Proiect Sisteme de Gestiune a Bazelor de Date

De: Irina Coman

Seria 23, grupa 233

Anul universitar 2024-2025

Cuprins

[**1. Prezentați pe scurt baza de date (utilitatea ei).** 3](#_Toc187418051)

[**2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD, model de diagrama entitate-relație; nu se va accepta alt format).** 4](#_Toc187418052)

[**3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.** 5](#_Toc187418053)

[**4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, adăugând toate constrângerile de integritate necesare (chei primare, cheile externe etc).** 6](#_Toc187418054)

[**5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).** 18](#_Toc187418055)

[**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.** 40](#_Toc187418056)

[**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.** 43](#_Toc187418057)

[**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.** 46](#_Toc187418058)

[**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.** 51](#_Toc187418059)

[**10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.** 56](#_Toc187418060)

[**11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.** 58](#_Toc187418061)

[**12. Definiți un trigger de tip LDD. Declanșați trigger-ul.** 60](#_Toc187418062)

[**13. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).** 62](#_Toc187418063)

**Introducere**

Pentru acest proiect am realizat o bază de date pentru gestiunea unui lanț de magazine cosmetice.

Versiune SGBD: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 – Production

RAM alocat: 16.0 GB

Sistem de operare: Windows 11 Pro

Mașină virtuală: nu s-a folosit.

# **1. Prezentați pe scurt baza de date (utilitatea ei).**

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, clasificate pe categorii și review-uri, angajații care lucrează în aceste magazine, furnizorii de la care se comandă produsele pentru menținerea stocului, comenzile cu detaliile lor, plasate către furnizori, transportatorii care trimit comenzile de la furnizori către magazine, 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.

# **2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD, model de diagrama entitate-relație; nu se va accepta alt format).**

Diagrama E/R

A diagram of a company

Description automatically generated

# **3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.**

Diagrama conceptuală

A diagram of a company

Description automatically generated

# **4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, adăugând toate constrângerile de integritate necesare (chei primare, cheile externe etc).**

CREATE SEQUENCE ORAS\_SECV START WITH 1;

CREATE SEQUENCE MAGAZIN\_SECV START WITH 1;

CREATE SEQUENCE ANGAJAT\_SECV START WITH 101;

CREATE SEQUENCE MANAGER\_SECV START WITH 1001;

CREATE SEQUENCE FURNIZOR\_SECV START WITH 1;

CREATE SEQUENCE PRODUS\_SECV START WITH 1;

CREATE SEQUENCE CLIENT\_SECV START WITH 1;

CREATE SEQUENCE COMANDA\_SECV START WITH 1;

CREATE SEQUENCE VANZARE\_SECV START WITH 1;

CREATE SEQUENCE RECEPTIONEAZA\_SECV START WITH 1;

CREATE SEQUENCE LUCREAZA\_SECV START WITH 1;

CREATE SEQUENCE VINDE\_SECV START WITH 1;

CREATE SEQUENCE TRANSPORTATOR\_SECV START WITH 1;

CREATE SEQUENCE CATEGORIE\_SECV START WITH 1;

CREATE SEQUENCE REVIEW\_SECV START WITH 1;

A screenshot of a computer code

Description automatically generated A white paper with black text

Description automatically generated

* **ORAȘ**

--tabel 1: ORAS

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)

);

A close-up of a word

Description automatically generated

* **MAGAZIN**

--tabel 2: 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)

);

A black text on a white background

Description automatically generated

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;

A close up of words

Description automatically generated

* **MANAGER**

--tabel 3: 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)

);

A black text on a white background

Description automatically generated

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;

A close-up of words

Description automatically generated

* **ANGAJAT**

--tabel 4: 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)

);

A close up of a logo

Description automatically generated

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;

A close up of words

Description automatically generated

* **CATEGORIE**

--tabel 5:CATEGORIE

CREATE TABLE CATEGORIE(

ID\_Categorie NUMBER(10) CONSTRAINT PK\_CATEGORIE PRIMARY KEY,

Nume\_Categorie VARCHAR2(100) CONSTRAINT NN\_Nume\_Categorie NOT NULL

);

A black text on a white background

Description automatically generated

* **REVIEW**

--tabel 6: REVIEW

CREATE TABLE REVIEW(

ID\_Review NUMBER(10) CONSTRAINT PK\_REVIEW PRIMARY KEY,

Tip\_Review VARCHAR2(100) CONSTRAINT NN\_Tip\_Review NOT NULL,

Nr\_Stele NUMBER(10) CONSTRAINT NN\_Nr\_Stele NOT NULL

);

A close up of a text

Description automatically generated

* **PRODUS**

--tabel 7: 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,

ID\_Review NUMBER(10) CONSTRAINT FK\_PRODUS\_REVIEW REFERENCES REVIEW(ID\_Review),

ID\_Categorie NUMBER(10) CONSTRAINT FK\_PRODUS\_CATEGORIE REFERENCES CATEGORIE(ID\_Categorie)

);

A black text on a white background

Description automatically generated

ALTER TABLE PRODUS DROP CONSTRAINT FK\_PRODUS\_CATEGORIE;

ALTER TABLE PRODUS

ADD CONSTRAINT FK\_PRODUS\_CATEGORIE FOREIGN KEY (ID\_Categorie)

REFERENCES CATEGORIE(ID\_Categorie) ON DELETE SET NULL;

ALTER TABLE PRODUS DROP CONSTRAINT FK\_PRODUS\_REVIEW;

ALTER TABLE PRODUS

ADD CONSTRAINT FK\_PRODUS\_REVIEW FOREIGN KEY (ID\_Review)

REFERENCES REVIEW(ID\_Review) ON DELETE SET NULL;

A white background with black text

Description automatically generated

* **CLIENT**

--tabel 8: 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

);

A close up of a text

Description automatically generated

* **VÂNZARE**

--tabel 9: VANZARE

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)

);

A black text on a white background

Description automatically generated

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;

A close up of words

Description automatically generated

* **TRANSPORTATOR**

--tabel 10: TRANSPORTATOR

CREATE TABLE TRANSPORTATOR(

ID\_Transportator NUMBER(10) CONSTRAINT PK\_TRANSPORTATOR PRIMARY KEY,

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

Email\_Transportator VARCHAR2(100),

Telefon\_Transportator VARCHAR2(15)

