***Lant de magazine – DVDs & Blu-Rays***

***Răican Mihai***

***Grupa 244***

**EXERCITIUL 1**

**Proiectul va consta in modelarea unei baze de date pentru un lanț de magazine, aflate in mai**

**multe orașe din tara. Un magazin se va ocupa cu vânzarea de DVD-uri si Blu-Ray-uri ce vor conține**

**filme sau seriale. Fiecare magazin va avea un manager si mai multi angajati**

**care se vor ocupa de vânzare, vor face recomandări clienților si vor pregăti produsele care**

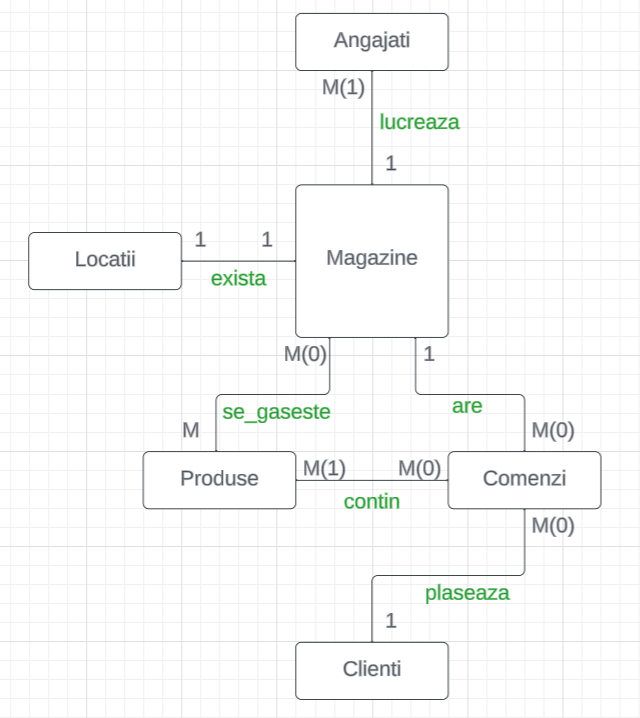
**urmează sa fie ridicate din magazin. Acest lanț de magazine are si un site in care clienții își pot face**

**cont si pot comanda produsele dorite: direct la domiciliu pentru orice produs de pe site, sau la**

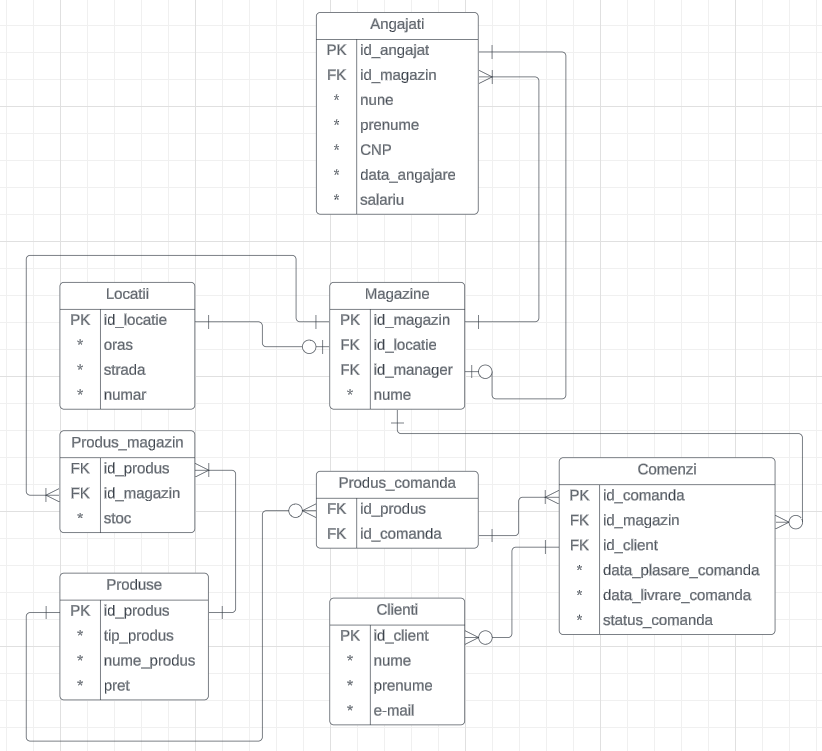
**unul dintre magazine, probabil cel mai apropiat, daca produsul respectiv nu se afla in stocul**

**magazinului respectiv, urmând ca produsul sa fie ridicat din magazin de către client.**

**EXERCITIUL 2**

****

**EXERCITIUL 3**



**EXERCITIUL 4**

CREATE SEQUENCE Angajat\_id

START WITH 1000

INCREMENT BY 1;

CREATE SEQUENCE Magazin\_id

START WITH 2000

INCREMENT BY 1;

CREATE SEQUENCE Locatie\_id

START WITH 3000

INCREMENT BY 1;

CREATE SEQUENCE Comanda\_id

START WITH 4000

INCREMENT BY 1;

CREATE SEQUENCE Client\_id

START WITH 5000

INCREMENT BY 1;

CREATE SEQUENCE Produs\_id

START WITH 6000

INCREMENT BY 1;

create table Angajati (

ID\_Angajat number(4),

ID\_Magazin number(4) not null,

Nume varchar2(25) not null,

Prenume varchar2(25) not null,

CNP number(13) not null,

Data\_Angajare date default sysdate,

Salariu number(5) default 1900,

constraint PK\_ID\_Angajat primary key (ID\_Angajat),

constraint UQ\_CNP unique (CNP),

constraint CK\_CNP check (length(CNP) = 13)

);

--------------------------------------------------------------------------------

create table Magazine (

ID\_Magazin number(4),

ID\_Locatie number(4) not null,

Nume varchar(50),

ID\_Manager number(4),

constraint PK\_ID\_Magazin primary key (ID\_Magazin)

);

--------------------------------------------------------------------------------

create table Locatii (

ID\_Locatie number(4),

Oras varchar2(20) not null,

Strada varchar2(20) not null,

Numar number(4) not null,

constraint PK\_ID\_Locatie primary key (ID\_Locatie)

);

--Magazine

ALTER TABLE Magazine

ADD CONSTRAINT FK\_ID\_Manager FOREIGN KEY (ID\_Manager) REFERENCES Angajati(ID\_Angajat);

ALTER TABLE Magazine

ADD CONSTRAINT FK\_ID\_Locatie FOREIGN KEY (ID\_Locatie) REFERENCES Locatii(ID\_Locatie);

--------------------------------------------------------------------------------

create table Produse (

ID\_Produs number(4),

Tip\_Produs varchar(20),

Nume\_Produs varchar(50),

Pret number(3),

constraint PK\_ID\_Produs primary key (ID\_Produs)

);

--------------------------------------------------------------------------------

create table Clienti (

ID\_Client number(6),

Nume varchar2(25),

Prenume varchar2(25),

Email varchar2(50),

constraint PK\_ID\_Client primary key (ID\_Client)

);

