fi interacțiunea directă cu clienții și oferirea de recomandări de produse. Evidența angajaților include informații personale, funcția ocupată, data angajării și managerul magazinul în care lucrează.

Managerii dețin responsabilități de supraveghere și coordonare în cadrul fiecărui magazin. Ei sunt responsabili de operarea zilnică a magazinului, de gestionarea echipei, de monitorizarea vânzărilor și a performanțelor angajaților, precum și de menținerea unui nivel ridicat de satisfacție a clienților. Datele despre manageri includ, pe lângă informațiile generale despre angajați, și specificarea magazinului unde își desfășoară activitatea, contribuind astfel la organizarea și eficiența operațională.

Produsele includ o gamă variată de articole pentru machiaj, îngrijirea pielii, a părului și altele de îngrijire personală. Fiecare produs este catalogat cu detalii esențiale precum un identificator unic, numele produsului, brandul, tipul și prețul.

Comenzile reprezintă solicitările de achiziție de produse plasate de magazine către furnizori pentru a-și reînnoi stocurile. Fiecare comandă include un identificator unic, detalii despre furnizor, data la care a fost plasată comanda, data estimată de livrare și valoarea totală a comenzii. Monitorizarea comenzilor este esențială pentru a asigura că magazinele au stocuri suficiente și pentru a evita situațiile de lipsă de produse.

Vânzările reprezintă tranzacțiile realizate între clienți și vânzători, în cadrul acestor magazine de cosmetice. Fiecare vânzare se deosebește de celelalte prin data în care a fost efectuată, prețul care a fost plătit, produsele cumpărate și vânzătorul care a realizat această operație.

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

Modelul de date respectă anumite restricţii de funcţionare:

* Un magazin are unu sau mai mulți angajați, iar fiecare angajat lucrează într-un singur magazin.
* Un magazin are un singur manager, iar fiecare manager gestionează un singur magazin.
* Un magazin poate da mai multe comenzi, iar fiecare comandă aparține unui singur magazin.
* Un oraș poate avea sau nu magazine, iar fiecare magazin se situează într-un singur oraș.
* Un manager conduce unu sau mai mulți angajați, iar fiecare angajat este sub conducerea unui singur manager.
* Un angajat efectuează mai multe vânzări mai multor clienți, iar o vânzare este efectuată de un singur angajat unui singur client.
* Un client poate fi implicat într-o vânzare sau nu, adică poate să cumpere ceva din magazin sau nu.
* Un furnizor primește una sau mai multe comenzi, iar fiecare comandă este plasată unui singur furnizor.
* O comandă conține mai multe produse de același fel.
* O vânzare implică cel puțin un produs.

# **3. Descrierea entităților, incluzând precizarea cheii primare.**

Pentru modelul de date referitor la gestiunea unui lanț de magazine de cosmetice, entitățile sunt: **MAGAZIN**, **ANGAJAT**, **MANAGER**, **ORAȘ**, **COMANDĂ**, **FURNIZOR**, **PRODUS**, **VÂNZARE** și **CLIENT**.

În această secțiune, vor fi prezentate aceste entități, dând o descriere completă și menționând cheia primară a fiecăreia.

Toate entităţile care vor fi prezentate sunt independente, cu excepţia entităţilor dependente **RECEPȚIONEAZĂ**, **LUCREAZĂ** și **VINDE**.

MAGAZIN = locul fizic de unde se pot achiziționa produse pentru machiaj și îngrijire personală. Fiecare magazin are angajați, manager și poate plasa comenzi către furnizori pentru a-și reînnoi stocurile de produse. Cheia primară a acestei entități este **ID\_Magazin**.

ANGAJAT = persoana care lucrează în unul dintre magazinele de cosmetice și este responsabilă de interacțiunea cu clienții și procesarea vânzărilor. Cheia primară a acestei entități este **ID\_Angajat**.

MANAGER = persoana care are rolul de supraveghere și de coordonare în cadrul unuia dintre magazine. Managerul este responsabil de operarea zilnică a magazinului, gestionarea echipei și monitorizarea vânzărilor și performanțelor angajaților. Cheia primară a acestei entități este **ID\_Manager**.

ORAȘ = locație geografică în care se află magazinele de cosmetice ale acestui lanț. Fiecare oraș poate găzdui unul sau mai multe magazine. Cheia primară a acestei entități este **ID\_Oraș**.

COMANDĂ = solicitare de achiziție de produse, plasată de magazine către furnizori. Fiecare comandă include detalii despre furnizor, data plasării, data estimată de livrare și valoarea totală și produsele care au fost cumpărate. Comenzile ajută magazinele să mențină stocurile necesare pentru vânzare. Cheia primară a acestei entități este **ID\_Comandă**.

FURNIZOR = entitate care furnizează produsele cosmetice magazinelor. Fiecare furnizor are un set de produse pe care le oferă și primește comenzi de la magazine pentru reînnoirea stocurilor. Datele despre furnizori includ informații de contact și detalii despre produsele furnizate. Cheia primară a acestei entități este **ID\_Furnizor**.

PRODUS = articol cosmetic vândut în magazine, inclusiv machiaj, produs de îngrijire personală. Fiecare produs este catalogat cu un identificator unic, numele produsului, brandul, prețul și stocul. Produsele sunt comandate de magazine pentru a-și menține stocurile. Cheia primară a acestei entități este **ID\_Produs**.

VÂNZARE = tranzacție efectuată în magazinele de cosmetice, în cadrul cărora clienții achiziționează produse. Fiecare vânzare include detalii precum data tranzacției, suma totală, produsele vândute și angajatul care a efectuat vânzarea. Aceste informații sunt esențiale pentru analiza vânzărilor și gestionarea inventarului. Cheia primară a acestei entități este **ID\_Vânzare**.

CLIENT = persoana care achiziționează produse din magazinele de cosmetice. Cheia primară a acestei entități este **ID\_Client**.

RECEPȚIONEAZĂ = procesul de recepționare a unei comenzi plasate de un magazin de către un furnizor. Cheia primară compusă este alcătuită din **ID\_Comandă** și **ID\_Furnizor**.

LUCREAZĂ = activitatea efectuată de un angajat în cadrul unuia dintre magazinele acestui lanț. Cheia primară compusă este alcătuită din **ID\_Angajat** și **ID\_Magazin**.

VINDE = un client cumpără un produs dintr-un magazin, această tranzacție fiind efectuată de unul dintre angajații acelui magazin. Cheia primară compusă este alcătuită din cheile primare **ID\_Angajat**, **ID\_Vânzare** și **ID\_Client**.

# **4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

În continuare, se vor prezenta relațiile dintre entitățile enumerate mai sus și se va da o descriere a acestora. Pentru fiecare relație, se va preciza cardinalitatea minimă și cea maximă.

**MANAGER\_gestionează\_MAGAZIN** = relație care leagă entitățile MANAGER și MAGAZIN, evidențiind legătura dintre acestea (Un magazin are un singur manager, iar fiecare manager gestionează un singur magazin). Relația are cardinalitatea minimă și cea maximă egale cu **1:1**.

**MAGAZIN\_dă\_COMANDĂ** = relație care leagă entitățile MAGAZIN și COMANDĂ, evidențiind legătura dintre acestea (Un magazin poate da mai multe comenzi, iar fiecare comandă aparține unui singur magazin). Relația are cardinalitatea minimă egală cu **1:0** și cardinalitatea maximă egală cu **1:n**.

**MAGAZIN\_se\_află\_în\_ORAȘ** = relație care leagă entitățile MAGAZIN și ORAȘ, evidențiind legătura dintre acestea (Un magazin este situat într-un singur oraș, iar un oraș poate avea sau nu magazine). Relația are cardinalitatea minimă egală cu **0:1** și cea maximă egală cu **n:1**.

**MANAGER\_conduce\_ANGAJAȚI** = relație care leagă entitățile MANAGER și ANGAJAȚI, evidențiind legătura dintre acestea (Un manager conduce unu sau mai mulți angajați, iar fiecare angajat este sub conducerea unui singur manager). Relația are cardinalitatea minimă egală cu **1:1** și cea maximă egală cu **1:n**.

**COMANDĂ\_conține\_PRODUS** = relație care leagă entitățile COMANDĂ și PRODUS, evidențiind legătura dintre acestea (O comandă conține mai multe produse de același tip). Relația are cardinalitatea minimă egală cu **0:n** și cea maximă egală cu **1:n**.

**VÂNZARE\_implică\_PRODUS** = relație care leagă entitățile VÂNZARE și PRODUS, evidențiind legătura dintre acestea (O vânzare implică cel puțin un produs). Relația are cardinalitatea minimă egală cu **0:1** și cea maximă egală cu **1:n**.