);

**A black text on a white background

Description automatically generated**

* **FURNIZOR**

--tabel 11: 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)

);

A black text on a white background

Description automatically generated

* **COMANDĂ**

--tabel 12: COMANDA

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),

ID\_Transportator NUMBER(10) CONSTRAINT FK\_COMANDA\_TRANSPORTATOR REFERENCES TRANSPORTATOR(ID\_Transportator)

);



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;

ALTER TABLE COMANDA

DROP CONSTRAINT FK\_COMANDA\_TRANSPORTATOR;

ALTER TABLE COMANDA

ADD CONSTRAINT FK\_COMANDA\_TRANSPORTATOR FOREIGN KEY (ID\_Transportator)

REFERENCES TRANSPORTATOR(ID\_Transportator)ON DELETE SET NULL;

A screenshot of a computer

Description automatically generated

* **LUCREAZĂ**

--tabel 13: LUCREAZA

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;

A screenshot of a computer screen

Description automatically generated

* **RECEPȚIONEAZĂ**

--tabel 14: RECEPTIONEAZA

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)

);

A black text on a white background

Description automatically generated

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;

A close-up of a white background

Description automatically generated

* **VINDE**

--tabel 15: 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)

);

A close-up of a sign

Description automatically generated

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;

A screenshot of a computer

Description automatically generated

# **5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).**

* **ORAȘ**

--tabel 1: ORAS

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;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **MAGAZIN**

--tabel 2: MAGAZIN

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 screenshot of a computer

Description automatically generated

* **MANAGER**

--tabel 3: MANAGER

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

Description automatically generated

A screen shot of a computer

Description automatically generated

* **ANGAJAT**

--tabel 4: ANGAJAT

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 screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

* **CATEGORIE**

--tabel 5: CATEGORIE

INSERT INTO CATEGORIE (ID\_Categorie, Nume\_Categorie) VALUES (11, 'Makeup');

INSERT INTO CATEGORIE (ID\_Categorie, Nume\_Categorie) VALUES (12, 'Creme fata');

INSERT INTO CATEGORIE (ID\_Categorie, Nume\_Categorie) VALUES (13, 'Creme corp');

INSERT INTO CATEGORIE (ID\_Categorie, Nume\_Categorie) VALUES (14, 'Ingrijirea parului');

INSERT INTO CATEGORIE (ID\_Categorie, Nume\_Categorie) VALUES (15, 'Produse de styling');

select \* from CATEGORIE;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **REVIEW**

INSERT INTO REVIEW (ID\_Review, Tip\_Review, Nr\_Stele) VALUES (100, 'Foarte rau', 1);

INSERT INTO REVIEW (ID\_Review, Tip\_Review, Nr\_Stele) VALUES (200, 'Rau', 2);

INSERT INTO REVIEW (ID\_Review, Tip\_Review, Nr\_Stele) VALUES (300, 'OK', 3);

INSERT INTO REVIEW (ID\_Review, Tip\_Review, Nr\_Stele) VALUES (400, 'Bun', 4);

INSERT INTO REVIEW (ID\_Review, Tip\_Review, Nr\_Stele) VALUES (500, 'Foarte bun', 5);

select \* from REVIEW;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **PRODUS**

--tabel 7: PRODUS

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (15327, 'Fond de ten', 'Estee Lauder', 190.50, 100, 500, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (78432, 'Ruj Color Sensational', 'Maybelline', 40.00, 200, 400, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (31475, 'Mascara Lash Paradise', 'LOreal', 45.75, 150, 500, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (42435, 'Pudra Translucenta', 'Clinique', 130.00, 300, 400, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (53113, 'Paleta de Farduri', 'Urban Decay', 275.00, 250, 400, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (63782, 'Sampon Elseve', 'LOreal', 22.00, 120, 200, 14);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (78245, 'Balsam de Par Hydra Source', 'Biolage', 62.50, 180, 300, 14);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (89213, 'Spray Fixativ Elnett', 'LOreal', 18.00, 140, 500, 15);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (93456, 'Fond de ten Double Wear', 'Estee Lauder', 195.00, 130, 400, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (10234, 'Creion de Ochi Super Liner', 'LOreal', 17.00, 160, 300, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (11324, 'Ruj Matte Ink', 'Maybelline', 35.00, 170, 300, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (12456, 'Paleta de Contur', 'Anastasia Beverly Hills', 245.00, 90, 400, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (13579, 'Ser pentru Par Frizz Ease', 'John Frieda', 125.50, 110, 300, 14);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (14789, 'Mascara Better Than Sex', 'Too Faced', 159.00, 100, 300, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (15987, 'Creion de Sprancene Brow Wiz', 'Anastasia Beverly Hills', 83.00, 130, 100, 11);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (16387, 'Crema de hidratare SPF 50', 'CeraVe', 79.00, 180, 400, 12);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (17443, 'Liftactiv Collagen Specialist', 'Vichy', 130.00, 50, 300, 12);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (18622, 'Revitalift Laser X3', 'LOreal', 90.00, 110, 500, 12);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (19718, 'Dream Coat', 'ColorWow', 160.00, 100, 500, 15);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (20921, 'Spray de par', 'Moroccanoil', 135.00, 60, 200, 15);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (21574, 'Brazilian Bum Bum Cream', 'Sol de Janeiro', 104.00, 140, 300, 13);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (22374, 'Moisturizers', 'CeraVe', 40.00, 100, 400, 13);

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (23900, 'Lipikar Baume AP+M', 'La Roche-Posay', 130.00, 120, 500, 13);

select \* from PRODUS;

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **CLIENT**

--tabel 8: CLIENT

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');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1123, 'Popa', 'Mirela');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1239, 'Stefan', 'Daniel');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1395, 'Andrei', 'Mariana');

INSERT INTO CLIENT (ID\_Client, Nume\_Client, Prenume\_Client) VALUES (1476, 'Pavel', 'Georgiana');

select \* from CLIENT;