--------------------------------------------------------------------------------

create table Comenzi (

ID\_Comanda number(6),

ID\_Magazin number(4),

ID\_Client number(6),

Data\_Plasare\_Comanda date default sysdate,

Data\_Livrare\_Comanda date,

Status\_Comanda varchar2(100),

constraint PK\_ID\_Comanda primary key (ID\_Comanda)

);

ALTER TABLE Comenzi

ADD CONSTRAINT FK\_ID\_Magazin\_Comenzi FOREIGN KEY (ID\_Magazin) REFERENCES Magazine(ID\_Magazin);

ALTER TABLE Comenzi

ADD CONSTRAINT FK\_ID\_Client FOREIGN KEY (ID\_Client) REFERENCES Clienti(ID\_Client);

--------------------------------------------------------------------------------

create table Produs\_Magazin (

ID\_Produs number(4),

ID\_Magazin number(4) not null,

Stoc number(4)

);

ALTER TABLE Produs\_Magazin

ADD CONSTRAINT FK\_ID\_Produs\_Magazin FOREIGN KEY (ID\_Produs) REFERENCES Produse(ID\_Produs);

ALTER TABLE Produs\_Magazin

ADD CONSTRAINT FK\_ID\_Magazin\_Produs FOREIGN KEY (ID\_Magazin) REFERENCES Magazine(ID\_Magazin);

--------------------------------------------------------------------------------

create table Produs\_Comanda (

ID\_Produs number(4),

ID\_Comanda number(6),

Cantitate number(3)

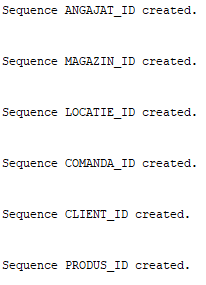
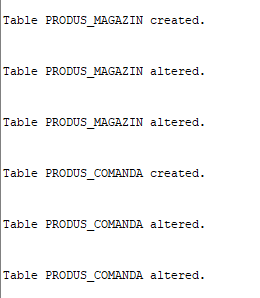
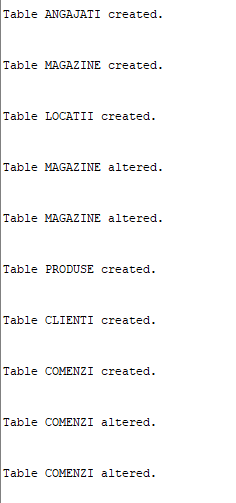
);

ALTER TABLE Produs\_Comanda

ADD CONSTRAINT FK\_ID\_Produs\_Comanda FOREIGN KEY (ID\_Produs) REFERENCES Produse(ID\_Produs);

ALTER TABLE Produs\_Comanda

ADD CONSTRAINT FK\_ID\_Comanda FOREIGN KEY (ID\_Comanda) REFERENCES Comenzi(ID\_Comanda);



**EXERCITIUL 5**

**ANGAJATI**

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2007, 'Vasilescu', 'Ionut', 1004071523456,

TO\_DATE('2011-08-20', 'YYYY-MM-DD'), 3200);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2008, 'Popescu', 'Maria', 1885060826789,

TO\_DATE('2019-05-10', 'YYYY-MM-DD'), 2500);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2004, 'Ionescu', 'Andrei', 1782100229876,

TO\_DATE('2008-12-01', 'YYYY-MM-DD'), 3000);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2002, 'Dragomir', 'Elena', 1891051023456,

TO\_DATE('2013-07-18', 'YYYY-MM-DD'), 2700);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2001, 'Stoica', 'Gabriel', 1989080323456,

TO\_DATE('2010-09-23', 'YYYY-MM-DD'), 2900);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2006, 'Dumitru', 'Adriana', 1990021723456,

TO\_DATE('2017-03-14', 'YYYY-MM-DD'), 2800);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2003, 'Radulescu', 'Mihai', 1986093023456,

TO\_DATE('2012-11-05', 'YYYY-MM-DD'), 3100);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2009, 'Constantin', 'Ana', 1937070723456,

TO\_DATE('2021-02-19', 'YYYY-MM-DD'), 2600);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2005, 'Florescu', 'Dragos', 1913021023456,

TO\_DATE('2018-09-30', 'YYYY-MM-DD'), 2700);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2000, 'Ilie', 'Daniel', 1831010123456,

TO\_DATE('2014-02-27', 'YYYY-MM-DD'), 2800);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2001, 'Ionescu', 'Alexandra', 1925051523456,

TO\_DATE('2016-07-08', 'YYYY-MM-DD'), 3000);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2003, 'Popescu', 'Marian', 1963072123456,

TO\_DATE('2009-12-15', 'YYYY-MM-DD'), 2500);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2004, 'Radoi', 'Ana', 1889100223456,

TO\_DATE('2013-06-19', 'YYYY-MM-DD'), 3200);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2002, 'Dragomir', 'Mihai', 1846071623456,

TO\_DATE('2010-05-21', 'YYYY-MM-DD'), 2700);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2006, 'Neagoe', 'Andreea', 1900040323456,

TO\_DATE('2017-01-10', 'YYYY-MM-DD'), 2900);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2005, 'Ionel', 'Cristian', 1758051723456,

TO\_DATE('2012-08-05', 'YYYY-MM-DD'), 2800);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2008, 'Albu', 'Elena', 1003123023456,

TO\_DATE('2022-03-27', 'YYYY-MM-DD'), 3100);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2007, 'Constantin', 'George', 1817080923456,

TO\_DATE('2019-11-12', 'YYYY-MM-DD'), 2600);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2009, 'Bucur', 'Mihaela', 1986100423456,

TO\_DATE('2020-06-25', 'YYYY-MM-DD'), 2800);

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2003, 'Minculescu', 'Diana', 1971122723456,

TO\_DATE('2015-04-03', 'YYYY-MM-DD'), 3000);



**LOCATII**

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Buzau', 'Principala', 43);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Brasov', 'Libertatii', 17);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Timisoara', 'Mihai Viteazu', 9);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Cluj-Napoca', 'Avram Iancu', 32);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Iasi', 'Stefan cel Mare', 25);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Constanta', 'Mamaia', 7);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Sibiu', 'Victoriei', 12);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Galati', 'Dunarii', 19);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Ploiesti', 'Revolu?iei', 8);

insert into Locatii (ID\_Locatie, Oras, Strada, Numar)

values (Locatie\_id.nextval, 'Oradea', 'Unirii', 14);



**MAGAZINE**

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3005, 'Media Metropolis Constanta', 1002);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3002, 'Film Fortress Timisoara', 1005);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3008, 'Cinematic Stash Prahova', null);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3003, 'DVD Delight Cluj-Napoca', 1011);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3007, 'Blu-ray Bazar Galati', 1001);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3001, 'Blockbuster Bliss Brasov', null);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3006, 'Blu-Ray Boutique Sibiu', 1004);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3009, 'Just Watch Oradea', null);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3004, 'Blockbuster Bliss Iasi', 1017);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