**FURNIZOR\_primește\_COMANDĂ** = relație care leagă entitățile FURNIZOR și COMANDĂ, evidențiind legătura dintre acestea (Un furnizor primește una sau mai multe comenzi, iar fiecare comandă este plasată către un singur furnizor). Relația are cardinalitatea minimă egală cu **1:1** și cea maximă egală cu **1:n**. Denumirea acestei relații va fi RECEPȚIONEAZĂ.

**ANGAJAT\_lucrează\_în\_MAGAZIN** = relație care leagă entitățile ANGAJAT și MAGAZIN, evidențiind legătura dintre acestea (Un magazin are unu sau mai mulți angajați, iar fiecare angajat lucrează într-un singur magazin). Relația are cardinalitatea minimă egală cu **1:1** și cardinalitatea maximă egală cu **n:1**. Denumirea acestei relații va fi LUCREAZĂ.

**ANGAJAT\_efectuează\_VÂNZARE\_unui\_CLIENT** = relație care leagă entitățile ANGAJAT, VÂNZARE și CLIENT, evidențiind legătura dintre acestea (Un angajat efectuează mai multe vânzări, iar o vânzare este efectuată de un singur angajat și un client poate cumpăra ceva sau nu). Denumirea acestei relații va fi VINDE.

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

Pentru fiecare entitate a modelului de date se vor prezenta atributele sale și descrierea acestora, incluzând tipul de date și eventualele constrângeri, valori implicite sau valori posibile.

Entitatea **MAGAZIN** are ca atribute:

* **ID\_Magazin** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al magazinului
* **Nume\_Magazin** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele magazinului
* **Adresă\_Magazin** = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa magazinului
* **ID\_Oraș** = variabilă de tip întreg, de lungime maximă 4, care reprezintă identificatorul unic al orașului din care face parte magazinul. Atributul trebuie să corespundă unei valori a cheii primare din tabelul ORAȘ

Entitatea **ANGAJAT** are ca atribute:

* **ID\_Angajat** = variabilă de tip întreg, de lungime maximă 3, care reprezintă identificatorul unic al angajatului
* **Nume\_Angajat** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele angajatului
* **Prenume\_Angajat** = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele angajatului
* **Email\_Angajat** = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa de email a angajatului
* **Dată\_Angajare\_A** = variabilă de tip dată calendaristică și reprezintă data angajării
* **ID\_Manager** = variabilă de tip întreg, de lungime maximă 3, care reprezintă identificatorul unic al managerului angajatului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul MANAGER

Entitatea **MANAGER** are ca atribute:

* **ID\_Manager** = variabilă de tip întreg, de lungime maximă 3, care reprezintă identificatorul unic al managerului
* **Nume\_Manager** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele managerului
* **Prenume\_Manager** = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele managerului
* **Email\_Manager** = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa de email a managerului
* **Dată\_Angajare** = variabilă de tip dată calendaristică și reprezintă data angajării
* **ID\_Magazin** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al magazinului gestionat de manager. Atributul trebuie să corespundă unei valori a cheii primare din tabelul MAGAZIN

Entitatea **ORAȘ** are ca atribute:

* **ID\_Oraș** = variabilă de tip întreg, de lungime maximă 4, care reprezintă identificatorul unic al orașului
* **Nume\_Oraș** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele orașului
* **Regiune** = variabilă de tip caracter, de lungime maximă 100, care reprezintă reguinea în care se află orașul
* **Cod\_Poștal** = variabilă de tip caracter, de lungime maximă 10, care reprezintă codul poștal al orașului

Entitatea **COMANDĂ** are ca atribute:

* **ID\_Comandă** = variabilă de tip întreg, de lungime maximă 10, care reprezintă identificatorul unic al comenzii
* **Dată\_Comandă** = variabilă de tip dată calendaristică și reprezintă data plasării comenzii
* **Dată\_Livrare** = variabilă de tip dată calendaristică și reprezintă data estimată de livrare
* **Valoare\_Totală** = variabilă de tip numeric cu 2 zecimale, de lungime maximă 10, care reprezintă valoarea totală a comenzii
* **ID\_Magazin** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al magazinului care a plasat comanda. Atributul trebuie să corespundă unei valori a cheii primare din tabelul MAGAZIN
* **ID\_Produs** = variabilă de tip întreg, de lungime maximă 6, care reprezintă identificatorul unic al produsului din comandă. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PRODUS

Entitatea **FURNIZOR** are ca atribute:

* **ID\_Furnizor** = variabilă de tip întreg, de lungime maximă 6, care reprezintă identificatorul unic al furnizorului
* **Nume\_Furnizor** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele furnizorului
* **Adresă\_Furnizor** = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa furnizorului
* **Email\_Furnizor** = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa de email a furnizorului
* **Telefon** = variabilă de tip caracter, de lungime maximă 15, care reprezintă numărul de telefon al furnizorului

Entitatea **PRODUS** are ca atribute:

* **ID\_Produs** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al produsului
* **Nume\_Produs** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele produsului
* **Brand** = variabilă de tip caracter, de lungime maximă 100, care reprezintă brandul produsului
* **Preț** = variabilă de tip numeric cu 2 zecimale, de lungime maximă 10, care reprezintă prețul produsului
* **Stoc** = variabilă de tip întreg, de lungime maximă 5, care reprezintă cantitatea disponibilă în stoc

Entitatea **VÂNZARE** are ca atribute:

* **ID\_Vânzare** = variabilă de tip întreg, de lungime maximă 8, care reprezintă identificatorul unic al vânzării
* **Dată\_Vânzare** = variabilă de tip dată calendaristică și reprezintă data efectuării vânzării
* **Sumă\_Totală** = variabilă de tip numeric cu 2 zecimale, de lungime maximă 10, care reprezintă suma totală a vânzării
* **ID\_Produs** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al produsului care a fost vândut. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PRODUS

Entitatea **CLIENT** are ca atribute:

* **ID\_Client** = variabilă de tip întreg, de lungime maximă 4, care reprezintă identificatorul unic al clientului
* **Nume\_Client** = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele clientului
* **Prenume\_Client** = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele clientului

Relația **ANGAJAT\_lucrează\_în\_MAGAZIN** (Entitatea **LUCREAZĂ**) are ca atribute:

* **ID\_Angajat** = variabilă de tip întreg, de lungime maximă 3, care reprezintă identificatorul unic al angajatului care lucrează în magazin. Atributul trebuie să corespundă unei valori a cheii primare din tabelul ANGAJAT
* **ID\_Magazin** = variabilă de tip întreg, de lungime maximă 5, care reprezintă identificatorul unic al magazinului unde lucrează angajatul. Atributul trebuie să corespundă unei valori a cheii primare din tabelul MAGAZIN
* **Poziție** = variabilă de tip caracter, de lungime maximă 50, care reprezintă poziția ocupată de angajat în magazin

Relația **ANGAJAT\_efectuează\_VÂNZARE\_unui\_CLIENT** (Entitatea **VINDE**) are ca atribute:

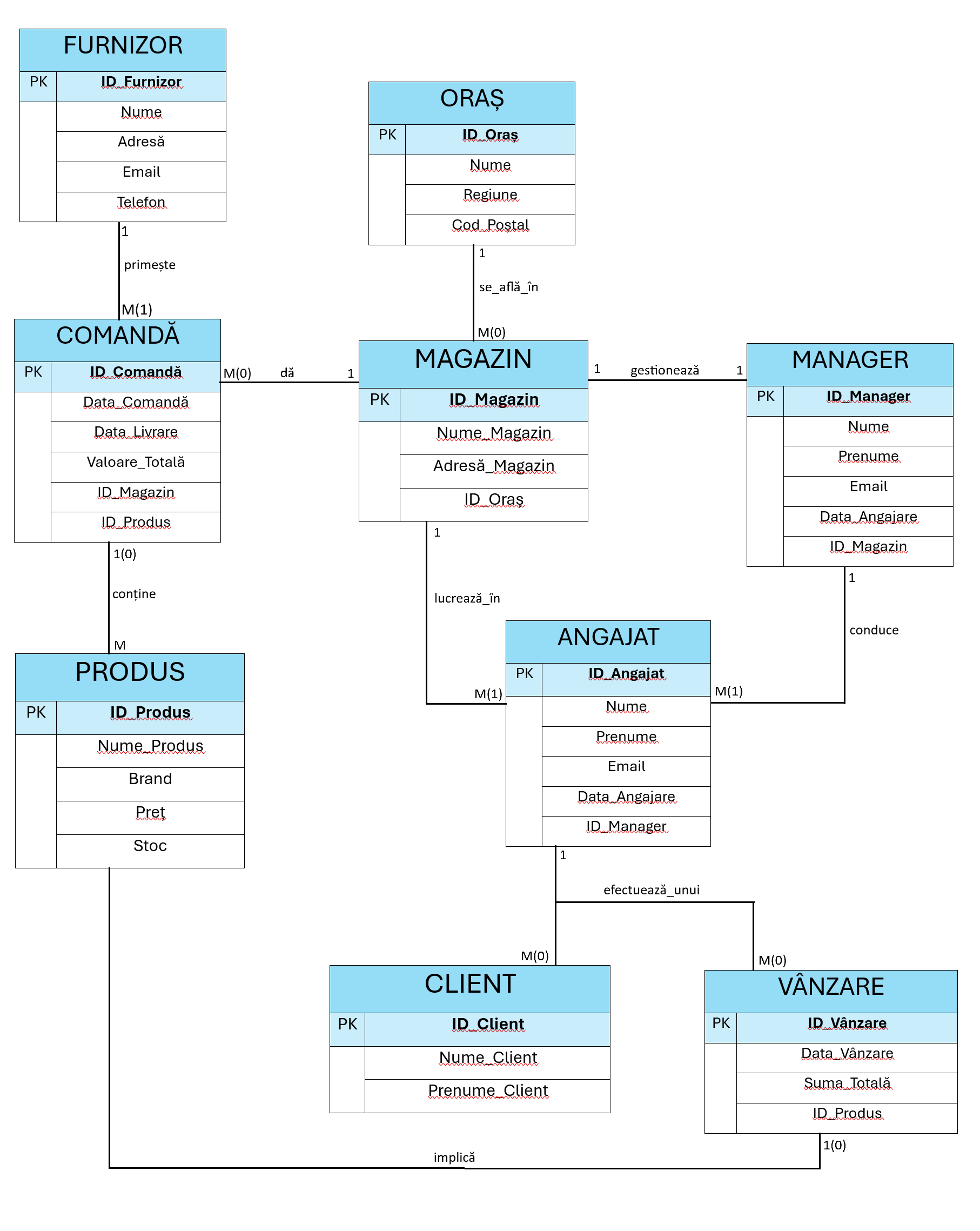
* **ID\_Angajat** = variabilă de tip întreg, de lungime maximă 3, care reprezintă identificatorul unic al angajatului care efectuează vânzarea. Atributul trebuie să corespundă unei valori a cheii primare din tabelul ANGAJAT
* **ID\_Vânzare** = variabilă de tip întreg, de lungime maximă 8, care reprezintă identificatorul unic al vânzării efectuate de angajat. Atributul trebuie să corespundă unei valori a cheii primare din tabelul VÂNZARE
* **ID\_Client** = variabilă de tip întreg, de lungime maximă 4, care reprezintă identificatorul unic al clientului căruia i s-a efectuat vânzarea. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CLIENT

Relația **FURNIZOR\_primește\_COMANDĂ** (Entitatea **RECEPȚIONEAZĂ**) are ca atribute:

* **ID\_Furnizor** = variabilă de tip întreg, de lungime maximă 6, care reprezintă identificatorul unic al furnizorului care primește comanda. Atributul trebuie să corespundă unei valori a cheii primare din tabelul FURNIZOR
* **ID\_Comandă** = variabilă de tip întreg, de lungime maximă 10, care reprezintă identificatorul unic al comenzii primite de furnizor. Atributul trebuie să corespundă unei valori a cheii primare din tabelul COMANDĂ
* **Dată\_Primire** = variabilă de tip dată calendaristică și reprezintă data la care furnizorul a recepționat comanda
* **Stare\_Comandă** = variabilă de tip caracter, de lungime maximă 50, care reprezintă starea comenzii
* **Cantitate** = variabilă de tip întreg, de lungime maximă 3, care reprezintă cantitatea comandată

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

**Diagrama E/R**



# **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 diagram of a company

Description automatically generated

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

Schemele relaţionalecorespunzătoare diagramei conceptuale sunt următoarele:

* **MAGAZIN** (ID\_Magazin#, Nume\_Magazin, Adresă\_Magazin, ID\_Oraș)
* **ANGAJAT** (ID\_Angajat#, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Dată\_Angajare\_A, ID\_Manager)
* **MANAGER** (ID\_Manager#, Nume\_Manager, Prenume\_Manager, Email\_Manager, Dată\_Angajare, ID\_Magazin)
* **ORAȘ** (ID\_Oraș#, Nume\_oraș, Regiune, Cod\_Poștal)
* **COMANDĂ** (ID\_Comandă#, Dată\_Comandă, Dată\_Livrare, Valoare\_Totală, ID\_Magazin, ID\_Produs)
* **FURNIZOR** (ID\_Furnizor#, Nume\_Furnizor, Adresă\_Furnizor, Email\_Furnizor, Telefon)
* **PRODUS** (ID\_Produs#, Nume\_Produs, Brand, Preț, Stoc)
* **VÂNZARE** (ID\_Vânzare#, Dată\_Vânzare, Sumă\_Totală, ID\_Produs)
* **CLIENT** (ID\_Client#, Nume\_Client, Prenume\_Client)
* **RECEPȚIONEAZĂ** (ID\_Furnizor#, ID\_Comandă#, Dată\_Primire, Stare\_Comandă, Cantitate)
* **LUCREAZĂ** (ID\_Angajat#, ID\_Magazin#, Poziție)
* **VINDE** (ID\_Angajat#, ID\_Vânzare#, ID\_Client#)

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

## Forma normală 1 (FN1)

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizabilă. Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

Un exemplu de formă NON-FN1 ar putea apărea în tabelul COMANDĂ, astfel:

|  |  |  |
| --- | --- | --- |
| **ID\_Comandă** | **Dată** | **Valoare\_Totală** |
| 6731983147 | 2022-01-15, 2022-01-20 | 1050.00 |
| 4729864671 | 2022-01-18, 2022-01-23 | 4000 |

Se observă că, în exemplul de mai sus, coloana Dată se poate despărți în două coloane distincte. Modelul corect în forma FN1, ce urmează să fie implementat, este următorul:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID\_Comandă** | **Dată\_Comandă** | **Dată\_Livrare** | **Valoare\_Totală** |
| 6731983147 | 2022-01-15 | 2022-01-20 | 1050.00 |
| 4729864671 | 2022-01-18 | 2022-01-23 | 4000 |

## Forma normală 2 (FN2)

O relație se află în a doua formă normală dacă și numai dacă această relație este deja în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară. FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Un exemplu de formă NON-FN2 ar putea apărea în tabelul VINDE, astfel:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_Angajat** | **ID\_Vânzare** | **ID\_Client** | **Dată\_Vânzare** | **Preț** |
| 101 | 94696243 | 3576 | 2024-03-10 | 195.00 |
| 101 | 55315628 | 6326 | 2024-02-20 | 40.00 |
| 102 | 32153543 | 7983 | 2024-02-10 | 45.75 |

Relația este în FN1 – avem identificator unic pentru toate intrările din tabele.

Se poate observa dependența funcțională între cheile primare ID\_Angajat, ID\_Vânzare și ID\_Client, însă atributele Data\_Vânzare și Preț sunt dependente doar de ID\_Vânzare, ceea ce plasează relația în afara formei FN2. Modelul corect în forma FN2, ce urmează să fie implementat, este următorul:

|  |  |  |
| --- | --- | --- |
| **ID\_Angajat** | **ID\_Vânzare** | **ID\_Client** |
| 101 | 94696243 | 3576 |
| 101 | 55315628 | 6326 |
| 102 | 32153543 | 7983 |

## Forma normală 3 (FN3)

O relație este în a treia formă normală dacă și numai dacă este în FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

Un exemplu de formă NON-FN3 poate apărea în tabelul MAGAZIN, astfel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID\_Magazin** | **Nume\_Magazin** | **Adresă\_Magazin** | **Nume\_Oraș** | **Regiune** | **Cod\_Poștal** |
| 17634 | Sevora Central | Str. Libertății nr. 12 | București | Muntenia | 302813 |
| 28637 | Sevora Nord | Str. Gării nr. 5 | Cluj-Napoca | Transilvania | 481462 |

Se poate observa că atributele Nume\_Oraș, Regiune și Cod\_Poștal nu depind de cheia primară ID\_Magazin. Pentru a aduce în FN3, separ atributele despre oraș din MAGAZIN, apărând tabelul ORAȘ. Astfel, înlocuiesc aceste atribute cu cheia străină ID\_Oraș pentru a determina mai ușor în ce oraș se află un magazin.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID\_Magazin** | **Nume\_Magazin** | **Adresă\_Magazin** | **ID\_Oraș** |
| 17634 | Sevora Central | Str. Libertății nr. 12 | 2793 |
| 28637 | Sevora Nord | Str. Gării nr. 5 | 3292 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID\_Oraș** | **Nume\_Oraș** | **Regiune** | **Cod\_Poștal** |
| 2793 | București | Muntenia | 302813 |
| 3292 | Cluj-Napoca | Transilvania | 481462 |