A white background with black text

Description automatically generatedA white background with black text

Description automatically generated

A screenshot of a table

Description automatically generated

* **VÂNZARE**

--tabel 9: VANZARE

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);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (11738684, DATE '2024-12-07', 190.50, 15327);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (12892486, DATE '2025-01-03', 90.0, 18622);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (13674794, DATE '2025-01-05', 40.00, 22374);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (14673574, DATE '2024-11-28', 45.75, 31475);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (15729319, DATE '2024-07-09', 130.00, 17443);

INSERT INTO VANZARE (ID\_Vanzare, Data\_Vanzare, Suma\_Totala, ID\_Produs) VALUES (16284030, DATE '2024-08-15', 104.00, 21574);

select \* from VANZARE;

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a table

Description automatically generated

* **TRANSPORTATOR**

--tabel 10: TRANSPORTATOR

INSERT INTO TRANSPORTATOR (ID\_Transportator, Nume\_Transportator, Email\_Transportator, Telefon\_Transportator) VALUES (143, 'Fan Courier', 'fancourier@gmail.com', '0743956778');

INSERT INTO TRANSPORTATOR (ID\_Transportator, Nume\_Transportator, Email\_Transportator, Telefon\_Transportator) VALUES (279, 'Sameday', 'sameday@yahoo.com', '0739275438');

INSERT INTO TRANSPORTATOR (ID\_Transportator, Nume\_Transportator, Email\_Transportator, Telefon\_Transportator) VALUES (339, 'Cargus', 'cargus@gmail.com', '0764975080');

INSERT INTO TRANSPORTATOR (ID\_Transportator, Nume\_Transportator, Email\_Transportator, Telefon\_Transportator) VALUES (494, 'DPD', 'dpd@gmail.com', '0747356944');

INSERT INTO TRANSPORTATOR (ID\_Transportator, Nume\_Transportator, Email\_Transportator, Telefon\_Transportator) VALUES (583, 'GLS', 'gls@yahoo.com', '0729385694');

select \* from TRANSPORTATOR;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **FURNIZOR**

--tabel 11: FURNIZOR

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');

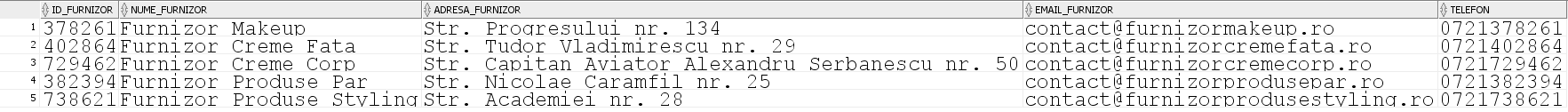
INSERT INTO FURNIZOR (ID\_Furnizor, Nume\_Furnizor, Adresa\_Furnizor, Email\_Furnizor, Telefon) VALUES

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

select \* from FURNIZOR;

A screenshot of a computer

Description automatically generated



* **COMANDĂ**

--tabel 12: COMANDA

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (6731983147, DATE '2022-01-15', DATE '2022-01-20', 1050.00, 28637, 78245, 143);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (4729864671, DATE '2022-01-18', DATE '2022-01-23', 4000.00, 34902, 53113, 339);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (3287346409, DATE '2023-01-21', DATE '2023-01-26', 2362.50, 52831, 89213, 494);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (4682289382, DATE '2023-01-25', DATE '2023-01-30', 9000.00, 52831, 42435, 279);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5643764201, DATE '2023-01-28', DATE '2023-02-02', 6250.00, 47644, 13579, 143);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (6721983197, DATE '2023-02-01', DATE '2023-02-06', 3200.00, 17634, 10234, 583);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5732964683, DATE '2023-02-03', DATE '2023-02-08', 2750.00, 34902, 31475, 279);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (4987346490, DATE '2023-02-05', DATE '2023-02-10', 3400.00, 17634, 78432, 494);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (4182289352, DATE '2023-02-08', DATE '2023-02-13', 2850.00, 34902, 93456, 143);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5343764210, DATE '2023-02-10', DATE '2023-02-15', 1500.00, 28637, 11324, 339);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (6021983147, DATE '2024-02-12', DATE '2024-02-17', 4700.00, 47644, 12456, 583);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5729864672, DATE '2024-02-14', DATE '2024-02-19', 5900.00, 28637, 15327, 143);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5287346410, DATE '2024-02-16', DATE '2024-02-21', 4100.00, 52831, 63782, 279);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (4782289383, DATE '2024-02-18', DATE '2024-02-23', 3500.00, 17634, 14789, 143);

INSERT INTO COMANDA (ID\_Comanda, Data\_Comanda, Data\_Livrare, Valoare\_Totala, ID\_Magazin, ID\_Produs, ID\_Transportator)

VALUES (5643764211, DATE '2024-02-20', DATE '2024-02-25', 4500.00, 17634, 15987, 339);

select \* from COMANDA;

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a data table

Description automatically generated

* **LUCREAZĂ**

--tabel 13: LUCREAZA

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

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **RECEPȚIONEAZĂ**

--tabel 14: RECEPTIONEAZA

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 generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **VINDE**

--tabel 15: VINDE

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);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (102, 11738684, 1123);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (107, 12892486, 9254);

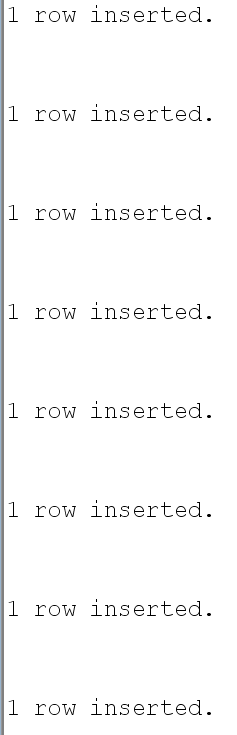
INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (105, 13674794, 1239);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (101, 14673574, 1123);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (104, 15729319, 1395);

INSERT INTO VINDE (ID\_Angajat, ID\_Vanzare, ID\_Client) VALUES (103, 16284030, 1476);

select \* from VINDE;