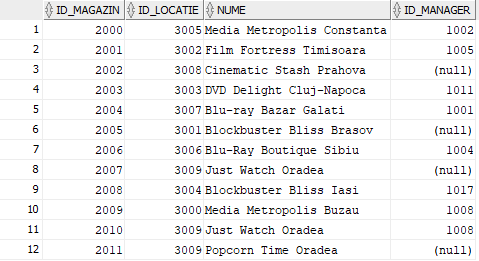
values (Magazin\_id.nextval, 3000, 'Media Metropolis Buzau', 1008);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3009, 'Just Watch Oradea', 1008);

insert into Magazine (ID\_Magazin, ID\_Locatie, Nume, ID\_Manager)

values (Magazin\_id.nextval, 3009, 'Popcorn Time Oradea', null);



**CLIENTI**

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Petrescu', 'Alin', 'petrescualin@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Ionescu', 'Maria', 'ionescumaria@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Popescu', 'Andrei', 'popescuandrei@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Stanescu', 'Elena', 'stanescuelena@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Mihai', 'Cristina', 'mihaicristina@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Georgescu', 'Alexandru', 'georgescualexandru@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Dumitrescu', 'Andreea', 'dumitrescuandreea@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

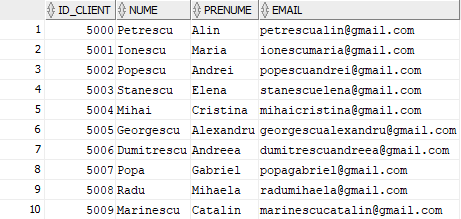
values (Client\_id.nextval, 'Popa', 'Gabriel', 'popagabriel@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Radu', 'Mihaela', 'radumihaela@gmail.com');

insert into clienti (ID\_CLIENT, NUME, PRENUME, EMAIL)

values (Client\_id.nextval, 'Marinescu', 'Catalin', 'marinescucatalin@gmail.com');



**COMENZI**

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2004, 5008, TO\_DATE('2023-05-24', 'YYYY-MM-DD'), TO\_DATE('2023-05-25', 'YYYY-MM-DD'), 'Procesare comanda');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2006, 5001, TO\_DATE('2023-05-22', 'YYYY-MM-DD'), TO\_DATE('2023-05-26', 'YYYY-MM-DD'), 'Produse predate curierului');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2003, 5009, TO\_DATE('2023-05-23', 'YYYY-MM-DD'), TO\_DATE('2023-05-30', 'YYYY-MM-DD'), 'Produse in depozitul curierului');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2008, 5005, TO\_DATE('2023-05-21', 'YYYY-MM-DD'), TO\_DATE('2023-05-23', 'YYYY-MM-DD'), 'Produse in curs de livrare');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2002, 5002, TO\_DATE('2023-05-20', 'YYYY-MM-DD'), TO\_DATE('2023-05-22', 'YYYY-MM-DD'), 'Produsele vor fi gata de ridicare din magazin');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2007, 5006, TO\_DATE('2023-05-19', 'YYYY-MM-DD'), TO\_DATE('2023-05-31', 'YYYY-MM-DD'), 'Procesare comanda');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2001, 5007, TO\_DATE('2023-05-21', 'YYYY-MM-DD'), TO\_DATE('2023-05-24', 'YYYY-MM-DD'), 'Produse predate curierului');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

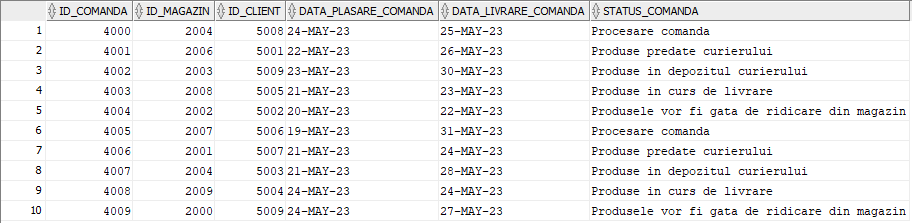
values (Comanda\_id.nextval, 2004, 5003, TO\_DATE('2023-05-21', 'YYYY-MM-DD'), TO\_DATE('2023-05-28', 'YYYY-MM-DD'), 'Produse in depozitul curierului');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2009, 5004, TO\_DATE('2023-05-24', 'YYYY-MM-DD'), TO\_DATE('2023-05-24', 'YYYY-MM-DD'), 'Produse in curs de livrare');

insert into Comenzi (ID\_COMANDA, ID\_MAGAZIN, ID\_CLIENT, DATA\_PLASARE\_COMANDA, DATA\_LIVRARE\_COMANDA, STATUS\_COMANDA)

values (Comanda\_id.nextval, 2000, 5009, TO\_DATE('2023-05-24', 'YYYY-MM-DD'), TO\_DATE('2023-05-27', 'YYYY-MM-DD'), 'Produsele vor fi gata de ridicare din magazin');



**PRODUSE**

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'DVD', 'Inception', 89);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'Blu-Ray UHD', 'Avengers: Endgame', 120);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'DVD', 'The Social Network', 45);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'Blu-Ray 4k', 'Interstellar', 99);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'Blu-Ray UHD', 'Mad Max: Fury Road', 75);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'DVD', 'Gone Girl', 32);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'Blu-Ray 4k', 'Dunkirk', 110);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

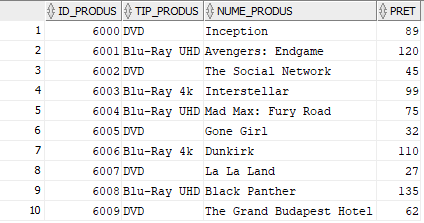
values (Produs\_id.nextval, 'DVD', 'La La Land', 27);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'Blu-Ray UHD', 'Black Panther', 135);

insert into produse (ID\_PRODUS, TIP\_PRODUS, NUME\_PRODUS, PRET)

values (Produs\_id.nextval, 'DVD', 'The Grand Budapest Hotel', 62);



**PRODUS\_COMANDA**

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6000, 4000, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6005, 4003, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6003, 4005, 2);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6001, 4006, 5);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6009, 4008, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6008, 4001, 2);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6004, 4007, 3);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6006, 4009, 2);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6007, 4004, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6002, 4002, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6003, 4002, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6009, 4004, 3);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6005, 4007, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6008, 4002, 1);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6000, 4005, 9);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6002, 4005, 2);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6007, 4000, 2);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

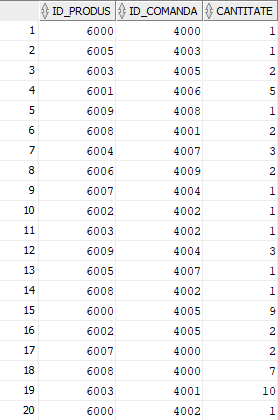
values (6008, 4000, 7);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6003, 4001, 10);

insert into Produs\_Comanda (ID\_PRODUS, ID\_COMANDA, Cantitate)

values (6000, 4002, 1);