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

Cod SQL:

CREATE SEQUENCE ORAS\_SEQ START WITH 1;

CREATE SEQUENCE MAGAZIN\_SEQ START WITH 1;

CREATE SEQUENCE ANGAJAT\_SEQ START WITH 1;

CREATE SEQUENCE MANAGER\_SEQ START WITH 1;

CREATE SEQUENCE FURNIZOR\_SEQ START WITH 1;

CREATE SEQUENCE PRODUS\_SEQ START WITH 1;

CREATE SEQUENCE CLIENT\_SEQ START WITH 1;

CREATE SEQUENCE COMANDA\_SEQ START WITH 1;

CREATE SEQUENCE VANZARE\_SEQ START WITH 1;

CREATE SEQUENCE RECEPTIONEAZA\_SEQ START WITH 1;

CREATE SEQUENCE LUCREAZA\_SEQ START WITH 1;

CREATE SEQUENCE VINDE\_SEQ START WITH 1;

Codul sursă:

A screenshot of a computer

Description automatically generated

Rezultatul obținut:

A screenshot of a computer

Description automatically generated A screenshot of a computer screen

Description automatically generated

# **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).**

* **ORAȘ**

CREATE TABLE ORAS(

ID\_Oras NUMBER(5) CONSTRAINT PK\_ORAS PRIMARY KEY,

Nume\_Oras VARCHAR2(100) CONSTRAINT NN\_Nume\_Oras NOT NULL,

Regiune VARCHAR2(100),

Cod\_Postal VARCHAR2(10)

);

INSERT INTO ORAS (ID\_Oras, Nume\_Oras, Regiune, Cod\_Postal) VALUES (2793, 'Bucuresti', 'Muntenia', '302813');

INSERT INTO ORAS (ID\_Oras, Nume\_Oras, Regiune, Cod\_Postal) VALUES (3292, 'Cluj-Napoca', 'Transilvania', '481462');

INSERT INTO ORAS (ID\_Oras, Nume\_Oras, Regiune, Cod\_Postal) VALUES (1761, 'Timisoara', 'Banat', '379430');

INSERT INTO ORAS (ID\_Oras, Nume\_Oras, Regiune, Cod\_Postal) VALUES (3829, 'Iasi', 'Moldova', '705227');

INSERT INTO ORAS (ID\_Oras, Nume\_Oras, Regiune, Cod\_Postal) VALUES (5386, 'Constanta', 'Dobrogea', '947832');

select \* from ORAS;

savepoint save1;

A close-up of a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

* **MAGAZIN**

CREATE TABLE MAGAZIN(

ID\_Magazin NUMBER(5) CONSTRAINT PK\_MAGAZIN PRIMARY KEY,

Nume\_Magazin VARCHAR2(100) CONSTRAINT NN\_Nume\_Magazin NOT NULL,

Adresa\_Magazin VARCHAR2(100),

ID\_Oras NUMBER(5) CONSTRAINT FK\_MAGAZIN\_ORAS REFERENCES ORAS(ID\_Oras)

);

ALTER TABLE MAGAZIN

ADD CONSTRAINT UK\_Nume\_Magazin UNIQUE (Nume\_Magazin);

ALTER TABLE MAGAZIN

DROP CONSTRAINT FK\_MAGAZIN\_ORAS;

ALTER TABLE MAGAZIN

ADD CONSTRAINT FK\_MAGAZIN\_ORAS

FOREIGN KEY (ID\_Oras)

REFERENCES ORAS(ID\_Oras)

ON DELETE SET NULL;

INSERT INTO MAGAZIN (ID\_Magazin, Nume\_Magazin, Adresa\_Magazin, ID\_Oras) VALUES (17634, 'Sevora Central', 'Str. Libertatii nr. 12', 2793);

INSERT INTO MAGAZIN (ID\_Magazin, Nume\_Magazin, Adresa\_Magazin, ID\_Oras) VALUES (28637, 'Sevora Nord', 'Str. Garii nr. 5', 3292);

INSERT INTO MAGAZIN (ID\_Magazin, Nume\_Magazin, Adresa\_Magazin, ID\_Oras) VALUES (34902, 'Sevora Sud', 'Str. Florilor nr. 8', 2793);

INSERT INTO MAGAZIN (ID\_Magazin, Nume\_Magazin, Adresa\_Magazin, ID\_Oras) VALUES (47644, 'Sevora Est', 'Str. Mihai Viteazu nr. 20', 5386);

INSERT INTO MAGAZIN (ID\_Magazin, Nume\_Magazin, Adresa\_Magazin, ID\_Oras) VALUES (52831, 'Sevora Vest', 'Str. Victoriei nr. 15', 2793);

select \* from MAGAZIN;

A screenshot of a computer

Description automatically generated

A close-up of some words

Description automatically generated

A screenshot of a computer

Description automatically generated

* **MANAGER**

CREATE TABLE MANAGER(

ID\_Manager NUMBER(5) CONSTRAINT PK\_MANAGER PRIMARY KEY,

Nume\_Manager VARCHAR2(100) CONSTRAINT NN\_Nume\_Manager NOT NULL,

Prenume\_Manager VARCHAR2(100) CONSTRAINT NN\_Prenume\_Manager NOT NULL,

Email\_Manager VARCHAR2(100) CONSTRAINT NN\_Email\_Manager NOT NULL,

Data\_Angajare DATE CONSTRAINT NN\_Data\_Angajare NOT NULL,

ID\_Magazin NUMBER(5) CONSTRAINT FK\_MANAGER\_MAGAZIN REFERENCES MAGAZIN(ID\_Magazin)

);

ALTER TABLE MANAGER

DROP CONSTRAINT FK\_MANAGER\_MAGAZIN;

ALTER TABLE MANAGER

ADD CONSTRAINT FK\_MANAGER\_MAGAZIN

FOREIGN KEY (ID\_Magazin)

REFERENCES MAGAZIN(ID\_Magazin)

ON DELETE SET NULL;

INSERT INTO MANAGER (ID\_Manager, Nume\_Manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin) VALUES (100, 'Popescu', 'Ion', 'ion.popescu@magazin.ro', DATE '2020-01-01', 47644);

INSERT INTO MANAGER (ID\_Manager, Nume\_Manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin) VALUES (200, 'Ionescu', 'Maria', 'maria.ionescu@magazin.ro', DATE '2020-02-01', 28637);

INSERT INTO MANAGER (ID\_Manager, Nume\_Manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin) VALUES (300, 'Georgescu', 'Andrei', 'andrei.georgescu@magazin.ro', DATE '2020-03-01', 52831);

INSERT INTO MANAGER (ID\_Manager, Nume\_Manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin) VALUES (400, 'Vasilescu', 'Elena', 'elena.vasilescu@magazin.ro', DATE '2020-04-01', 17634);

INSERT INTO MANAGER (ID\_Manager, Nume\_Manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin) VALUES (500, 'Dumitrescu', 'Cristina', 'cristina.dumitrescu@magazin.ro', DATE '2020-05-01', 34902);

select \* from MANAGER;

A screenshot of a computer code

Description automatically generated

A white background with black text

Description automatically generated

A screen shot of a computer

Description automatically generated

* **ANGAJAT**

CREATE TABLE ANGAJAT(

ID\_Angajat NUMBER(5) CONSTRAINT PK\_ANGAJAT PRIMARY KEY,

Nume\_Angajat VARCHAR2(100) CONSTRAINT NN\_Nume\_Angajat NOT NULL,

Prenume\_Angajat VARCHAR2(100) CONSTRAINT NN\_Prenume\_Angajat NOT NULL,

Email\_Angajat VARCHAR2(100) CONSTRAINT NN\_Email\_Angajat NOT NULL,

Data\_Angajare\_A DATE CONSTRAINT NN\_Data\_Angajare\_A NOT NULL,

ID\_Manager NUMBER(5) CONSTRAINT FK\_ANGAJAT\_MANAGER REFERENCES MANAGER(ID\_Manager)

);

ALTER TABLE ANGAJAT DROP CONSTRAINT FK\_ANGAJAT\_MANAGER;

ALTER TABLE ANGAJAT

ADD CONSTRAINT FK\_ANGAJAT\_MANAGER FOREIGN KEY (ID\_Manager)

REFERENCES MANAGER(ID\_Manager) ON DELETE SET NULL;