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

# **6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.**

Să se creeze o procedură stocată independentă prin care să se afișeze primele 3 branduri de produse după media review-urilor, indiferent de categorie. Dacă 2 branduri au aceeași medie, atunci se vor afișa în ordine descrescătoare după numărul de produse vândute ale acestui brand.

set serveroutput on

create or replace procedure branduri3 is

type t\_medie\_review is table of number index by varchar2(100);

medie\_review t\_medie\_review;

type t\_nr\_vanzari is table of number index by varchar2(100);

nr\_vanzari t\_nr\_vanzari;

type t\_branduri is table of varchar2(100);

branduri t\_branduri;

type t\_top3 is varray(5) of varchar2(100);

top3 t\_top3;

cursor c1 is

select p.Brand brand, avg(r.Nr\_Stele) medie

from PRODUS p join REVIEW r on p.ID\_Review = r.ID\_Review

group by p.Brand

order by medie desc;

cursor c2 is

select p.Brand brand, count(v.ID\_Vanzare) nrvanzari

from PRODUS p join VANZARE v on p.ID\_Produs = v.ID\_Produs(+)

group by p.Brand;

indexx pls\_integer :=1;

v\_brand varchar2(100);

v\_medie number;

v\_nrvanzari number;

type t\_combined is record (brand varchar2(100), medie number, nrvanzari number);

type t\_combined\_indexed is table of t\_combined index by pls\_integer;

combined t\_combined\_indexed;

begin

branduri := t\_branduri();

top3 := t\_top3();

combined := t\_combined\_indexed();

for i in c1 loop

branduri.extend;

branduri(indexx) := i.brand;

medie\_review(i.brand) := i.medie;

indexx := indexx + 1;

end loop;

for j in c2 loop

nr\_vanzari(j.brand) := j.nrvanzari;

end loop;

indexx := 1;

for i in 1..branduri.count loop

combined(indexx).brand := branduri(i);

combined(indexx).medie := medie\_review(branduri(i));

combined(indexx).nrvanzari := nr\_vanzari(branduri(i));

indexx := indexx + 1;

end loop;

for i in 1..combined.count-1 loop

for j in i+1..combined.count loop

if (combined(i).medie < combined(j).medie) or

(combined(i).medie = combined(j).medie and combined(i).nrvanzari < combined(j).nrvanzari) then

declare

temp\_brand varchar2(100);

temp\_medie number;

temp\_nrvanzari number;

begin

temp\_brand := combined(i).brand;

temp\_medie := combined(i).medie;

temp\_nrvanzari := combined(i).nrvanzari;

combined(i).brand := combined(j).brand;

combined(i).medie := combined(j).medie;

combined(i).nrvanzari := combined(j).nrvanzari;

combined(j).brand := temp\_brand;

combined(j).medie := temp\_medie;

combined(j).nrvanzari := temp\_nrvanzari;

end;

end if;

end loop;

end loop;

top3 := t\_top3(combined(1).brand, combined(2).brand, combined(3).brand);

dbms\_output.put\_line('Top 3 branduri:');

for i in 1..top3.count loop

dbms\_output.put\_line('Brand:' || top3(i));

end loop;

end branduri3;

/

begin

branduri3;

end;

/

A black text on a white background

Description automatically generated

A white background with black text

Description automatically generated

# **7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.**

Să se creeze o procedură stocată independentă care primește ca parametru o valoare de tip întreg și care afișează mai multe detalii despre angajatul care a avut cele mai multe vânzări, în funcție de valoare. Dacă valoarea introdusă ca parametru a procedurii este 1, atunci se afișează numele clienților cărora acest angajat le-a facut vânzări. Dacă valoarea este 2, atunci se afișează suma totală a vânzărilor făcute de acest angajat. Altfel, se afișează toate detaliile despre managerul acestuia.

set serveroutput on

create or replace procedure info\_top\_ang(indice in number) is

cursor c\_top\_ang is

select a.ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, count(v.ID\_Vanzare)

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

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

having count(\*) = (select max(count(\*))

from VINDE

group by ID\_Angajat);

cursor c\_clienti\_vanzari(id\_ang in number) is

select Nume\_Client, Prenume\_Client, Suma\_Totala

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

join ANGAJAT a on a.ID\_Angajat = vi.ID\_Angajat

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

where a.ID\_Angajat = id\_ang;

cursor c\_manager(id\_ang in number) is

select m.ID\_Manager, Nume\_manager, Prenume\_Manager, Email\_Manager, Data\_Angajare, ID\_Magazin

from MANAGER m join ANGAJAT a on a.ID\_Manager = m.ID\_Manager

where a.ID\_Angajat = id\_ang;

v\_id\_ang number;

v\_nume\_ang varchar2(100);

v\_prenume\_ang varchar2(100);

v\_nrvanzari number;

suma\_totala number := 0;

type rec\_info\_manager is record(

id\_man MANAGER.ID\_Manager%type,

nume\_man MANAGER.Nume\_Manager%type,

prenume\_man MANAGER.Prenume\_Manager%type,

email\_man MANAGER.Email\_Manager%type,

dataang\_man MANAGER.Data\_Angajare%type,

id\_mag MANAGER.ID\_Magazin%type);

info\_manager rec\_info\_manager;

begin

open c\_top\_ang;

fetch c\_top\_ang into v\_id\_ang, v\_nume\_ang, v\_prenume\_ang, v\_nrvanzari;

close c\_top\_ang;

dbms\_output.put\_line('Angajatul ' || v\_nume\_ang || ' ' || v\_prenume\_ang || ', cu ID-ul ' || v\_id\_ang || ' a efectuat ' || v\_nrvanzari || ' vanzari.');

if indice = 1 then

dbms\_output.put\_line('Clientii angajatului ' || v\_nume\_ang || ' ' || v\_prenume\_ang || ':');

for client in c\_clienti\_vanzari(v\_id\_ang) loop

dbms\_output.put\_line(client.Nume\_Client || ' ' || client.Prenume\_Client);

end loop;

elsif indice = 2 then

dbms\_output.put\_line('Suma totala incasata de ' || v\_nume\_ang || ' ' || v\_prenume\_ang || ':');

for v in c\_clienti\_vanzari(v\_id\_ang) loop

suma\_totala := suma\_totala + v.Suma\_Totala;

end loop;

dbms\_output.put\_line(suma\_totala);

else

open c\_manager(v\_id\_ang);

fetch c\_manager into info\_manager.id\_man, info\_manager.nume\_man, info\_manager.prenume\_man,

info\_manager.email\_man, info\_manager.dataang\_man, info\_manager.id\_mag;

close c\_manager;

dbms\_output.put\_line(info\_manager.id\_man || ' ' || info\_manager.nume\_man || ' ' || info\_manager.prenume\_man || ' ' ||

info\_manager.email\_man || ' ' || info\_manager.dataang\_man || ' ' || info\_manager.id\_mag);

end if;

end info\_top\_ang;