**PRODUS\_MAGAZIN**

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6000, 2005, 78);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6001, 2001, 43);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6002, 2009, 105);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6003, 2007, 92);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6004, 2003, 17);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6005, 2000, 124);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6006, 2008, 5);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6007, 2002, 89);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6008, 2006, 1);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6009, 2004, 130);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6000, 2002, 42);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6001, 2008, 77);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6002, 2006, 13);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6003, 2000, 108);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6004, 2004, 61);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6005, 2001, 29);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6006, 2003, 73);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6007, 2009, 96);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6008, 2005, 8);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6009, 2007, 115);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6000, 2006, 93);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6001, 2003, 57);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6002, 2005, 19);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6003, 2001, 114);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6004, 2007, 80);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6005, 2004, 32);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6006, 2009, 68);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

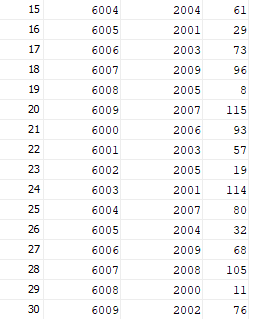
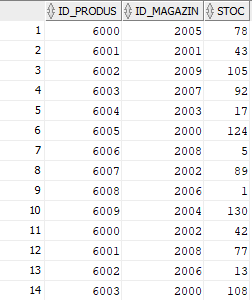
values (6007, 2008, 105);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6008, 2000, 11);

insert into produs\_magazin (ID\_PRODUS, ID\_MAGAZIN, STOC)

values (6009, 2002, 76);



**EXERCITIUL 6**

**Scrieti un subprogram stocat independent care primeste ca parametru un nume de oras si care:**

* **Foloseste un tablou indexat pentru a afisa numele, prenumele si starea comenzilor clientilor.**
* **Foloseste un tablou imbricat pentru a afisa numele primilor 10 angajati in functie de salariu.**
* **Foloseste un vector pentru a afisa toate orasele in care exista magazine.**
* **Afiseaza un mesaj care sa ne informeze daca in orasul transmis ca parametru exista sau nu un magazin.**

create or replace procedure ex6\_subprogram\_stocat\_independent

(v\_nume\_oras locatii.oras%type default 'Buzau')

is

type cl\_com\_rec is record (

Nume CLIENTI.NUME%TYPE,

Prenume CLIENTI.PRENUME%TYPE,

Status\_comanda COMENZI.STATUS\_COMANDA%TYPE

);

-- tablou indexat

type tablou\_indexat is table of cl\_com\_rec

index by pls\_integer;

t\_indexat tablou\_indexat;

-- tablou imbricat

type tablou\_imbricat is table of angajati%rowtype;

t\_imbricat tablou\_imbricat := tablou\_imbricat();

-- vector

type vector is varray(20) of locatii.oras%type;

t\_vector vector := vector();

v\_oras locatii.oras%type;

begin

-- tablou indexat

for i in (

select nume, prenume, status\_comanda

from clienti cl, comenzi com

where cl.id\_client = com.id\_client

) loop

t\_indexat(t\_indexat.count + 1).Nume := i.Nume;

t\_indexat(t\_indexat.count).Prenume := i.Prenume;

t\_indexat(t\_indexat.count).Status\_comanda := i.Status\_comanda;

end loop;

for i in t\_indexat.first..t\_indexat.last loop

dbms\_output.put\_line(t\_indexat(i).nume || ' ' || t\_indexat(i).prenume

|| ' ' || t\_indexat(i).status\_comanda);

end loop;

dbms\_output.new\_line;

-- tablou imbricat

for i in (

select \*

from (

select \*

from angajati

order by salariu desc

) Top10Salarii

where rownum <= 10

) loop

t\_imbricat.extend;

t\_imbricat(t\_imbricat.count) := i;

end loop;

for i in t\_imbricat.first..t\_imbricat.last loop

dbms\_output.put\_line(t\_imbricat(i).nume || ' ' || t\_imbricat(i).salariu);

end loop;

dbms\_output.new\_line;

-- vector

for i in (

select oras

from locatii l, magazine m

where l.id\_locatie = m.id\_locatie

order by oras

) loop

t\_vector.extend;

t\_vector(t\_vector.count) := i.oras;

end loop;

for i in t\_vector.first..t\_vector.last loop

dbms\_output.put\_line(i || ' ' || t\_vector(i));

end loop;

select oras into v\_oras

from locatii

where upper(oras) = upper(v\_nume\_oras);

dbms\_output.put\_line('In ' || v\_nume\_oras || ' exista un magazin');

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista magazine in ' || v\_nume\_oras);

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multe magazine in ' || v\_nume\_oras);

WHEN OTHERS THEN

dbms\_output.put\_line('Alta eroare');

end ex6\_subprogram\_stocat\_independent;

/

begin

ex6\_subprogram\_stocat\_independent;

end;

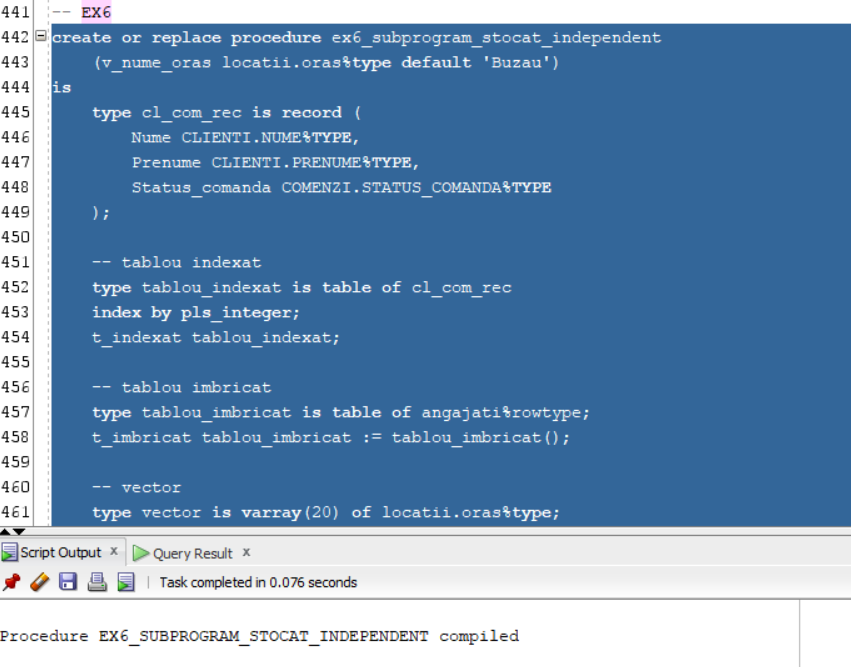
/

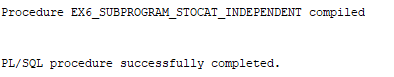
begin

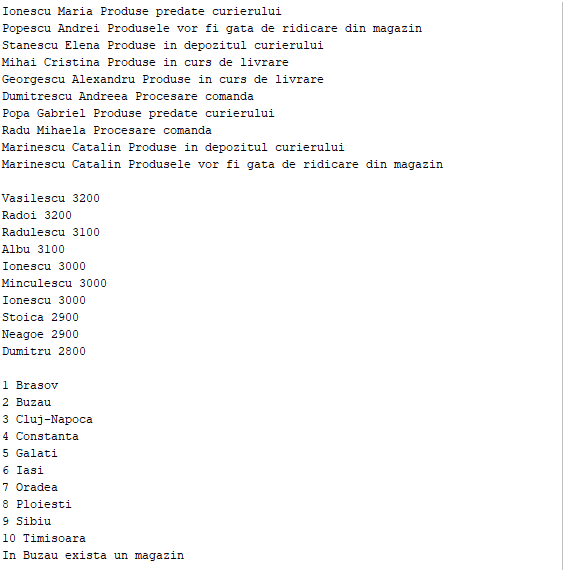
ex6\_subprogram\_stocat\_independent(‘Craiova’);