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(101, 'Mihai', 'Alexandru', 'alexandru.mihai@magazin.ro', DATE '2021-01-14', 100);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(102, 'Popa', 'Roxana', 'roxana.popa@magazin.ro', DATE '2021-02-07', 200);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(103, 'Stan', 'Daniel', 'daniel.stan@magazin.ro', DATE '2021-03-21', 300);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(104, 'Iliescu', 'Ioana', 'ioana.iliescu@magazin.ro', DATE '2021-04-20', 400);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(105, 'Radu', 'Marius', 'marius.radu@magazin.ro', DATE '2021-05-10', 500);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(106, 'Neagu', 'Adriana', 'adriana.neagu@magazin.ro', DATE '2021-06-24', 300);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(107, 'Lazar', 'Cosmin', 'cosmin.lazar@magazin.ro', DATE '2021-07-19', 400);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(108, 'Voicu', 'Gabriel', 'gabriel.voicu@magazin.ro', DATE '2021-08-12', 400);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(109, 'Marin', 'Ana', 'ana.marin@magazin.ro', DATE '2021-09-21', 200);

INSERT INTO ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager) VALUES

(110, 'Constantin', 'Lucian', 'lucian.constantin@magazin.ro', DATE '2021-10-25', 500);

select \* from ANGAJAT;

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screen shot of a computer

Description automatically generated

* **PRODUS**

CREATE TABLE PRODUS(

ID\_Produs NUMBER(5) CONSTRAINT PK\_PRODUS PRIMARY KEY,

Nume\_Produs VARCHAR2(100) CONSTRAINT NN\_Nume\_Produs NOT NULL,

Brand VARCHAR2(100),

Pret NUMBER(10, 2) CONSTRAINT NN\_Pret NOT NULL,

Stoc NUMBER(5) CONSTRAINT NN\_Stoc NOT NULL

);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (15327, 'Fond de ten', 'Estee Lauder', 190.50, 100);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (78432, 'Ruj Color Sensational', 'Maybelline', 40.00, 200);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (31475, 'Mascara Lash Paradise', 'LOreal', 45.75, 150);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (42435, 'Pudra Translucenta', 'Clinique', 130.00, 300);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (53113, 'Paleta de Farduri', 'Urban Decay', 275.00, 250);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (63782, 'Sampon Elseve', 'LOreal', 22.00, 120);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (78245, 'Balsam de Par Hydra Source', 'Biolage', 62.50, 180);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (89213, 'Spray Fixativ Elnett', 'LOreal', 18.00, 140);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (93456, 'Fond de ten Double Wear', 'Estee Lauder', 195.00, 130);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (10234, 'Creion de Ochi Super Liner', 'LOreal', 17.00, 160);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (11324, 'Ruj Matte Ink', 'Maybelline', 35.00, 170);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (12456, 'Paleta de Contur', 'Anastasia Beverly Hills', 245.00, 90);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (13579, 'Ser pentru Par Frizz Ease', 'John Frieda', 125.50, 110);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (14789, 'Mascara Better Than Sex', 'Too Faced', 159.00, 100);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc) VALUES (15987, 'Creion de Sprancene Brow Wiz', 'Anastasia Beverly Hills', 83.00, 130);

select \* from PRODUS;

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

* **COMANDĂ**

CREATE TABLE COMANDA(

ID\_Comanda NUMBER(10) CONSTRAINT PK\_COMANDA PRIMARY KEY,

Data\_Comanda DATE CONSTRAINT NN\_Data\_Comanda NOT NULL,

Data\_Livrare DATE,

Valoare\_Totala NUMBER(10, 2),

ID\_Magazin NUMBER(5) CONSTRAINT FK\_COMANDA\_MAGAZIN REFERENCES MAGAZIN(ID\_Magazin),

ID\_Produs NUMBER(5) CONSTRAINT FK\_COMANDA\_PRODUS REFERENCES PRODUS(ID\_Produs)

);

ALTER TABLE COMANDA

DROP CONSTRAINT FK\_COMANDA\_MAGAZIN;

ALTER TABLE COMANDA

ADD CONSTRAINT FK\_COMANDA\_MAGAZIN FOREIGN KEY (ID\_Magazin)

REFERENCES MAGAZIN(ID\_Magazin)ON DELETE SET NULL;

ALTER TABLE COMANDA

DROP CONSTRAINT FK\_COMANDA\_PRODUS;

ALTER TABLE COMANDA

ADD CONSTRAINT FK\_COMANDA\_PRODUS FOREIGN KEY (ID\_Produs)

REFERENCES PRODUS(ID\_Produs)ON DELETE SET NULL;

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (6731983147, DATE '2022-01-15', DATE '2022-01-20', 1050.00, 28637, 78245);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (4729864671, DATE '2022-01-18', DATE '2022-01-23', 4000.00, 34902, 53113);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (3287346409, DATE '2023-01-21', DATE '2023-01-26', 2362.50, 52831, 89213);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (4682289382, DATE '2023-01-25', DATE '2023-01-30', 9000.00, 52831, 42435);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5643764201, DATE '2023-01-28', DATE '2023-02-02', 6250.00, 47644, 13579);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (6721983197, DATE '2023-02-01', DATE '2023-02-06', 3200.00, 17634, 10234);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5732964683, DATE '2023-02-03', DATE '2023-02-08', 2750.00, 34902, 31475);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (4987346490, DATE '2023-02-05', DATE '2023-02-10', 3400.00, 17634, 78432);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (4182289352, DATE '2023-02-08', DATE '2023-02-13', 2850.00, 34902, 93456);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5343764210, DATE '2023-02-10', DATE '2023-02-15', 1500.00, 28637, 11324);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (6021983147, DATE '2024-02-12', DATE '2024-02-17', 4700.00, 47644, 12456);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5729864672, DATE '2024-02-14', DATE '2024-02-19', 5900.00, 28637, 15327);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5287346410, DATE '2024-02-16', DATE '2024-02-21', 4100.00, 52831, 63782);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (4782289383, DATE '2024-02-18', DATE '2024-02-23', 3500.00, 17634, 14789);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs) VALUES (5643764211, DATE '2024-02-20', DATE '2024-02-25', 4500.00, 17634, 15987);

select \* from COMANDA;

A screenshot of a computer

Description automatically generated

A group of black text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **FURNIZOR**

CREATE TABLE FURNIZOR(

ID\_Furnizor NUMBER(10) CONSTRAINT PK\_FURNIZOR PRIMARY KEY,

Nume\_Furnizor VARCHAR2(100) CONSTRAINT NN\_Nume\_Furnizor NOT NULL,

Adresa\_Furnizor VARCHAR2(200),

Email\_Furnizor VARCHAR2(100),

Telefon VARCHAR2(15)

);

INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

(378261, 'Furnizor Makeup', 'Str. Progresului nr. 134', 'contact@furnizormakeup.ro', '0721378261');

INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

(402864, 'Furnizor Creme Fata', 'Str. Tudor Vladimirescu nr. 29', 'contact@furnizorcremefata.ro', '0721402864');

INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

(729462, 'Furnizor Creme Corp', 'Str. Capitan Aviator Alexandru Serbanescu nr. 50', 'contact@furnizorcremecorp.ro', '0721729462');

INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

(382394, 'Furnizor Produse Par', 'Str. Nicolae Caramfil nr. 25', 'contact@furnizorprodusepar.ro', '0721382394');

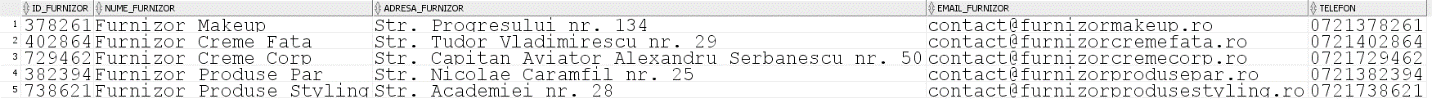
INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

(738621, 'Furnizor Produse Styling', 'Str. Academiei nr. 28', 'contact@furnizorprodusestyling.ro', '0721738621');

select \* from FURNIZOR;

A close up of a text

Description automatically generated



* **VÂNZARE**

CREATE TABLE VANZARE(

ID\_Vanzare NUMBER(10) CONSTRAINT PK\_VANZARE PRIMARY KEY,

Data\_Vanzare DATE CONSTRAINT NN\_Data\_Vanzare NOT NULL,

Suma\_Totala NUMBER(10, 2) CONSTRAINT NN\_Suma\_Totala NOT NULL,

ID\_Produs NUMBER(5) CONSTRAINT FK\_VANZARE\_PRODUS REFERENCES PRODUS(ID\_Produs)

);

ALTER TABLE VANZARE

DROP CONSTRAINT FK\_VANZARE\_PRODUS;

ALTER TABLE VANZARE

ADD CONSTRAINT FK\_VANZARE\_PRODUS FOREIGN KEY (ID\_Produs)