/

A close-up of a sign

Description automatically generated

begin

info\_top\_ang(1);

end;

/

A screen shot of a computer

Description automatically generated

begin

info\_top\_ang(2);

end;

/

A close-up of a computer screen

Description automatically generated

begin

info\_top\_ang(3);

end;

/

A close-up of a computer screen

Description automatically generated

# **8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

Să se creeze o funcție care să returneze un șir de caractere, ce conține informații despre managerul care lucrează într-un magazin situat în orașul cu ID-ul citit de la tastatură, dat ca parametru al funcției. Informațiile ce trebuie sa includă șirul de returnat sunt: ID-ul, numele și prenumele managerului și numele și regiunea orașului în care lucrează. Șirul de caractere ce va fi returnat de funcție va fi de forma 'Managerul cu ID-ul (v\_ID\_Manager) si numele (v\_Nume\_Manager, v\_Prenume\_Manager) lucreaza in orasul (v\_Nume\_Oras) din regiunea ( v\_Regiune)’.

Dacă în orașul dat nu există niciun manager, va apărea excepția NO\_DATA\_FOUND, iar dacă sunt cel puțin 2 manageri ce lucrează în acest oraș, va apărea excepția TOO\_MANY\_ROWS. De asemenea, se va trata și cazul în care se introduce la tastatură un oraș inexistent.

create or replace function info\_manager\_oras (id\_orass in number)

return varchar2

deterministic

is

v\_ID\_Manager MANAGER.ID\_Manager%type;

v\_Nume\_Manager MANAGER.Nume\_Manager%type;

v\_Prenume\_Manager MANAGER.Prenume\_Manager%type;

v\_Nume\_Oras ORAS.Nume\_Oras%type;

v\_Regiune ORAS.Regiune%type;

v\_exista\_oras number;

rezultat varchar2(500);

begin

select count(\*)

into v\_exista\_oras

from ORAS

where ID\_Oras = id\_orass;

if v\_exista\_oras = 0 then

raise\_application\_error(-20003, 'Orasul nu exista');

end if;

select m.ID\_Manager, m.Nume\_Manager, m.Prenume\_Manager, o.Nume\_Oras, o.Regiune

into v\_ID\_Manager, v\_Nume\_Manager, v\_Prenume\_Manager, v\_Nume\_Oras, v\_Regiune

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

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

where o.ID\_Oras = id\_orass;

rezultat := 'Managerul cu ID-ul ' || v\_ID\_Manager || ' si numele ' || v\_Nume\_Manager || ' ' || v\_Prenume\_Manager ||

' lucreaza in orasul ' || v\_Nume\_Oras || ' din regiunea ' || v\_Regiune;

return rezultat;

exception

when no\_data\_found then

raise\_application\_error(-20000, 'Nu exista manageri in orasul dat');

when too\_many\_rows then

raise\_application\_error(-20001, 'Exista mai multi manageri in orasul dat');

end info\_manager\_oras;

/

set serveroutput on

declare

id\_oras ORAS.ID\_Oras%type;

begin

id\_oras := &p\_id\_oras;

dbms\_output.put\_line(info\_manager\_oras(id\_oras));

end;

/

A black text on a white background

Description automatically generated

La introducerea orașului cu ID-ul 2793, care are 3 manageri, va apărea următoarea eroare

A screenshot of a computer program

Description automatically generated

La introducerea orașului cu ID-ul 3292, care are exact un manager, programul se va executa cu succes

A white background with black text

Description automatically generated

La introducerea orașului cu ID-ul 1761, care nu are niciun manager, va apărea următoarea eroare

A screenshot of a computer program

Description automatically generated

La introducerea orașului cu ID-ul 1, care nu există în baza de date, va apărea următoarea eroare

A screenshot of a computer code

Description automatically generated

# **9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.**

Să se creeze o procedură stocată independentă care, pe baza ID-ului unui produs și a unui număr minim de stele, date ca parametri, să afișeze numele produsului, brandul produsului, ID-ul comenzii și numele furnizorului în care este implicat produsul respectiv. Aceste informații se vor afișa doar dacă produsul are numărul de stele mai mare sau egal decât cel dat ca parametru în procedură și dacă starea comenzii în care este implicat produsul este "Livrată".

create or replace procedure info\_prod\_com\_furn(id\_prod number, nrminstele number) is

v\_Nume\_produs PRODUS.Nume\_Produs%type;

v\_Brand PRODUS.Brand%type;

v\_Nr\_Stele REVIEW.Nr\_Stele%type;

v\_ID\_Comanda COMANDA.ID\_Comanda%type;

v\_Nume\_Furnizor FURNIZOR.Nume\_Furnizor%type;

v\_Stare\_Comanda RECEPTIONEAZA.Stare\_Comanda%type;

exista\_prod number := 0;

exception\_no\_product exception;

exception\_lower\_than\_nrminstele exception;

exception\_no\_delivered\_order exception;

exception\_lower\_than\_nrminstele\_and\_no\_delivered\_order exception;

begin

select count(\*)

into exista\_prod

from PRODUS

where ID\_Produs = id\_prod;

if exista\_prod = 0 then

raise exception\_no\_product;

end if;

select p.Nume\_Produs, p.Brand, r.Nr\_Stele, c.ID\_Comanda, rec.Stare\_Comanda, f.Nume\_Furnizor