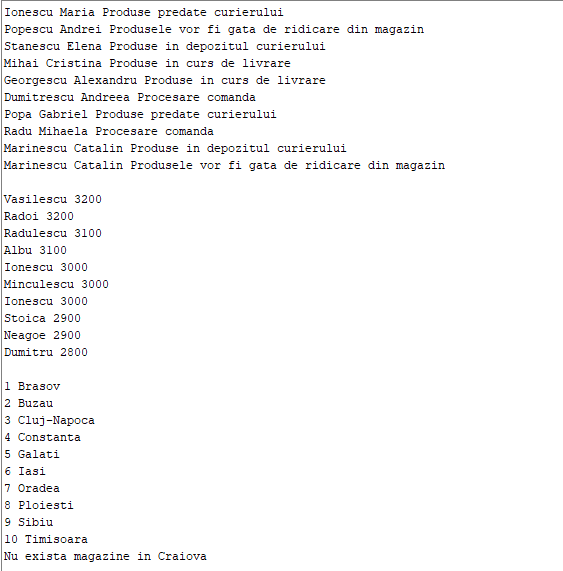
end;

/









**EXERCITIUL 7**

**Sa se afiseze numele magazinelor si angajatii care lucreaza acolo. Folositi doua tipuri diferite de cursoare: unul dintre acestea sa fie parametrizat, dependent de celalalt cursor.**

create or replace procedure ex7\_subprogram\_stocat\_independent is

cursor m is select id\_magazin, nume

from magazine;

cursor a (param magazine.id\_magazin%type) is

select nume, prenume, cnp

from angajati

where id\_magazin = param;

v\_id\_magazin magazine.id\_magazin%type;

v\_nume\_magazin magazine.nume%type;

begin

open m;

loop

fetch m into v\_id\_magazin, v\_nume\_magazin;

exit when m%notfound;

dbms\_output.put\_line('-----------------------------------------------');

dbms\_output.put\_line('Magazinul ' || v\_nume\_magazin);

dbms\_output.put\_line('-----------------------------------------------');

for i in a(v\_id\_magazin) loop

dbms\_output.put\_line(i.nume || ' ' || i.prenume);

end loop;

end loop;

end ex7\_subprogram\_stocat\_independent;

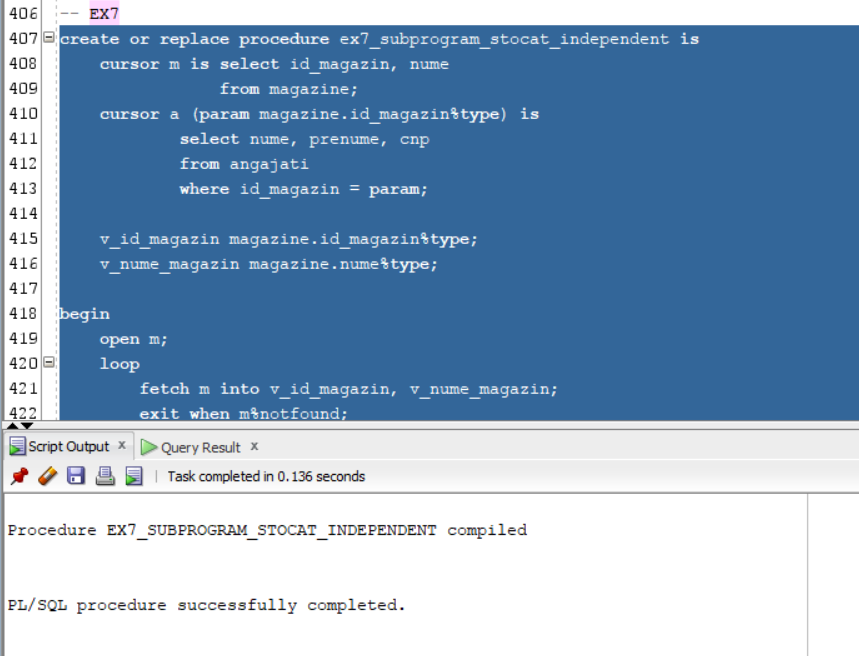
/

begin

ex7\_subprogram\_stocat\_independent;

end;

/





**EXERCITIUL 8**

**Scrieti un subprogram stocat independent care primeste ca parametru un ID de magazin si care returneaza un mesaj ce contine numele si prenumele managerului magazinului si orasul in care se afla acesta, daca se afla in Muntenia. Sa se trateze toate exceptiile in mod corespunzator.**

create or replace function ex8\_functie

(v\_id\_magazin magazine.id\_magazin%type default 6000)

return varchar2 is

v\_nume\_manager angajati.nume%type;

v\_prenume\_manager angajati.prenume%type;

v\_oras locatii.oras%type;

fara\_manager exception;

in\_afara\_munteniei exception;

fara\_manager\_in\_afara\_munteniei exception;

rezultat varchar2(1024);

begin

begin

select a.nume, a.prenume, oras into v\_nume\_manager, v\_prenume\_manager, v\_oras

from angajati a, locatii l, magazine m

where a.id\_angajat (+) = m.id\_manager

and m.id\_locatie = l.id\_locatie

and m.id\_magazin = v\_id\_magazin;

if v\_nume\_manager is null and v\_prenume\_manager is null

then if upper(v\_oras) not in ('ARGES', 'BRAILA', 'BUZAU', 'CALARASI', 'DAMBOVITA',

'GIURGIU', 'IALOMITA', 'ILFOV', 'PRAHOVA', 'TELEORMAN')

then raise fara\_manager\_in\_afara\_munteniei;

end if;

end if;

if v\_nume\_manager is null and v\_prenume\_manager is null

then raise fara\_manager;

end if;

if upper(v\_oras) not in ('ARGES', 'BRAILA', 'BUZAU', 'CALARASI', 'DAMBOVITA',

'GIURGIU', 'IALOMITA', 'ILFOV', 'PRAHOVA', 'TELEORMAN')

then raise in\_afara\_munteniei;

end if;

rezultat := 'Magazinul cu ID ' || v\_id\_magazin || ' are ca manager pe ' ||

v\_nume\_manager || ' ' || v\_prenume\_manager || ' ' || ' si se afla in ' ||

'orasul ' || v\_oras;

exception

when no\_data\_found

then dbms\_output.put\_line('Magazinul cu ID ' || v\_id\_magazin ||

' nu exista');

when fara\_manager

then dbms\_output.put\_line('Magazinul cu ID ' || v\_id\_magazin ||

' nu are manager');

when in\_afara\_munteniei

then dbms\_output.put\_line('Magazinul cu ID ' || v\_id\_magazin ||

' nu se afla in Muntenia');

when fara\_manager\_in\_afara\_munteniei

then dbms\_output.put\_line('Magazinul cu ID ' || v\_id\_magazin ||

' nu are manager si nu se afla in Muntenia');

end;

return rezultat;

end ex8\_functie;