REFERENCES PRODUS(ID\_Produs)ON DELETE SET NULL;

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (17854032, DATE '2024-02-01', 62.50, 78245);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (25684596, DATE '2024-02-05', 18.00, 89213);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (32153543, DATE '2024-02-10', 45.75, 31475);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (48046263, DATE '2024-02-15', 125.50, 13579);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (55315628, DATE '2024-02-20', 40.00, 78432);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (63415764, DATE '2024-02-25', 35.00, 11324);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (74523695, DATE '2024-03-01', 245.00, 12456);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (85780893, DATE '2024-03-05', 22.00, 63782);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (94696243, DATE '2024-03-10', 195.00, 93456);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (10826538, DATE '2024-03-15', 190.50, 15327);

select \* from VANZARE;

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **CLIENT**

CREATE TABLE CLIENT(

ID\_Client NUMBER(5) CONSTRAINT PK\_CLIENT PRIMARY KEY,

Nume\_Client VARCHAR2(100) CONSTRAINT NN\_Nume\_Client NOT NULL,

Prenume\_Client VARCHAR2(100) CONSTRAINT NN\_Prenume\_Client NOT NULL

);

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1965, 'Moldovan', 'Cristian');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (2439, 'Popescu', 'Alina');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (3576, 'Stan', 'Vasile');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (4920, 'Ionescu', 'Mihaela');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (5864, 'Dumitru', 'Andrei');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (6326, 'Georgescu', 'Elena');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (7983, 'Radu', 'Mihai');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (8028, 'Marin', 'Ana');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (9254, 'Pavel', 'Ioan');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1029, 'Barbu', 'Laura');

select \* from CLIENT;

A screenshot of a computer code

Description automatically generated

A screenshot of a table

Description automatically generated

* **RECEPȚIONEAZĂ**

CREATE TABLE RECEPTIONEAZA(

ID\_Furnizor NUMBER(10),

ID\_Comanda NUMBER(10),

Data\_Primire DATE CONSTRAINT NN\_Data\_Primire NOT NULL,

Stare\_Comanda VARCHAR2(50),

Cantitate NUMBER(5),

CONSTRAINT PK\_RECEPTIONEAZA PRIMARY KEY (ID\_Furnizor, ID\_Comanda),

CONSTRAINT FK\_RECEPTIONEAZA\_FURNIZOR FOREIGN KEY (ID\_Furnizor) REFERENCES FURNIZOR(ID\_Furnizor),

CONSTRAINT FK\_RECEPTIONEAZA\_COMANDA FOREIGN KEY (ID\_Comanda) REFERENCES COMANDA(ID\_Comanda)

);

ALTER TABLE RECEPTIONEAZA DROP CONSTRAINT FK\_RECEPTIONEAZA\_FURNIZOR;

ALTER TABLE RECEPTIONEAZA

ADD CONSTRAINT FK\_RECEPTIONEAZA\_FURNIZOR FOREIGN KEY (ID\_Furnizor)

REFERENCES FURNIZOR(ID\_Furnizor) ON DELETE SET NULL;

ALTER TABLE RECEPTIONEAZA DROP CONSTRAINT FK\_RECEPTIONEAZA\_COMANDA;

ALTER TABLE RECEPTIONEAZA

ADD CONSTRAINT FK\_RECEPTIONEAZA\_COMANDA FOREIGN KEY (ID\_Comanda)

REFERENCES COMANDA(ID\_Comanda) ON DELETE SET NULL;

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (729462, 6731983147, DATE '2022-01-15', 'Livrata', 100);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (378261, 4729864671, DATE '2022-01-18', 'Livrata', 200);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (382394, 3287346409, DATE '2023-01-21', 'Livrata', 150);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (402864, 4682289382, DATE '2023-01-25', 'Anulata', 300);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (738621, 5643764201, DATE '2023-01-28', 'Livrata', 250);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (378261, 6721983197, DATE '2023-02-01', 'Livrata', 100);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (729462, 5732964683, DATE '2023-02-03', 'Livrata', 200);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (382394, 4987346490, DATE '2023-02-05', 'Livrata', 150);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (402864, 4182289352, DATE '2023-02-08', 'Anulata', 300);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (738621, 5343764210, DATE '2023-02-10', 'Livrata', 250);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (378261, 6021983147, DATE '2024-02-12', 'Livrata', 100);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (729462, 5729864672, DATE '2024-02-14', 'Anulata', 200);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (402864, 5287346410, DATE '2024-02-16', 'Livrata', 150);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (382394, 4782289383, DATE '2024-02-18', 'Livrata', 250);

INSERT INTO RECEPTIONEAZA (ID\_Furnizor, ID\_Comanda, Data\_Primire, Stare\_Comanda, Cantitate) VALUES (738621, 5643764211, DATE '2024-02-20', 'Anulata', 150);

select \* from RECEPTIONEAZA;

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **LUCREAZĂ**

CREATE TABLE LUCREAZA(

ID\_Angajat NUMBER(5),

ID\_Magazin NUMBER(5),

Pozitie VARCHAR2(100),

CONSTRAINT PK\_LUCREAZA PRIMARY KEY (ID\_Angajat, ID\_Magazin),

CONSTRAINT FK\_LUCREAZA\_ANGAJAT FOREIGN KEY (ID\_Angajat) REFERENCES ANGAJAT(ID\_Angajat),

CONSTRAINT FK\_LUCREAZA\_MAGAZIN FOREIGN KEY (ID\_Magazin) REFERENCES MAGAZIN(ID\_Magazin)

);

ALTER TABLE LUCREAZA DROP CONSTRAINT FK\_LUCREAZA\_ANGAJAT;

ALTER TABLE LUCREAZA

ADD CONSTRAINT FK\_LUCREAZA\_ANGAJAT FOREIGN KEY (ID\_Angajat)

REFERENCES ANGAJAT(ID\_Angajat) ON DELETE SET NULL;

ALTER TABLE LUCREAZA DROP CONSTRAINT FK\_LUCREAZA\_MAGAZIN;

ALTER TABLE LUCREAZA

ADD CONSTRAINT FK\_LUCREAZA\_MAGAZIN FOREIGN KEY (ID\_Magazin)

REFERENCES MAGAZIN(ID\_Magazin) ON DELETE SET NULL;

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (101, 47644, 'Vanzator');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (102, 28637, 'Casier');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (103, 52831, 'Vanzator');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (104, 17634, 'Casier');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (105, 34902, 'Vanzator');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (106, 52831, 'Casier');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (107, 17634, 'Vanzator');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (108, 17634, 'Casier');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (109, 28637, 'Vanzator');

INSERT INTO LUCREAZA (ID\_Angajat, ID\_Magazin, Pozitie) VALUES (110, 34902, 'Casier');

select \* from LUCREAZA;

A screenshot of a computer program

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **VINDE**

CREATE TABLE VINDE(

ID\_Angajat NUMBER(5),

ID\_Vanzare NUMBER(10),

ID\_Client NUMBER(5),

CONSTRAINT PK\_VINDE PRIMARY KEY (ID\_Angajat, ID\_Vanzare, ID\_Client),

CONSTRAINT FK\_VINDE\_ANGAJAT FOREIGN KEY (ID\_Angajat) REFERENCES ANGAJAT(ID\_Angajat),

CONSTRAINT FK\_VINDE\_VANZARE FOREIGN KEY (ID\_Vanzare) REFERENCES VANZARE(ID\_Vanzare),

CONSTRAINT FK\_VINDE\_CLIENT FOREIGN KEY (ID\_Client) REFERENCES CLIENT(ID\_Client)

);

ALTER TABLE VINDE DROP CONSTRAINT FK\_VINDE\_ANGAJAT;

ALTER TABLE VINDE

ADD CONSTRAINT FK\_VINDE\_ANGAJAT FOREIGN KEY (ID\_Angajat)

REFERENCES ANGAJAT(ID\_Angajat) ON DELETE SET NULL;

ALTER TABLE VINDE DROP CONSTRAINT FK\_VINDE\_VANZARE;

ALTER TABLE VINDE

ADD CONSTRAINT FK\_VINDE\_VANZARE FOREIGN KEY (ID\_Vanzare)

REFERENCES VANZARE(ID\_Vanzare) ON DELETE SET NULL;

ALTER TABLE VINDE DROP CONSTRAINT FK\_VINDE\_CLIENT;

ALTER TABLE VINDE

ADD CONSTRAINT FK\_VINDE\_CLIENT FOREIGN KEY (ID\_Client)

REFERENCES CLIENT(ID\_Client) ON DELETE SET NULL;

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (101, 94696243, 3576);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (101, 55315628, 6326);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (102, 32153543, 7983);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (101, 74523695, 1965);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (103, 10826538, 5864);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (109, 48046263, 1029);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (109, 85780893, 8028);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (104, 17854032, 4920);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (104, 25684596, 2439);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (105, 63415764, 9254);

select \* from VINDE;

A screenshot of a computer program

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

# **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

b) subcereri nesincronizate în clauza FROM

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)

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

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

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

**Exercițiul 1:** Să se afișeze codul, numele și prenumele angajatului care a efectuat cele mai multe vânzări. De asemenea, să se afișeze și codul și numele orașului în care se află magazinul în care lucrează acest angajat. Dacă există mai mulți angajați cu același număr de vânzări, se va afișa primul în ordine descrescătoare după codul angajatului.