into v\_Nume\_Produs, v\_Brand, v\_Nr\_Stele, v\_ID\_Comanda, v\_Stare\_Comanda, v\_Nume\_Furnizor

from COMANDA c right outer join PRODUS p on c.ID\_Produs = p.ID\_Produs

join REVIEW r on p.ID\_Review = r.ID\_Review

join RECEPTIONEAZA rec on c.ID\_Comanda = rec.ID\_Comanda

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

where p.ID\_Produs = id\_prod;

if v\_Nr\_Stele < nrminstele and v\_Stare\_Comanda = 'Anulata' then

raise exception\_lower\_than\_nrminstele\_and\_no\_delivered\_order;

end if;

if v\_Nr\_Stele < nrminstele then

raise exception\_lower\_than\_nrminstele;

end if;

if v\_Stare\_Comanda = 'Anulata' then

raise exception\_no\_delivered\_order;

end if;

dbms\_output.put\_line('Produsul ' || v\_Nume\_Produs || ' de la brandul ' || v\_Brand || ' se regaseste in comanda cu ID-ul ' || v\_ID\_Comanda || ' furnizata de ' || v\_Nume\_Furnizor);

exception

when exception\_no\_product then

raise\_application\_error(-20000, 'Produsul dat nu exista');

when NO\_DATA\_FOUND then

raise\_application\_error(-20001, 'Produsul nu este implicat in nicio comanda');

when exception\_lower\_than\_nrminstele\_and\_no\_delivered\_order then

raise\_application\_error(-20005, 'Produsul are un numar de stele mai mic decat numarul dat si nu este implicat intr-o comanda livrata');

when exception\_lower\_than\_nrminstele then

raise\_application\_error(-20002, 'Produsul are un numar de stele mai mic decat numarul dat');

when exception\_no\_delivered\_order then

raise\_application\_error(-20003, 'Produsul nu este implicat intr-o comanda livrata');

when others then

raise\_application\_error(-20004, 'Alta eroare');

end info\_prod\_com\_furn;

/

A close-up of a logo

Description automatically generated

Caz în care produsul nu există

set serveroutput on

begin

info\_prod\_com\_furn(1, 1);

end;

/

A computer error message

Description automatically generated

Caz în care produsul nu este implicat în nicio comandă

set serveroutput on

begin

info\_prod\_com\_furn(16387, 1);

end;

/

A computer screen shot of a computer code

Description automatically generated

Caz în care produsul are un număr de stele mai mic decât numărul dat

set serveroutput on

begin

info\_prod\_com\_furn(10234, 5);

end;

/

A close-up of a computer code

Description automatically generated

Caz în care produsul nu este implicat într-o comandă livrată

set serveroutput on

begin

info\_prod\_com\_furn(42435, 1);

end;

/

A screen shot of a computer code

Description automatically generated

Caz în care produsul are un număr de stele mai mic decât numărul dat și nu este implicat într-o comandă livrată

set serveroutput on

begin

info\_prod\_com\_furn(42435, 5);

end;

/

A black text on a white background

Description automatically generated

Caz în care procedura a fost efectuată cu succes

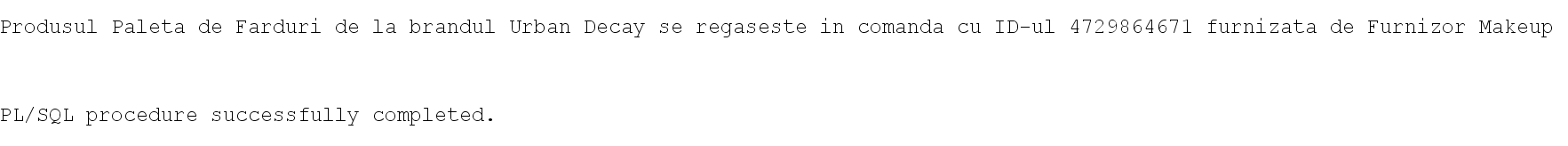
set serveroutput on

begin

info\_prod\_com\_furn(53113, 3);

end;

/



# **10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.**

Să se creeze un declanșator la nivel de comandă care după orice comandă LMD

(INSERT, UPDATE sau DELETE) să afișeze un mesaj de tipul „<<Operația LMD>> a fost realizată cu succes!”.

create or replace trigger afisare\_mesaj

after insert or update or delete on PRODUS

begin

if inserting then

dbms\_output.put\_line('Inserarea a fost realizata cu succes!');

elsif deleting then

dbms\_output.put\_line('Stergerea a fost realizata cu succes!');

else

dbms\_output.put\_line('Actualizarea a fost realizata cu succes!');

end if;

end afisare\_mesaj;

/

set serveroutput on

INSERT INTO PRODUS (ID\_Produs, Nume\_Produs, Brand, Pret, Stoc, ID\_Review, ID\_Categorie) VALUES (1, 'Sampon uscat', 'Batiste', 30, 50, 200, 15);

update PRODUS

set Stoc = stoc + 100

where ID\_Produs = 1;

delete from PRODUS where ID\_Produs = 1;

alter trigger afisare\_mesaj disable;

A close up of a sign

Description automatically generated

A white text box with black text

Description automatically generated

A close up of a sign

Description automatically generated

# **11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.**

Să se definească un declanșator la nivel de linie care să nu permită creșterea prețurilor produselor care au stocul mai mic sau egal decât 100.

create or replace trigger crestere\_pret\_interzisa

before update of Pret on PRODUS

for each row

when (NEW.Pret > OLD.Pret)

begin

if :OLD.Stoc <= 100 then

raise\_application\_error(-20010, 'Cresterea pretului unui produs avand stocul cel mult 100 este interzisa');

end if;

end crestere\_pret\_interzisa;

/

A close-up of a white background

Description automatically generated

Din cele 3 produse, doar produsul cu ID-ul 78432 are stocul > 100.

update PRODUS

set Pret = Pret + 1

where ID\_Produs in (15327, 78432, 12456);

A close-up of a document

Description automatically generated

update PRODUS

set Pret = Pret + 1

where Stoc > 100;

rollback;

A close-up of words

Description automatically generated

update PRODUS

set Pret = Pret + 1

where Stoc <= 100;