/

begin

dbms\_output.put\_line(ex8\_functie(2009));

end;

/

begin

dbms\_output.put\_line(ex8\_functie(2077));

end;

/

begin

dbms\_output.put\_line(ex8\_functie(2002));

end;

/

begin

dbms\_output.put\_line(ex8\_functie(2007));

end;

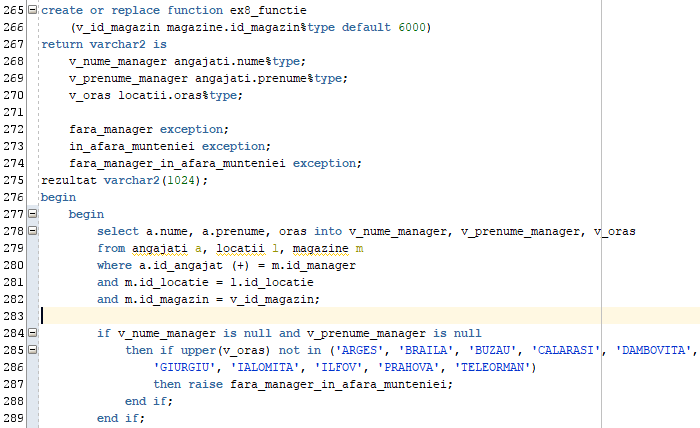
/

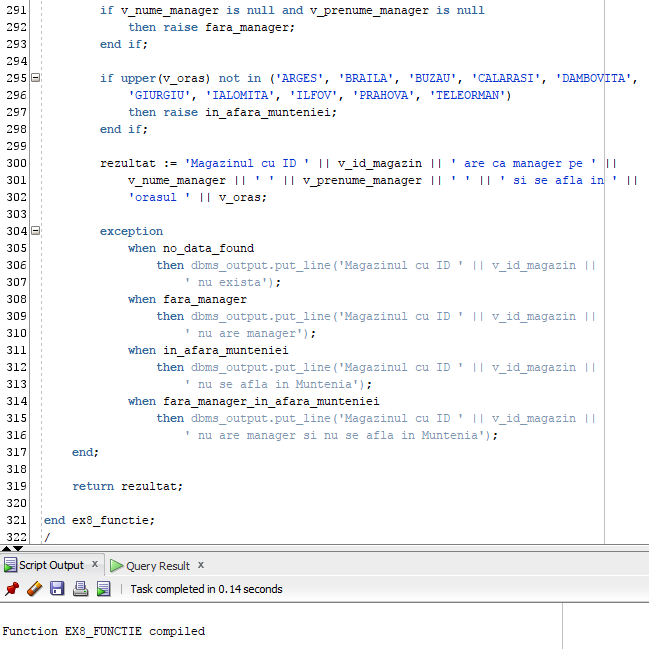
begin

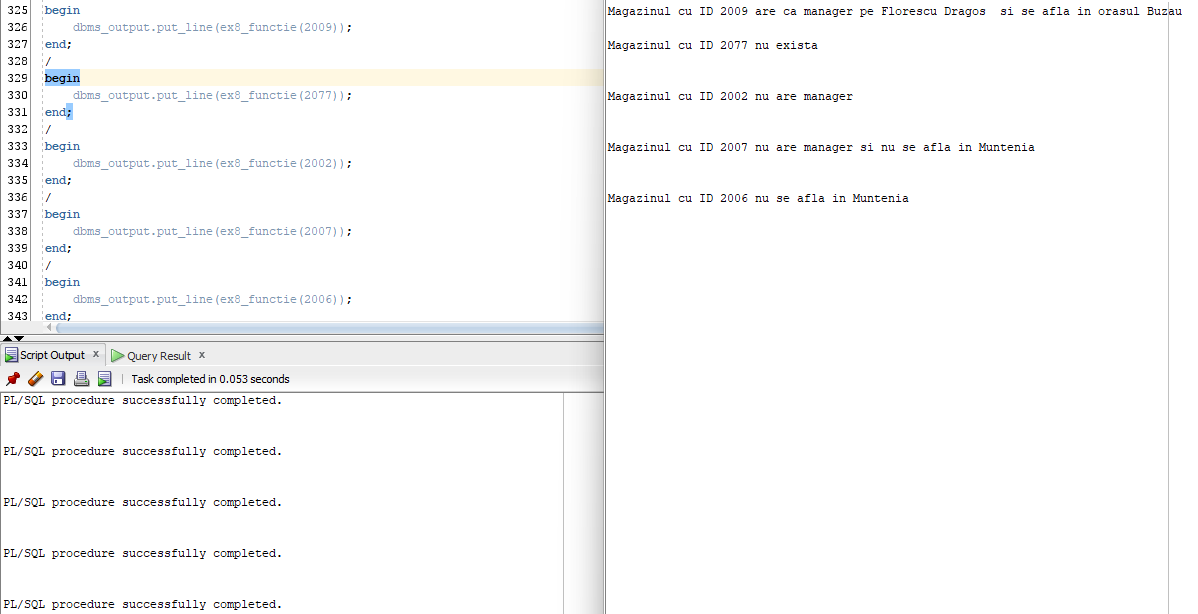
dbms\_output.put\_line(ex8\_functie(2006));

end;

/





****

**EXERCITIUL 9**

**Scrieti un subprogram stocat independent de tip procedura care primeste ca parametru un nume de magazin si care:**

* **Afiseaza orasul in care se afla magazinul.**
* **Afiseaza produsele care se gasesc in magazine (ID, stoc, nume).**
* **Afiseaza un client care a cumparat produse din acel magazin**

create or replace procedure ex9\_procedura

(v\_nume\_magazin magazine.nume%type default 'Nume magazin')

is

v\_id\_locatie magazine.id\_locatie%type;

v\_oras locatii.oras%type;

type rec\_tablou\_produse is record (

nume\_client clienti.nume%type,

prenume\_client clienti.prenume%type,

id\_produs produs\_magazin.id\_produs%type,

nume\_produs produse.nume\_produs%type,

stoc produs\_magazin.stoc%type

);

type tablou\_produse is table of rec\_tablou\_produse;

t\_produse tablou\_produse := tablou\_produse();

magazinul\_nu\_are\_produse exception;

begin

begin

select m.id\_locatie, oras into v\_id\_locatie, v\_oras

from magazine m, locatii l

where upper(v\_nume\_magazin) = upper(m.nume)

and m.id\_locatie = l.id\_locatie;

exception

when no\_data\_found

then raise\_application\_error(-20000, 'Nu exista niciun magazin cu numele '

|| v\_nume\_magazin);