Rezolvare:

with top\_angajati as (select ID\_Angajat, count(ID\_Vanzare) nr\_vanzari

from VINDE

group by ID\_Angajat

order by 2 desc, 1)

select ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, ID\_Oras, Nume\_Oras

from (select ta.ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, o.ID\_Oras, o.Nume\_Oras, nr\_vanzari

from top\_angajati ta join ANGAJAT a on ta.ID\_Angajat = a.ID\_Angajat

join LUCREAZA l on a.ID\_Angajat = l.ID\_Angajat

join MAGAZIN m on l.ID\_Magazin = m.ID\_Magazin

join ORAS o on m.ID\_Oras = o.ID\_Oras

order by nr\_vanzari desc, ID\_Angajat desc) top\_angajat

where rownum = 1;

A white paper with black text

Description automatically generated



Acest exercițiu respectă următoarele cerințe:

b) subcereri nesincronizate în clauza FROM (from (...) top\_angajat)

c) grupări de date (group by ID\_Angajat) și funcții grup (count)

d) ordonări (order by 2 desc, 1)

f) utilizarea a cel puțin 1 bloc de cerere (with top\_angajati as (...))

**Exercițiul 2:** Să se afișeze codul, numele complet și un mesaj particular al angajaților care au efectuat vânzări până pe data de 15 februarie 2024 și al căror prenume începe cu 'A' și codul, numele complet și un mesaj particular al clienților care au făcut vânzări după data de 15 februarie 2024. La angajați, mesajul particular constă în afișarea sumei totale a vânzării pentru angajații cu nume de familie care se termină în 'n', afișarea produsului vândut pentru angajații cu nume de familie care se termină în 'u', iar -1 în orice alt caz. La clienți, mesajul particular constă în afișarea sumei totale a vânzării pentru clienții cu prenume care se termină în 'a', afișarea produsului vândut pentru clienții cu prenume care se termină în 'n', iar -1 în orice alt caz. Datele vor fi afișate în ordine crescătoare după numele persoanelor.

Rezolvare:

select distinct a.ID\_Angajat, concat(concat(Nume\_Angajat, ' '), Prenume\_Angajat) Nume\_Complet, decode(substr(Nume\_Angajat, -1), 'n', Suma\_totala, 'u', ID\_Produs, nvl(null, -1)) mesaj\_particular

from ANGAJAT a join VINDE v on a.ID\_Angajat = v.ID\_Angajat

join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

where va.Data\_Vanzare <= to\_date('15-FEB-2024', 'dd-mon-yyyy')

and upper(Prenume\_Angajat) like 'A%'

union

select distinct c.ID\_Client, concat(concat(Nume\_Client, ' '), Prenume\_Client) Nume\_Complet, decode(substr(Prenume\_Client, -1), 'a', Suma\_totala, 'n', ID\_Produs, nvl(null, -1)) mesaj\_particular

from CLIENT c join VINDE v on c.ID\_Client = v.ID\_Client

join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

where to\_char(va.Data\_Vanzare, 'dd-MON-yyyy') >= to\_date('15-FEB-2024', 'dd-mon-yyyy')

order by 2;

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

Acest exercițiu respectă următoarele cerințe:

d) ordonări (order by 2) și utilizarea funcțiilor NVL (nvl(null, -1)) și DECODE (decode(substr(Nume\_Angajat, -1), 'n', Suma\_totala, 'u', ID\_Produs, nvl(null, -1)) mesaj\_particular) (în cadrul aceleiași cereri)

e) utilizarea a cel puțin 2 funcții pe șiruri de caractere(concat, upper), o funcție pe date calendaristice(to\_date('15-FEB-2024', 'dd-mon-yyyy'))

**Exercițiul 3:** Să se afișeze numele angajaților și numărul total de vânzări efectuate de aceștia, pentru angajații care au realizat vânzări totale cu o valoare mai mare decât valoarea medie a vânzărilor din toate orașele. Să se ordoneze rezultatele după numărul total de vânzări în ordine descrescătoare. În plus, să se eticheteze angajații în funcție de performanța lor astfel:

* 'Ridicată' pentru cei care au efectuat mai mult de 2 vânzări
* 'Medie' pentru cei cu 2 vânzări
* 'Scăzută' pentru cei cu mai puțin de 2 vânzări.

Rezolvare:

select a.Nume\_Angajat, a.Prenume\_Angajat, count(v.ID\_Vanzare) as Total\_Vanzari, case when count(v.ID\_Vanzare) > 2 then 'Ridicata' when count(v.ID\_Vanzare) = 2 then 'Medie' else 'Scazuta' end as Performanta

from ANGAJAT a join VINDE v on a.ID\_Angajat = v.ID\_Angajat

join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

group by a.Nume\_Angajat, a.Prenume\_Angajat

having sum(va.Suma\_Totala) > (select avg(t.Valoare\_Totala)

from (select sum(va1.Suma\_Totala) as Valoare\_Totala

from VINDE v1 join VANZARE va1 on v1.ID\_Vanzare = va1.ID\_Vanzare

join ANGAJAT a1 on v1.ID\_Angajat = a1.ID\_Angajat

group by v1.ID\_Angajat) t)

order by Total\_Vanzari desc;

A white background with black text

Description automatically generated

A close up of a computer screen

Description automatically generated

Acest exercițiu respectă următoarele cerințe:

b) subcereri nesincronizate în clauza FROM

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)

d) ordonări (order by Total\_Vanzari desc)

e) utilizarea unei expresii CASE

**Exercițiul 4:** Să se afișeze codul, valoarea totală, starea și un mesaj particular al comenzilor care au minimum 6 luni de când au fost plasate și care au fost plasate de un magazin aflat într-un oraș care are cel puțin 2 magazine. Pentru comenzile livrate, mesajul particular constă în afișarea numelui magazinului căruia a fost livrată comanda, iar pentru cele anulate se va afișa numele furnizorului căruia i-a fost plasată comanda.

Rezolvare:

select c.ID\_Comanda, c.Valoare\_Totala, r.Stare\_Comanda, case when r.Stare\_Comanda = 'Livrata' then m.Nume\_Magazin when r.Stare\_Comanda = 'Anulata' then f.Nume\_furnizor end as Nume\_magazin\_furnizor

from COMANDA c join RECEPTIONEAZA r on r.ID\_Comanda = c.ID\_Comanda

join FURNIZOR f on f.ID\_Furnizor = r.ID\_Furnizor

join MAGAZIN m on c.ID\_Magazin = m.ID\_Magazin

join ORAS o on m.ID\_Oras = o.ID\_Oras

where c.Data\_Comanda <= add\_months(sysdate, -6)

and o.ID\_Oras in (select ID\_Oras

from MAGAZIN

group by ID\_Oras

having count(ID\_Magazin) >= 2);

A white background with black text

Description automatically generated

A screen shot of a computer

Description automatically generated

Acest exercițiu respectă următoarele cerințe:

c) grupări de date, funcții grup

e) utilizarea unei funcții pe date calendaristice(add\_months), a unei expresii CASE

**Exercițiul 5:** Să se afișeze codul, data de livrare și valoarea totală a comenzilor date de magazine care au cel puțin 2 angajați și un manager care s-a angajat înainte de 15 martie 2020 și care are litera 'o' în numele de familie.

Rezolvare:

select c.ID\_Comanda, c.Data\_Livrare, c.Valoare\_Totala

from COMANDA c join MAGAZIN m on c.ID\_Magazin = m.ID\_Magazin

where m.ID\_Magazin in (select l.ID\_Magazin

from LUCREAZA l join ANGAJAT a on l.ID\_Angajat = a.ID\_Angajat

join MANAGER mg on a.ID\_Manager = mg.ID\_Manager

where (select count(\*)

from LUCREAZA l2

where l2.ID\_Magazin = l.ID\_Magazin) >= 2

and mg.ID\_Magazin = m.ID\_Magazin

and mg.Data\_Angajare < to\_date('15-03-2020', 'DD-MM-YYYY')

and mg.Nume\_Manager like '%o%')

order by c.Data\_Livrare;

A white screen with black text

Description automatically generated

A screenshot of a calendar

Description automatically generated

Acest exercițiu respectă următoarele cerințe:

a) subcereri sincronizate în care intervin cel puțin 3 tabele

e) utilizarea unei funcții pe date calendaristice(to\_date)

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

**Exercițiul 1:** Să se actualizeze poziția angajaților în tabelul LUCREAZĂ în 'Angajatul lunii' pentru angajații care au suma totală a vânzărilor mai mare decât media vânzărilor tuturor angajaților.