A close-up of a computer screen

Description automatically generated

alter trigger crestere\_pret\_interzisa disable;



# **12. Definiți un trigger de tip LDD. Declanșați trigger-ul.**

Să se definească un declanșator care, în urma unei operațiuni CREATE sau DROP asupra obiectelor din schemă, să afișeze pe ecran un mesaj de tipul 'Utilizatorul ... a creat un obiect de tip ... cu numele ... '.

set serveroutput on

create or replace trigger info\_create\_drop

after create or drop on schema

declare

v\_event varchar2(50);

v\_object varchar2(50);

v\_obj\_type varchar2(50);

v\_user varchar2(50);

begin

v\_event := sys.sysevent;

v\_object := sys.dictionary\_obj\_name;

v\_obj\_type := sys.dictionary\_obj\_type;

v\_user := sys.login\_user;

if v\_event = 'CREATE' then

dbms\_output.put\_line('Utilizatorul ' || v\_user || ' a creat un obiect de tip ' || v\_obj\_type || ' cu numele ' || v\_object);

elsif v\_event = 'DROP' then

dbms\_output.put\_line('Utilizatorul ' || v\_user || ' a sters un obiect de tip ' || v\_obj\_type || ' cu numele ' || v\_object);

end if;

end;

/



create table PATRON(

ID\_Patron NUMBER(5),

Nume\_Patron VARCHAR2(100),

Prenume\_Patron VARCHAR2(100)

);

drop table PATRON;

A close-up of a white background

Description automatically generated

alter trigger info\_create\_drop disable;

A black text on a white background

Description automatically generated

# **13. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).**

Să se definească un pachet care să permită gestiunea angajaților și a vânzărilor din magazine. Pachetul va conține:

a. o procedură care determină adăugarea unui angajat, dându-se informații complete despre acesta:

- codul angajatului va fi generat automat utilizându-se o secvență;

- informațiile personale vor fi date ca parametri (nume, prenume, email);

- data angajării va fi data curentă;

- codul managerului se va obține cu ajutorul unei funcții stocate în pachet care va avea ca parametri numele și prenumele managerului;

b. o funcție ce va returna angajatul cu cea mai mare sumă încasată din vânzări; dacă există mai mulți angajați cu aceeași sumă, se va alege primul în ordinea alfabetică;

c. o procedură pentru schimbarea poziției angajatului cu cea mai mare sumă încasată din vânzări în 'Angajatul lunii';

d. o funcție ce va returna cea mai recentă vânzare făcută de un angajat cu ID-ul dat;

e. o procedură care determină ștergerea unui angajat, după un ID dat.

Pentru fiecare funcție și procedură se vor trata toate excepțiile corespunzătoare.

create or replace package gestionare\_angajati\_vanzari as

type angajat\_vanzari is table of number(5) index by pls\_integer;

type angajat\_sters is record (

id\_ang ANGAJAT.ID\_Angajat%type,

nume\_ang ANGAJAT.Nume\_Angajat%type,

prenume\_ang ANGAJAT.Prenume\_Angajat%type,

email\_ang ANGAJAT.Email\_Angajat%type

);

procedure adauga\_angajat (

nume\_ang in ANGAJAT.Nume\_Angajat%type,

prenume\_ang in ANGAJAT.Prenume\_Angajat%type,

email\_ang in ANGAJAT.Email\_Angajat%type,

nume\_man in MANAGER.Nume\_Manager%type,

prenume\_man in MANAGER.Prenume\_manager%type

);

function top\_angajat return ANGAJAT.ID\_Angajat%type;

procedure angajatul\_lunii (id\_ang in ANGAJAT.ID\_Angajat%type);

function cea\_mai\_recenta\_vanzare (id\_ang in ANGAJAT.ID\_Angajat%type) return VANZARE.ID\_Vanzare%type;

procedure sterge\_angajat (id\_angg in ANGAJAT.ID\_Angajat%type);

end gestionare\_angajati\_vanzari;

/



create or replace package body gestionare\_angajati\_vanzari is

procedure adauga\_angajat (

nume\_ang in ANGAJAT.Nume\_Angajat%type,

prenume\_ang in ANGAJAT.Prenume\_Angajat%type,

email\_ang in ANGAJAT.Email\_Angajat%type,

nume\_man in MANAGER.Nume\_Manager%type,

prenume\_man in MANAGER.Prenume\_manager%type)

is

id\_angajat ANGAJAT.ID\_Angajat%type;

id\_manager ANGAJAT.ID\_Manager%type;

begin

id\_angajat := ANGAJAT\_SECV.nextval;

dbms\_output.put\_line('ID-ul angajatului generat: ' || id\_angajat);

select ID\_Manager into id\_manager

from MANAGER

where Nume\_Manager = nume\_man

and Prenume\_Manager = prenume\_man;

dbms\_output.put\_line('ID-ul managerului obtinut: ' || id\_manager);

insert into ANGAJAT (ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat, Data\_Angajare\_A, ID\_Manager)

values (id\_angajat, nume\_ang, prenume\_ang, email\_ang, sysdate, id\_manager);

dbms\_output.put\_line('Angajatul a fost adaugat cu succes!');

exception

when no\_data\_found then

raise\_application\_error(-20001, 'Managerul nu exista');

when others then

raise\_application\_error(-20002, 'Eroare in adaugarea angajatului');

end adauga\_angajat;

function top\_angajat return ANGAJAT.ID\_Angajat%type

is

ang\_sum\_v angajat\_vanzari;

max\_vanzari number := 0;

top\_angajat\_id ANGAJAT.ID\_Angajat%type;

begin

for ang in (

select a.ID\_Angajat, sum(v.Suma\_Totala) as Total\_Vanzari

from ANGAJAT a

join VINDE vi on a.ID\_Angajat = vi.ID\_Angajat

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

group by a.ID\_Angajat) loop

ang\_sum\_v(ang.ID\_Angajat) := ang.Total\_Vanzari;

if ang.Total\_Vanzari > max\_vanzari then

max\_vanzari := ang.Total\_Vanzari;

top\_angajat\_id := ang.ID\_Angajat;

end if;

end loop;

return top\_angajat\_id;

exception

when no\_data\_found then

raise\_application\_error(-20003, 'Niciun angajat nu a facut o vanzare');

when others then

raise\_application\_error(-20004, 'Alta eroare in gasirea angajatului cu cea mai mare suma incasata in vanzari');

end top\_angajat;

procedure angajatul\_lunii (id\_ang in ANGAJAT.ID\_Angajat%type) is

begin

update LUCREAZA

set Pozitie = 'Angajatul lunii'

where ID\_Angajat = id\_ang;

dbms\_output.put\_line('Angajatul lunii a fost actualizat cu succes');

exception

when others then

raise\_application\_error(-20005, 'Eroare la actualizarea angajatului lunii');

end angajatul\_lunii;

function cea\_mai\_recenta\_vanzare (id\_ang in ANGAJAT.ID\_Angajat%type)

return VANZARE.ID\_Vanzare%type

is

id\_vanzare\_recenta VANZARE.ID\_Vanzare%type;

exceptie\_angajatul\_nu\_exista exception;

--exceptie\_angajatul\_nu\_a\_vandut\_nimic exception;

exista\_ang number := 0;

begin

select count(\*) into exista\_ang

from ANGAJAT

where ID\_Angajat = id\_ang;

if exista\_ang = 0 then

raise exceptie\_angajatul\_nu\_exista;

end if;

with vanz\_ang as

(select v.ID\_Vanzare, a.ID\_Angajat

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

join ANGAJAT a on vi.ID\_Angajat = a.ID\_Angajat

where a.ID\_Angajat = id\_ang

order by Data\_Vanzare desc)

select va.ID\_Vanzare into id\_vanzare\_recenta

from vanz\_ang va

where rownum = 1;

return id\_vanzare\_recenta;

exception

when exceptie\_angajatul\_nu\_exista then

raise\_application\_error(-20006, 'Angajatul nu exista');

when no\_data\_found then

raise\_application\_error(-20007, 'Angajatul nu a efectuat nicio vanzare');

when others then

raise\_application\_error(-20008, 'Alta eroare in gasirea celei mai recente vanzari');

end cea\_mai\_recenta\_vanzare;

procedure sterge\_angajat (id\_angg in ANGAJAT.ID\_Angajat%type) is

angajat\_sters1 angajat\_sters;

exista\_ang number := 0;

exceptie\_angajatul\_nu\_exista exception;

begin

select count(\*) into exista\_ang

from ANGAJAT

where ID\_Angajat = id\_angg;

if exista\_ang = 0 then

raise exceptie\_angajatul\_nu\_exista;

end if;

select ID\_Angajat, Nume\_Angajat, Prenume\_Angajat, Email\_Angajat

into angajat\_sters1.id\_ang, angajat\_sters1.nume\_ang, angajat\_sters1.prenume\_ang, angajat\_sters1.email\_ang

from ANGAJAT

where ID\_Angajat = id\_angg;

delete from ANGAJAT where ID\_Angajat = id\_angg;

dbms\_output.put\_line('Angajatul ' || angajat\_sters1.nume\_ang || ' ' || angajat\_sters1.prenume\_ang ||

', cu ID-ul ' || angajat\_sters1.id\_ang || ' si email-ul ' || angajat\_sters1.email\_ang || ' a fost sters cu succes!');

exception

when exceptie\_angajatul\_nu\_exista then

raise\_application\_error(-20009, 'Angajatul nu exista');

when others then

raise\_application\_error(-20008, 'Alta eroare in stergerea angajatului');

end sterge\_angajat;

end gestionare\_angajati\_vanzari;

/



Cu ajutorul pachetului definit,

a1. să se adauge angajatul Pop Ion, cu adresa de email popion@gmail.com, aflat sub conducerea managerului Ionescu Maria.

a2. ce se întâmplă dacă se încearcă adăugarea unui angajat cu un manager inexistent?

b. să se afișeze, cu ajutorul funcției corespunzătoare din pachet, angajatul cu cea mai mare sumă încasată din vânzări și să se schimbe poziția acestuia în ’Angajatul lunii’.

c. care este cea mai recentă vânzare făcută de angajatul cu ID-ul 101? Dar 108?

d. să se șteargă angajatul introdus în subpunctul a) și să se afișeze mesajul corespunzător.

drop sequence ANGAJAT\_SECV;

CREATE SEQUENCE ANGAJAT\_SECV START WITH 200 INCREMENT BY 1;

A white background with black text

Description automatically generated

set serveroutput on

--a1

begin

gestionare\_angajati\_vanzari.adauga\_angajat('Pop', 'Ion', 'popion@gmail.com', 'Ionescu', 'Maria');

end;

/

A white background with black text

Description automatically generated

--a2

begin

gestionare\_angajati\_vanzari.adauga\_angajat('Pop', 'Ion', 'popion@gmail.com', 'Ionescuuu', 'Maria');

end;

/

A close-up of a computer screen

Description automatically generated

--b

declare

id\_top\_angajat ANGAJAT.ID\_Angajat%type;

begin

id\_top\_angajat := gestionare\_angajati\_vanzari.top\_angajat;

gestionare\_angajati\_vanzari.angajatul\_lunii(id\_top\_angajat);

end;

/

A close-up of words

Description automatically generated

--c

declare

id\_vanzare\_101 VANZARE.ID\_Vanzare%type;

id\_vanzare\_108 VANZARE.ID\_Vanzare%type;

begin

id\_vanzare\_101 := gestionare\_angajati\_vanzari.cea\_mai\_recenta\_vanzare(101);

dbms\_output.put\_line('Cea mai recenta vanzarea a angajatului cu ID-ul 101: ' || id\_vanzare\_101);

id\_vanzare\_108 := gestionare\_angajati\_vanzari.cea\_mai\_recenta\_vanzare(108);

dbms\_output.put\_line('Cea mai recenta vanzarea a angajatului cu ID-ul 108: ' || id\_vanzare\_108);

end;

/

A white screen with black text

Description automatically generated

--d

begin

gestionare\_angajati\_vanzari.sterge\_angajat(205);

end;

/

A close-up of a white background

Description automatically generated