when too\_many\_rows

then raise\_application\_error(-20001, 'Exista mai multe magazine cu numele '

|| v\_nume\_magazin);

end;

begin

select cl.nume, cl.prenume, pm.id\_produs, p.nume\_produs, pm.stoc

bulk collect into t\_produse

from produs\_magazin pm, produse p, magazine m, comenzi com, clienti cl

where p.id\_produs = pm.id\_produs

and pm.id\_magazin = m.id\_magazin

and m.id\_magazin = com.id\_magazin

and com.id\_client = cl.id\_client

and m.nume = v\_nume\_magazin

and m.id\_locatie = v\_id\_locatie;

if t\_produse.count = 0

then raise magazinul\_nu\_are\_produse;

end if;

dbms\_output.put\_line('In magazinul ' || v\_nume\_magazin || ' din '

|| v\_oras || ' gasim urmatoarele produse: ');

for i in t\_produse.first..t\_produse.last loop

dbms\_output.put\_line('(ID: ' || t\_produse(i).id\_produs || ') ' ||

t\_produse(i).stoc || ' x ' || t\_produse(i).nume\_produs);

end loop;

dbms\_output.put\_line(t\_produse(1).nume\_client || ' ' ||

t\_produse(1).prenume\_client || ' a comandat de la acest magazin');

exception

when magazinul\_nu\_are\_produse

then dbms\_output.put\_line('Magazinul ' || v\_nume\_magazin

|| ' nu are niciun produs');

end;

end ex9\_procedura;

/

begin

ex9\_procedura;

end;

/

begin

ex9\_procedura('DVD Delight Cluj-Napoca');

end;

/

begin

ex9\_procedura('Just Watch Oradea');

end;

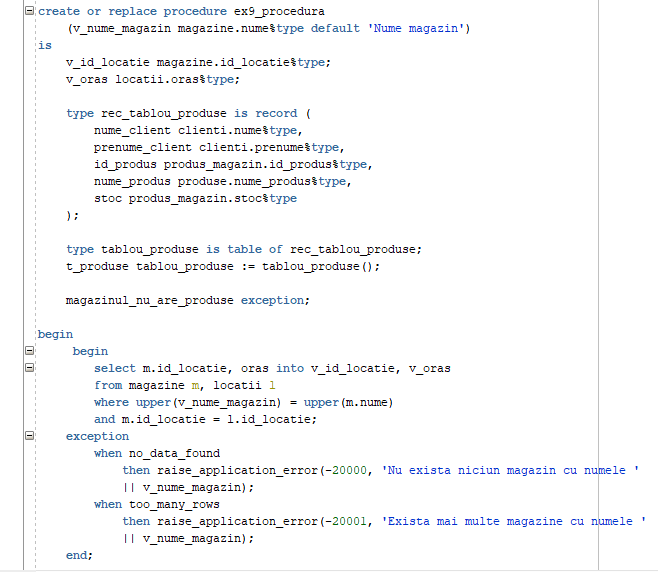
/

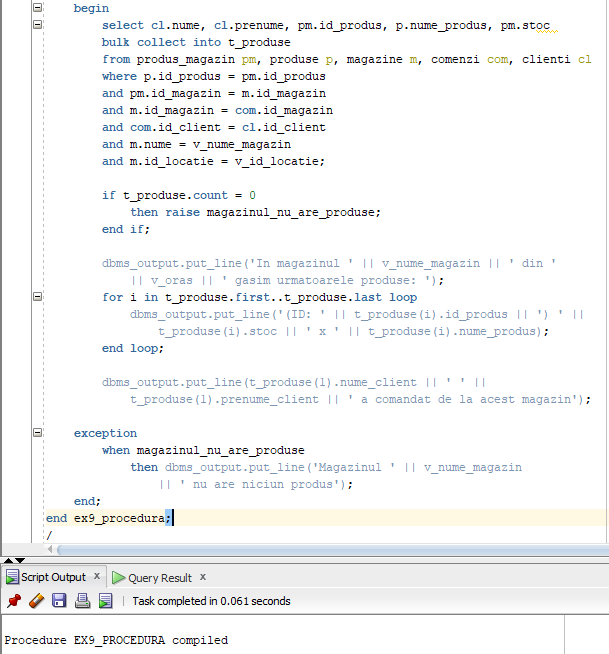
begin

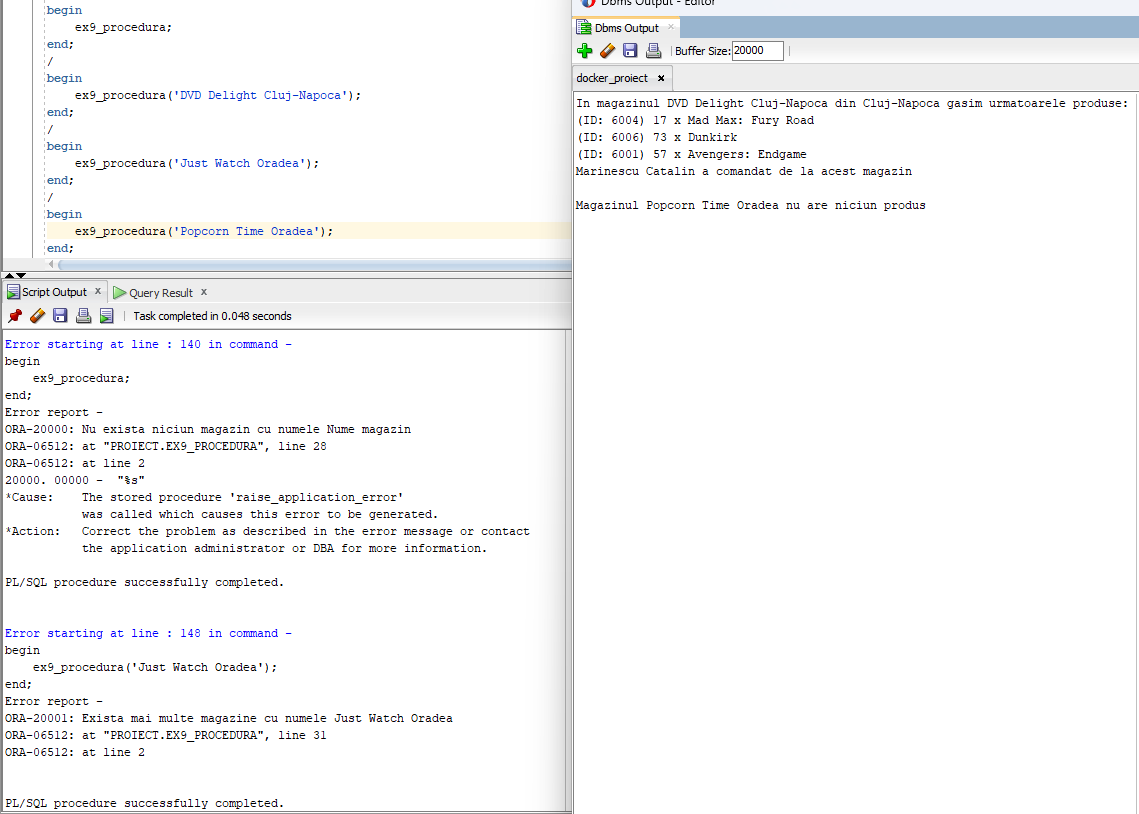
ex9\_procedura('Popcorn Time Oradea');

end;

/







**EXERCITIUL 10**

**Scrieti un trigger la nivel de comanda care sa nu permita adaugarea unui nou angajat in week-end.**

create or replace trigger ex10\_trigger

before insert on angajati

begin

if (to\_char(sysdate, 'D') = 7)

or (to\_char(sysdate, 'D') = 1)

then raise\_application\_error(-20005, 'Nu se pot adauga angajati in week-end');

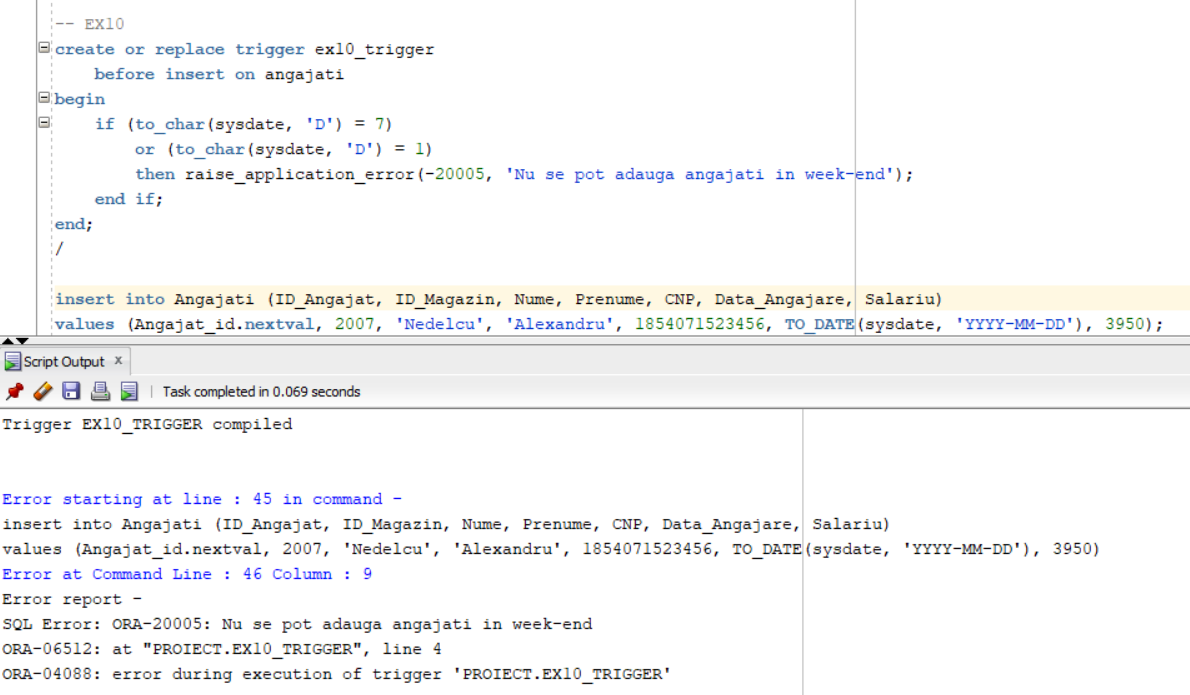
end if;

end;

/

insert into Angajati (ID\_Angajat, ID\_Magazin, Nume, Prenume, CNP, Data\_Angajare, Salariu)

values (Angajat\_id.nextval, 2007, 'Nedelcu', 'Alexandru', 1854071523456, TO\_DATE(sysdate, 'YYYY-MM-DD'), 3950);



**EXERCITIUL 11**

**Scrieti un trigger la nivel de linie care sa nu permita marirea pretului unui produs in zilele de duminica.**

create or replace trigger ex11\_trigger

before update of pret on produse

for each row

begin

if (:new.pret > :old.pret) and (to\_char(sysdate, 'D') = 1)

then raise\_application\_error(-20005, 'Pretul nu poate fi marit in zilele'

|| ' de duminica');

end if;

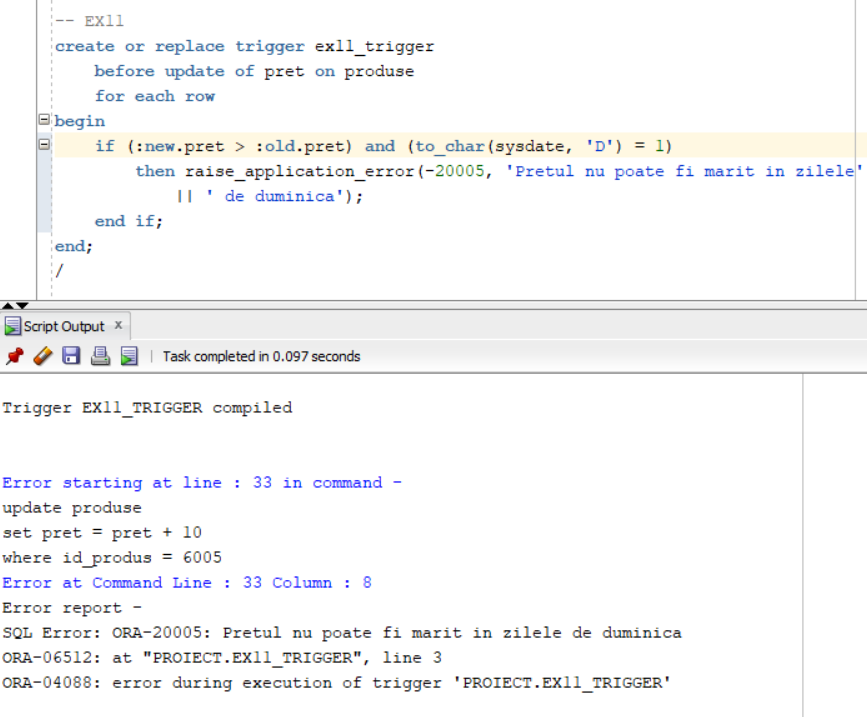
end;

/

update produse

set pret = pret + 10

where id\_produs = 6005;



**EXERCITIUL 12**

**Scrieti un trigger care sa introduca date (numele utilizatorului, numele bazei de date, evenimentul sistem, numele obiectului, data si ora producerii evenimentului) intr-un tabel info\_actiuni**

create table info\_actiuni

(utilizator varchar2(30),

nume\_bd varchar2(50),

eveniment varchar2(20),

nume\_obiect varchar2(30),

ora timestamp

);

create or replace trigger ex12\_trigger

after create or drop or alter on schema

begin

insert into info\_actiuni

values (sys.login\_user, sys.database\_name, sys.sysevent, sys.dictionary\_obj\_name, current\_timestamp);

end;

/

create table exemplu (

id number,

name varchar2(50)

);

drop table exemplu;

SELECT \* FROM info\_actiuni;

drop table audit\_mra;