Rezolvare:

update LUCREAZA

set Pozitie = 'Angajatul lunii'

where ID\_Angajat in (select ID\_Angajat

from (select ID\_Angajat, sum(Suma\_Totala) as total\_vanzari

from VINDE v join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

group by ID\_Angajat) totaluri

where total\_vanzari > (select avg(total\_vanzari)

from (select sum(Suma\_Totala) as total\_vanzari

from VINDE v join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

group by ID\_Angajat) medie\_totala));

A screenshot of a computer

Description automatically generated

A close-up of a text

Description automatically generated

**Exercițiul 2:** Să se actualizeze email-urile managerilor cu noul format Nume\_Manager + ID\_Manager + @magazin + prima cifră din ID\_Manager + .ro pentru managerii care gestionează magazine cu cel puțin 2 angajați.

Rezolvare:

update MANAGER

set Email\_Manager = concat(concat(concat(concat(lower(Nume\_Manager), ID\_Manager), '@magazin'), substr(ID\_Manager, 1, 1)), '.ro')

where ID\_Manager in (select m.ID\_Manager

from MANAGER m join MAGAZIN mg on m.ID\_Magazin = mg.ID\_Magazin

join LUCREAZA l on mg.ID\_Magazin = l.ID\_Magazin

group by m.ID\_Manager

having count(l.ID\_Angajat) >= 2);

A white background with black text

Description automatically generated

A black text on a white background

Description automatically generated

**Exercițiul 3:** Să se șteargă orașele în care nu se află niciun magazin.

Rezolvare:

delete from ORAS

where ID\_Oras not in (select ID\_Oras

from MAGAZIN);

rollback to save1;

A screenshot of a computer code

Description automatically generated

A black text on a white background

Description automatically generated

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

Să presupunem că dorim să creăm o vizualizare care afișează informații despre angajații, managerii lor și magazinele în care lucrează, pentru magazinele care se află în orașul București și pentru angajații care au efectuat cel puțin o vânzare.

Cod SQL:

create view ViewMagazin as

select a.ID\_Angajat, a.Nume\_Angajat, a.Prenume\_Angajat, a.Data\_Angajare\_A, a.Email\_Angajat, m.ID\_Manager, m.Nume\_Manager, m.Prenume\_Manager, m.Data\_Angajare, mg.ID\_Magazin, mg.Nume\_Magazin

from ANGAJAT a join LUCREAZA l on a.ID\_Angajat = l.ID\_Angajat

join MAGAZIN mg on l.ID\_Magazin = mg.ID\_Magazin

join ORAS o on mg.ID\_Oras = o.ID\_Oras

join MANAGER m on mg.ID\_Magazin = m.ID\_Magazin

left outer join VINDE v on a.ID\_Angajat = v.ID\_Angajat

join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

where lower(o.Nume\_Oras) = 'bucuresti'

group by a.ID\_Angajat, a.Nume\_Angajat, a.Prenume\_Angajat, a.Data\_Angajare\_A, a.Email\_Angajat, m.ID\_Manager, m.Nume\_Manager, m.Prenume\_Manager, m.Data\_Angajare, mg.ID\_Magazin, mg.Nume\_Magazin

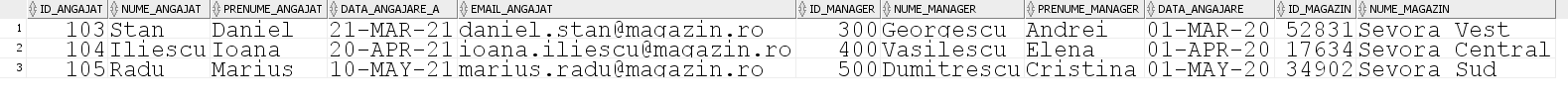
having count(va.ID\_Vanzare) >= 1;

A white background with black text

Description automatically generated

A black text on a white background

Description automatically generated



## **Operație LMD Permisă**

Operația de selecție (SELECT) de mai jos este permisă, deoarece se referă la o coloană care nu implică o agregare sau o relație complexă.

Cod SQL:

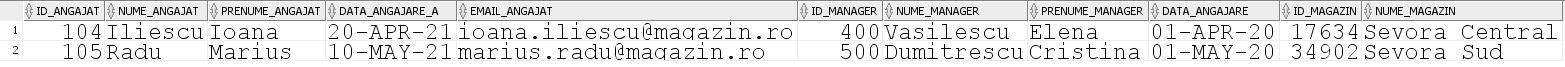
select \*

from ViewMagazin

where lower(Nume\_Angajat) like '%u%';

A close-up of a white background

Description automatically generated



Operația din imagine afișează toate informațiile din ViewMagazin pentru angajații care au litera ‚u’ în numele de familie.

## **Operație LMD Nepermisă**

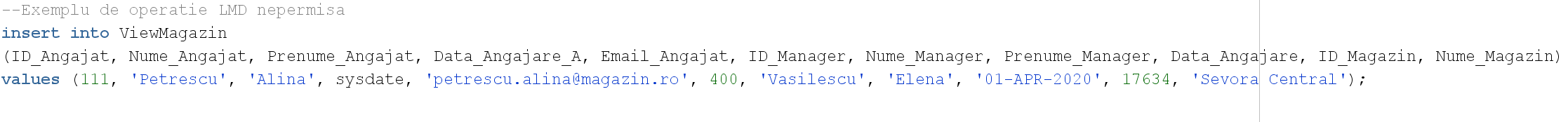
Operația de inserție (INSERT) de mai jos este nepermisă, deoarece vizualizarea include mai multe tabele și relații complexe.

Cod SQL:

insert into ViewMagazin

(ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Data\_Angajare\_A, Email\_Angajat, ID\_Manager, Nume\_Manager, Prenume\_Manager, Data\_Angajare, ID\_Magazin, Nume\_Magazin)

values (111, 'Petrescu', 'Alina', sysdate, 'petrescu.alina@magazin.ro', 400, 'Vasilescu', 'Elena', '01-APR-2020', 17634, 'Sevora Central');



A close-up of a letter

Description automatically generated

Acest INSERT nu este permis, deoarece vizualizarea include coloane din mai multe tabele (ANGAJAT, MAGAZIN, MANAGER), iar baza de date nu poate determina cum să distribuie aceste valori între tabelele de bază.

# **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:** Să se afișeze toate magazinele, managerii, angajații și comenzile, chiar dacă anumite legături lipsesc, pentru a avea o vedere completă a tuturor datelor.

Rezolvare:

select nvl(nvl(nvl(mg.ID\_Magazin, m.ID\_Magazin), l.ID\_Magazin), c.ID\_Magazin) as ID\_Magazin, mg.Nume\_Magazin,

m.ID\_Manager, m.Nume\_Manager, m.Prenume\_Manager, a.ID\_Angajat, a.Nume\_Angajat, a.Prenume\_Angajat, c.ID\_Comanda, c.Valoare\_Totala

from MAGAZIN mg full outer join MANAGER m on mg.ID\_Magazin = m.ID\_Magazin

full outer join LUCREAZA l on mg.ID\_Magazin = l.ID\_Magazin or m.ID\_Magazin = l.ID\_Magazin

full outer join ANGAJAT a on l.ID\_Angajat = a.ID\_Angajat

full outer join COMANDA c on mg.ID\_Magazin = c.ID\_Magazin;

A close up of text

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Division:** Să se afișeze angajații care au efectuat vânzări și care lucrează în magazine care se află în orașul cu ID-ul 2793.

Rezolvare:

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Analiza top-n:** Să se afișeze primii 5 angajați care au realizat cele mai mari vânzări în funcție de valoarea totală a comenzilor.

Rezolvare:

select ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Total\_Vanzari

from (select a.ID\_Angajat, a.Nume\_Angajat, a.Prenume\_Angajat, sum(va.Suma\_Totala) as Total\_Vanzari

from ANGAJAT a join VINDE v ON a.ID\_Angajat = v.ID\_Angajat

join VANZARE va on v.ID\_Vanzare = va.ID\_Vanzare

group by a.ID\_Angajat, a.Nume\_Angajat, a.Prenume\_Angajat

order by Total\_Vanzari desc) top

WHERE rownum <= 5;

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated